

AEA 4-03

Monitorització

NAGIOS

Marçal Ball-Iloera i Marc Villalobos
SEGURETAT I ALTA DISPONIBILITAT | ASIX 2

ÍNDEX

ENUNCIAT	2
Nagios és un sistema de monitorització, mitjançant una aplicació web, que permet monitoritzar màquines, programari i serveis.....	2
Buscar a Internet la documentació necessària per a configurar aquesta aplicació sobre una màquina per tal que permeti monitoritzi una altre.	2
Les característiques de la màquina a controlar seran,.....	2
♣ S'anomenarà 'SERVER-WEB'	3
♣ Serà una màquina virtual amb 1 GiB de memòria Ram i un disc de 40 GiB.	3
♣ Com a sistema operatiu emprarà Debian, actualitzat i amb les VMware Tools instal·lades.....	5
♣ Tecnologia LAMP i servidor SSH.	8
♣ Disposarà d'una IP pròpia.	11
Les característiques de la màquina que contindrà el sistema de monitorització seran,	11
♣ S'anomenarà 'SERVER-MONITOR'	11
♣ Serà una màquina virtual amb 1 GiB de memòria Ram i un disc de 20 GiB.	12
♣ Com a sistema operatiu emprarà Debian, actualitzat i amb les VMware Tools instal·lades.....	13
♣ Aplicació Nagios.	17
♣ Disposarà d'una IP pròpia.	44
Des de una màquina que no sigui 'SERVER-WEB' ni 'SERVER-MONITOR', a través del sistema de monitorització, es podrà controlar mitjançant components gràfics (icones i gràfiques) els següents elements de la màquina 'SERVER-WEB',	49
• El processador (gràfica temporal de la càrrega)	49
• La memòria (gràfica temporal de la memòria disponible).	50
• El espai de disc (gràfiques temporals de l'espai disponible i l'ocupat).....	50
• El servei Apache.	50
• El servei MySQL.....	51
El servei SSH.	51

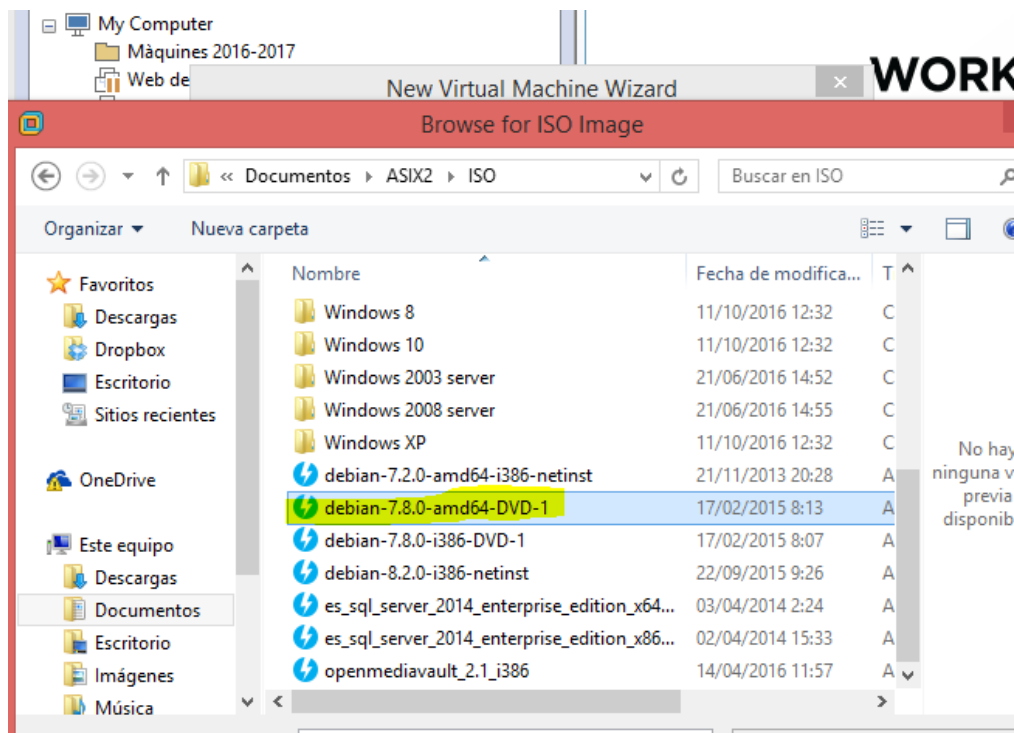
ENUNCIAT

Nagios és un sistema de monitorització, mitjançant una aplicació web, que permet monitoritzar màquines, programari i serveis.

Buscar a Internet la documentació necessària per a configurar aquesta aplicació sobre una màquina per tal que permeti monitoritzi una altre.

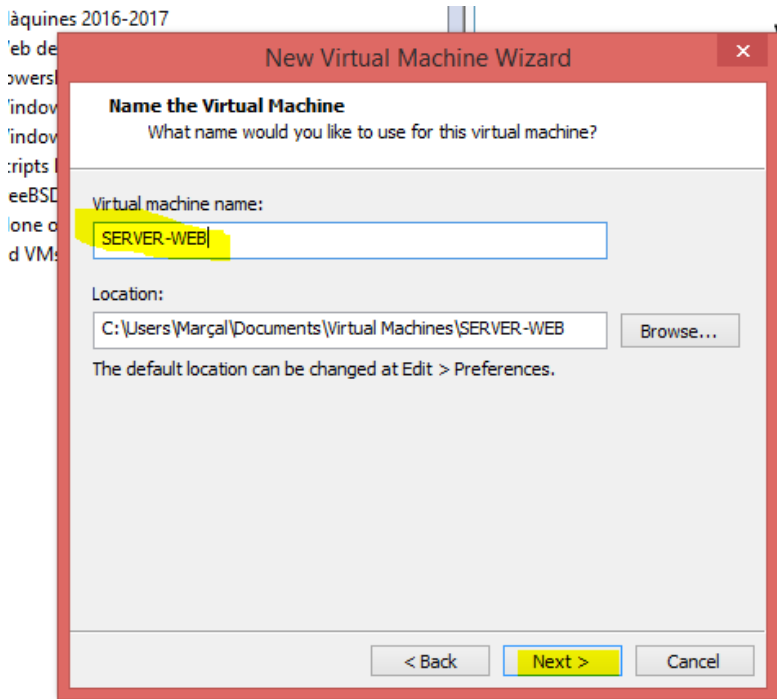
Les característiques de la màquina a controlar seran,

Per començar, seleccionarem el sistema operatiu que hi vulguem posar. Serà un “debian” amb versió 7.8

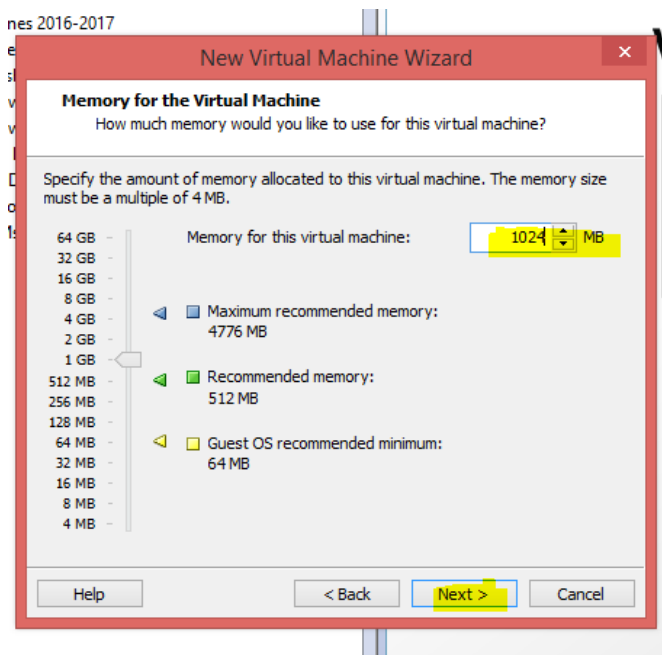


♣ S'anomenarà 'SERVER-WEB'.

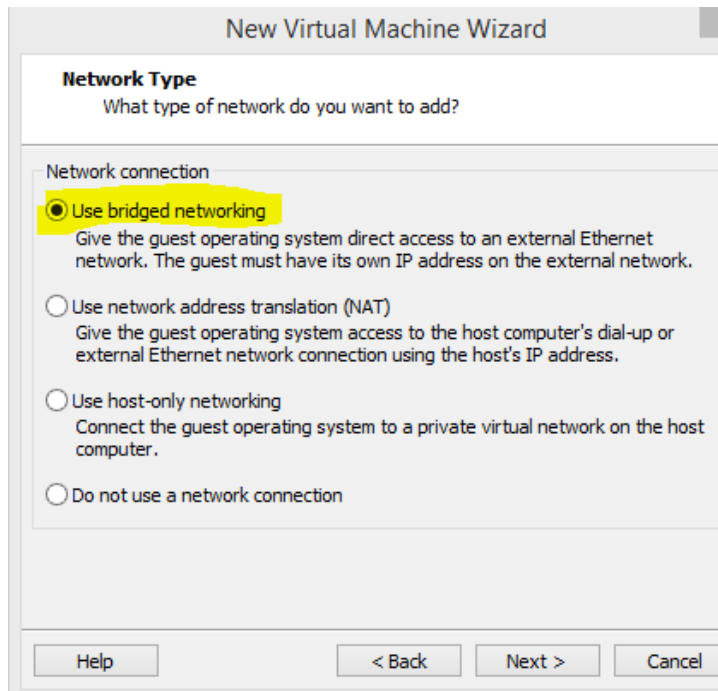
L'anomenarem "SERVER-WEB", tal i com ens demana que sigui així, a l'enunciat. I cliquem a "Next"



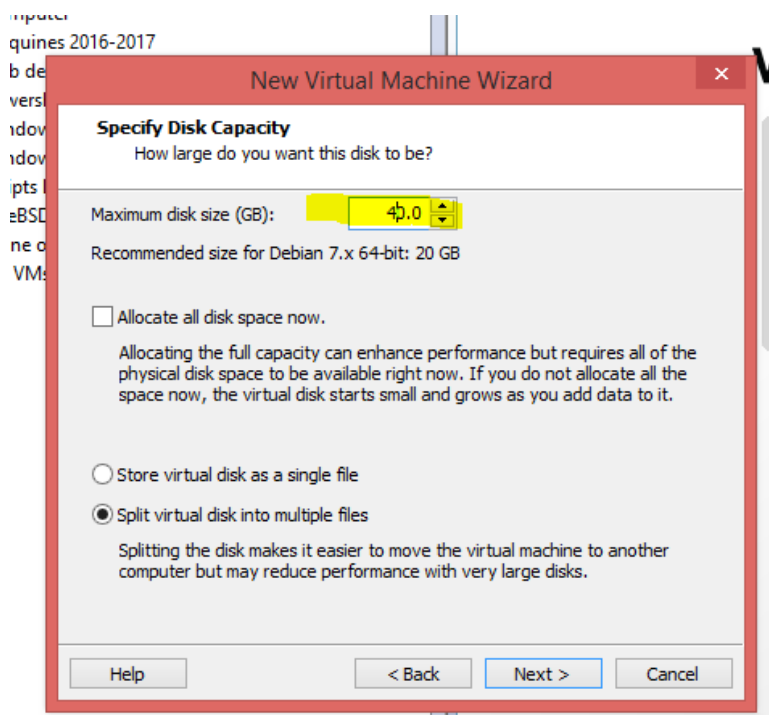
♣ Serà una màquina virtual amb 1 GiB de memòria Ram i un disc de 40 GiB. Li assignarem 1 GB de memòria RAM. Concretament, per ser exactes, li posem: 1024 MB



La connexió serà, com sempre, amb totes les màquines creades: “Bridged”, perquè ens assigni una IP privada, només per a nosaltres

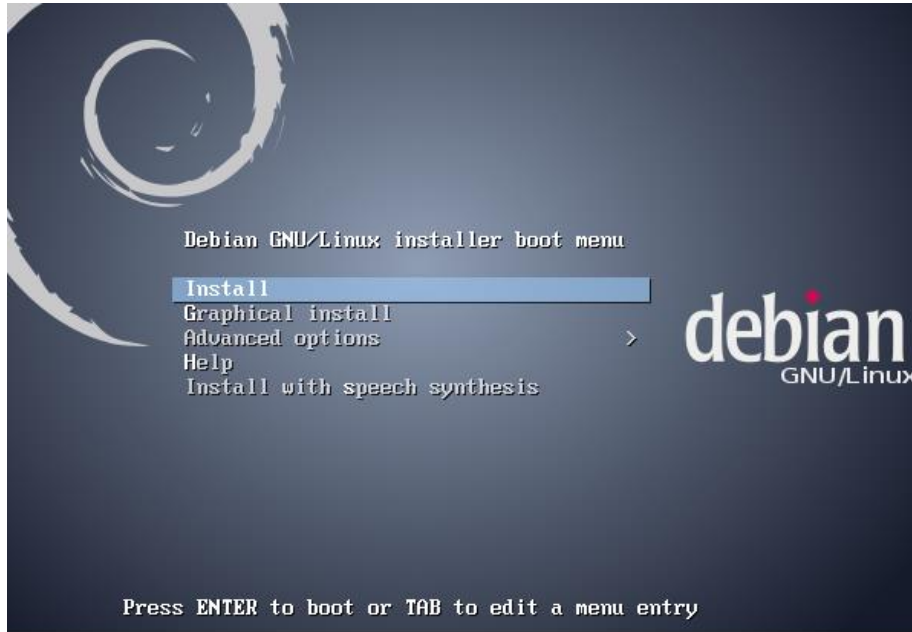


Com també ens demana a l'enunciat, li assignem 40 GB de disc dur

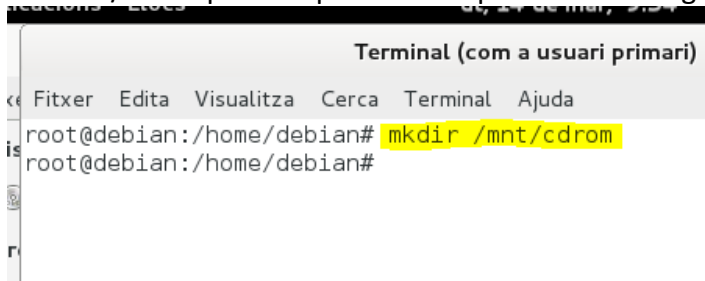


♣ Com a sistema operatiu emprarà Debian, actualitzat i amb les VMware Tools instal·lades.

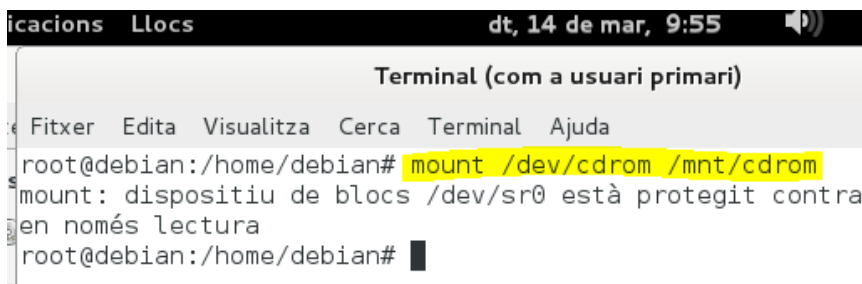
Com ja hem dit anteriorment, caldrà instal·lar la màquina Debian com a sistema operatiu. Cliquem a “Install” per a procedir a executar la instal·lació



Posem la comanda “mkdir /mnt/cdrom” per crear el directori “cdrom” i a dins d’ell, creem “/mnt” que serà per tenir un punt de muntatge.



Executem el “cdrom” i ho fem amb la comanda “mount /dev/cdrom /mnt/cdrom”



A continuació fem un “cp /mnt/cdrom/VMwareTools-10.1.5-5055693.tar.gz /tmp” per així poder copiar l’arxiu VMwareTools a dins el directori “/tmp”.

```
root@debian:/home/debian# cp /mnt/cdrom/VMwareTools-10.1.5-5055693.tar.gz /tmp
root@debian:/home/debian#
```

Fem un “cd /tmp” per anar en aquest directori

```
Fitxer Edita Visualitza Cerca
root@debian:/tmp# cd ..
root@debian:/# cd /tmp
root@debian:/tmp#
```

Executem la comanda “tar -zxvf VMwareTools-10.1.5-5055693.tar.gz” per extreure o descomprimir el que hi ha a dins aquest arxiu a dins el directori “/tmp” tal com podem veure.

```
vmware-tools-distrib/FILES
root@debian:/tmp# tar -zxvf VMwareTools-10.1.5-5055693.tar.gz
cdrom0
```

[cdrom0] [vmware-tools-d...] Terminal (com a ...) cdrom0

Amb un “ls”, observem com els fitxers s’han descomprimit correctament.

A continuació, accedim al directori “vmware-tools-distrib” amb la comanda “cd vmware-tools-distrib”

```
root@debian:/tmp# ls
libgksu-AUA3ny pulse-PKdhtXMmr18n VMwareTools-10.1.5-5055693.tar.gz
pulse-2L9K88eMlGn7 ssh-p5t2DQWZBHX0 vmware-tools-distrib
pulse-CcctT9RwKSB1 tracker-debian
root@debian:/tmp# cd vmware-tools-distrib/
root@debian:/tmp/vmware-tools-distrib#
```

Una vegada ja som a dins, executem la comanda “./vmware-install.pl” per acabar d’instal·lar i configurar les “vmware tools”

```
Terminal (com a usuari primari)
Fitxer Edita Visualitza Cerca Terminal Ajuda
root@debian:/tmp/vmware-tools-distrib# ./vmware-install.pl
```

Observem com, finalment ja tenim instal·lades les VMware Tools. Només faltaria reiniciar el sistema operatiu

```

Fitxer  Edita  Visualitza  Cerca  Terminal  Ajuda
Guest operating system daemon: done
VGAAuthService: done
Common Agent: done
The configuration of VMware Tools 10.1.5 build-5055693 for Linux for this
running kernel completed successfully.

You must restart your X session before any mouse or graphics changes take
effect.

You can now run VMware Tools by invoking "/usr/bin/vmware-toolbox-cmd" from the
command line.

To enable advanced X features (e.g., guest resolution fit, drag and drop, and
file and text copy/paste), you will need to do one (or more) of the following:
1. Manually start /usr/bin/vmware-user
2. Log out and log back into your desktop session
3. Restart your X session.

Enjoy,

--the VMware team

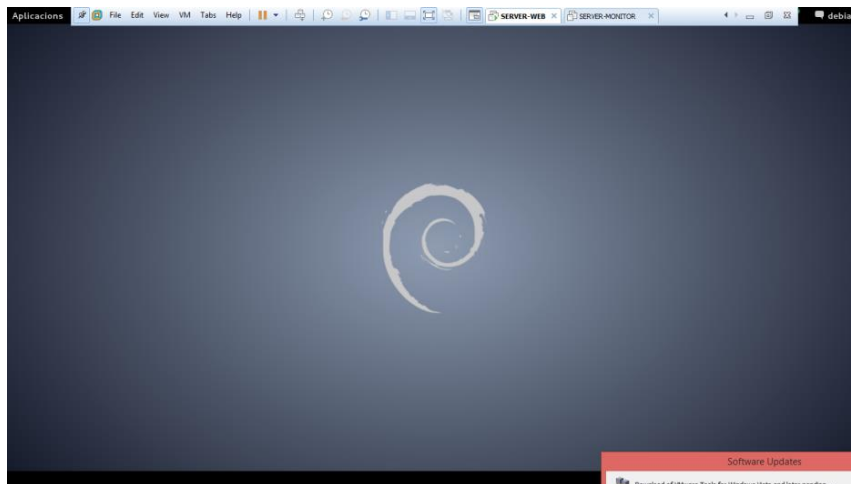
Found VMware Tools CDRROM mounted at /media/cdrom0. Ejecting device /dev/sr0 ...

Found VMware Tools CDRROM mounted at /mnt/cdrom. Ejecting device /dev/sr0 ...
root@debian:/tmp/vmware-tools-distrib#

```



Una vegada reiniciada la màquina, podrem comprovar que ja tenim les VMware Tools en funcionament.



A la màquina virtual clonada ja tenim les VMware Tools instal·lades. Per comprovar la versió de les VMware Tools farem la següent comanda:

“grep buildNr /usr/bin/vmware-config-tools.pl”

```
root@debian:/home/usuario# grep buildNr /usr/bin/vmware-config-tools.pl
my $buildNr;
$buildNr = '9.6.0 build-1294478';
return remove_whitespace($buildNr);
root@debian:/home/usuario#
```

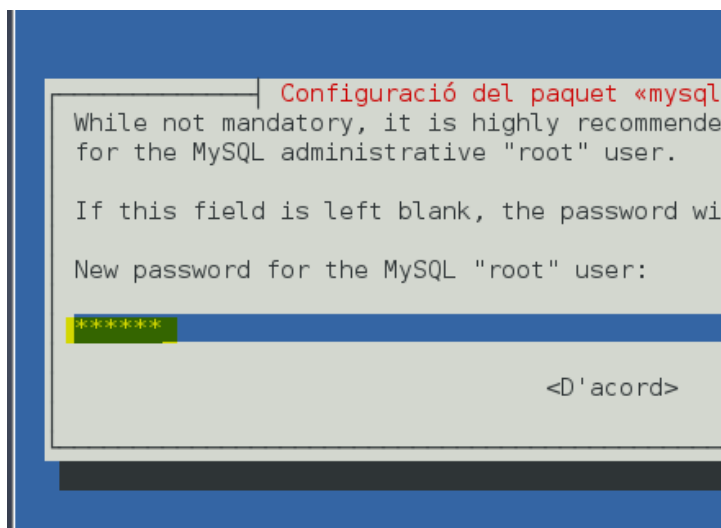
♣ Tecnologia LAMP i servidor SSH.

Procedirem a instal·lar el “LAMP”, d’aquesta manera, començarem amb la comanda:

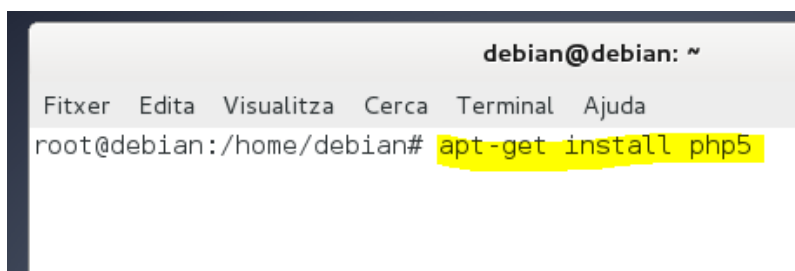
“apt-get install mysql-server-5.5”

```
root@debian:/home/debian# apt-get install mysql-server-5.5
S'està llegint la llista de paquets... Fet 0%
S'està construint l'arbre de dependències
S'està llegint la informació de l'estat... Fet%
S'instal·laran els següents paquets extres:
  libaiol libdbd-mysql-perl libdbi-perl libhtml-template-perl lib
  mysql-client-5.5 mysql-common mysql-server-core-5.5
Paquets suggerits:
  libipc-sharedcache-perl libterm-readkey-perl tinyca
S'instal·laran els paquets NOUS següents:
  libaiol libdbd-mysql-perl libdbi-perl libhtml-template-perl lib
  mysql-client-5.5 mysql-common mysql-server-5.5 mysql-server-cor
0 actualitzats, 9 nous a instal·lar, 0 a suprimir i 0 no actualitz
S'ha d'obtenir 8881 kB/9854 kB d'arxius.
Després d'aquesta operació s'empraran 97,6 MB d'espai en disc add
Voleu continuar [S/n]?
```

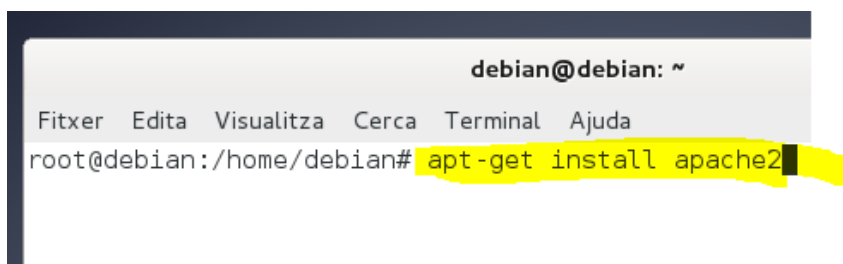
Li assignarem una contrasenya, s'anomenarà "debian"



A continuació, procedim a instal·lar "Php", amb la comanda: "apt-get install php5" que serà allà on hi ha les bases de dades corresponents



Per ultim, instal·lem l'"Apache", on, ja tindrem el "LAMP" instal·lat i configurat correctament. Amb la comanda "apt-get install apache2"



Seguidament procedirem a instal·lar l'SSH. Per fer-ho utilitzarem la comanda: “apt-get install openssh-server”

```

Archivo  Editar  Ver  Buscar  Terminal  Ayuda
root@debian:/home/usuario# apt-get install openssh-server
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes extras:
  openssh-client
Paquetes sugeridos:
  ssh-askpass libpam-ssh keychain monkeysphere rssh molly-guard ufw
Se actualizarán los siguientes paquetes:
  openssh-client openssh-server
2 actualizados, 0 se instalarán, 0 para eliminar y 218 no actualizados.
Necesito descargar 1.388 kB de archivos.
Se utilizarán 0 B de espacio de disco adicional después de esta operación.
¿Desea continuar [S/n]? █

```

Després utilitzarem el servei SSH per connectar-nos a la màquina local amb l'usuari “usuario” utilitzant la comanda: “ssh usuario@localhost”

```

root@debian:/home/usuario# ssh usuario@localhost
The authenticity of host 'localhost (:::1)' can't be established.
ECDSA key fingerprint is 57:f7:a3:e4:4e:04:92:2f:88:be:14:eb:dc:17:53:f7.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'localhost' (ECDSA) to the list of known hosts.
usuario@localhost's password:
Linux debian 3.2.0-4-686-pae #1 SMP Debian 3.2.78-1 i686

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
You have new mail.
usuario@debian:~$ █

```

♣ Disposarà d'una IP pròpia.

Observem com, disposarà d'una IP pròpia, on l'agafa per "DHCP" i no és estàtica, per tant, cada vegada que tanquem la màquina, ens canviarà la IP

```
root@debian:/home/debian# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:0e:a3:06
          inet addr:172.17.201.189  Bcast:172.17.255.255  Mask:255.
          inet6 addr: fe80::20c:29ff:fe0e:a306/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:15525 errors:0 dropped:0 overruns:0 frame:0
          TX packets:148 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2390181 (2.2 MiB)  TX bytes:12880 (12.5 KiB)

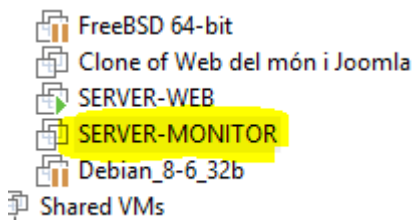
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:8 errors:0 dropped:0 overruns:0 frame:0
          TX packets:8 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:480 (480.0 B)  TX bytes:480 (480.0 B)

root@debian:/home/debian#
```

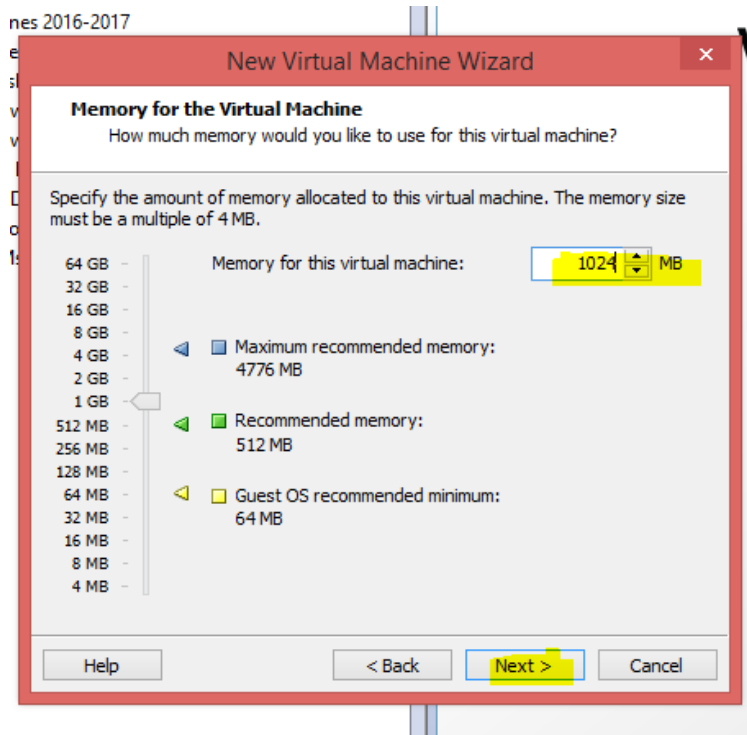
Les característiques de la màquina que contindrà el sistema de monitorització seran,

♣ S'anomenarà 'SERVER-MONITOR'.

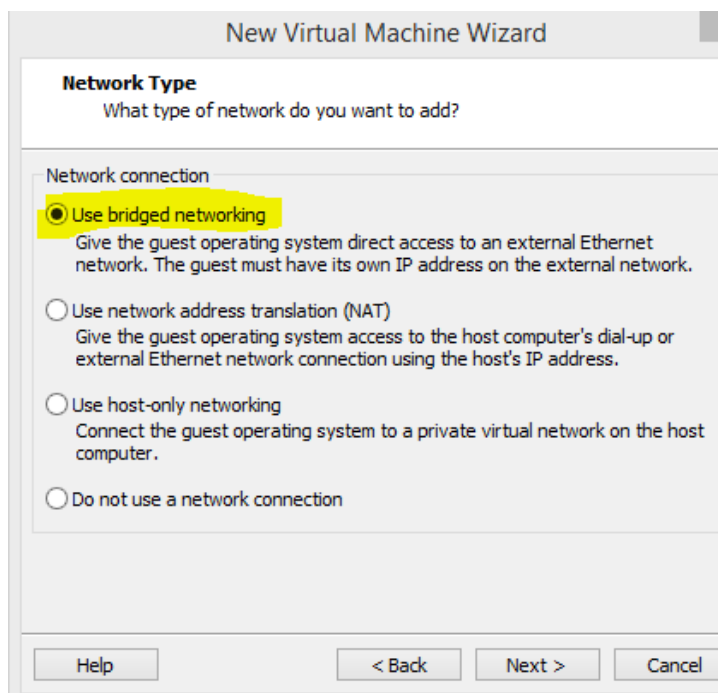
L'anomenarem "SERVER-MONITOR", tal i com ens demana que sigui així, a l'enunciat. I cliquem a "Next"



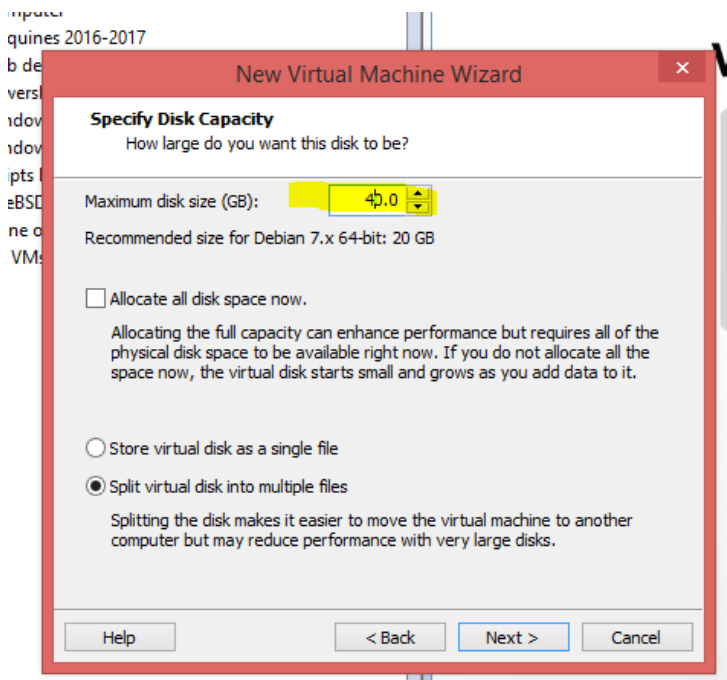
♣ Serà una màquina virtual amb 1 GiB de memòria Ram i un disc de 20 GiB. Li assignarem 1 GB de memòria RAM. Concretament, per ser exactes, li posem: 1024 MB



La connexió serà, com sempre, amb totes les màquines creades: “Bridged”, perquè ens assigni una IP privada, només per a nosaltres

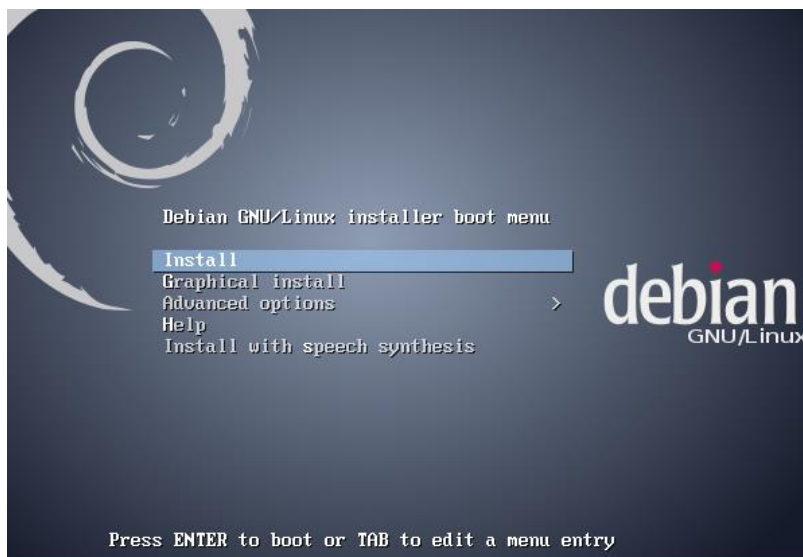


Com també ens demana a l'enunciat, li assignem 40 GB de disc dur



♣ Com a sistema operatiu emprarà Debian, actualitzat i amb les VMware Tools instal·lades.

Com ja hem dit anteriorment, caldrà instal·lar la màquina Debian com a sistema operatiu. Cliquem a “Install” per a procedir a executar la instal·lació



Posem la comanda “mkdir /mnt/cdrom” per crear el directori “cdrom” i a dins d’ell, creem “/mnt” que serà per tenir un punt de muntatge.

```

Terminal (com a usuari primari)
Fitxer Edita Visualitza Cerca Terminal Ajuda
root@debian:/home/debian# mkdir /mnt/cdrom
root@debian:/home/debian#

```

Executem el “cdrom” i ho fem amb la comanda “mount /dev/cdrom /mnt/cdrom”

```

icacions Llocs dt, 14 de mar, 9:55
Terminal (com a usuari primari)
Fitxer Edita Visualitza Cerca Terminal Ajuda
root@debian:/home/debian# mount /dev/cdrom /mnt/cdrom
mount: dispositiu de blocs /dev/sr0 està protegit contra
en només lectura
root@debian:/home/debian#

```

A continuació fem un “cp /mnt/cdrom/VMwareTools-10.1.5-5055693.tar.gz /tmp” per així poder copiar l’arxiu VMwareTools a dins el directori “/tmp”.

```

root@debian:/home/debian# cp /mnt/cdrom/VMwareTools-10.1.5-5055693.tar.gz /tmp
root@debian:/home/debian#

```

Fem un “cd /tmp” per anar en aquest directori

```

Fitxer Edita Visualitza Cerca
root@debian:/tmp# cd ..
root@debian:/# cd /tmp
root@debian:/tmp#

```

Executem la comanda “tar -zxvf VMwareTools-10.1.5-5055693.tar.gz”. Per extreure o descomprimir el que hi ha a dins aquest arxiu a dins el directori “/tmp” tal com podem veure.

```
vmware-tools-distrib/FILES
root@debian:/tmp# tar -zxvf VMwareTools-10.1.5-5055693.tar.gz
```

Amb un “ls”, observem com els fitxers s’han descomprimit correctament.

A continuació, accedim al directori “vmware-tools-distrib” amb la comanda “cd vmware-tools-distrib”

```
root@debian:/tmp# ls
libgksu-AUA3ny      pulse-PKdhtXMmr18n  VMwareTools-10.1.5-5055693.tar.gz
pulse-2L9K88eMlGn7  ssh-p5t2DQWZBHX0    vmware-tools-distrib
pulse-CcctT9RwKSB1  tracker-debian
root@debian:/tmp# cd vmware-tools-distrib/
root@debian:/tmp/vmware-tools-distrib#
```

Una vegada ja som a dins, executem la comanda “./vmware-install.pl” per acabar d’instal·lar i configurar les “vmware tools”

```
Terminal (com a usuari primari)
Fitxer  Edita  Visualitza  Cerca  Terminal  Ajuda
root@debian:/tmp/vmware-tools-distrib# ./vmware-install.pl
```


Observem com, finalment ja tenim instal·lades les VMware Tools. Només faltaria reiniciar el sistema operatiu

```

Fitxer  Edita  Visualitza  Cerca  Terminal  Ajuda
Guest operating system daemon: done
VGAAuthService: done
Common Agent: done
The configuration of VMware Tools 10.1.5 build-5055693 for Linux for this
running kernel completed successfully.

You must restart your X session before any mouse or graphics changes take
effect.

You can now run VMware Tools by invoking "/usr/bin/vmware-toolbox-cmd" from the
command line.

To enable advanced X features (e.g., guest resolution fit, drag and drop, and
file and text copy/paste), you will need to do one (or more) of the following:
1. Manually start /usr/bin/vmware-user
2. Log out and log back into your desktop session
3. Restart your X session.

Enjoy,

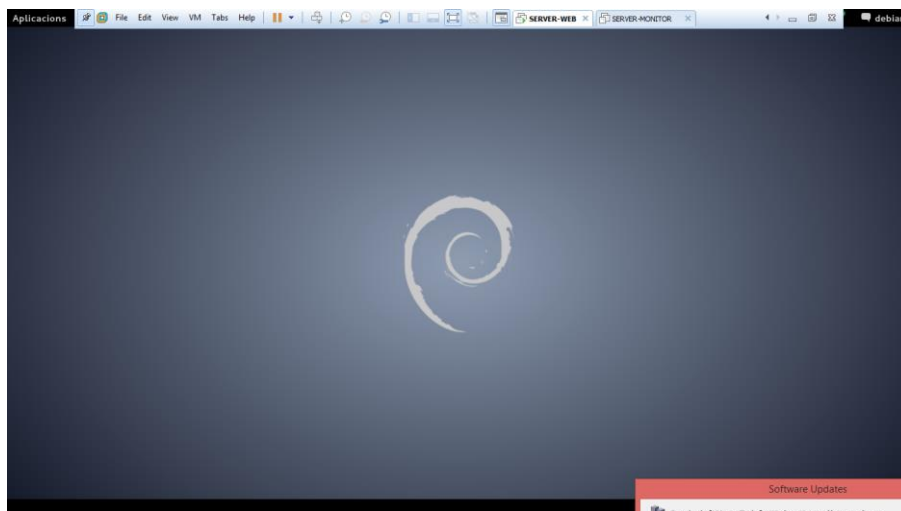
--the VMware team

Found VMware Tools CDRom mounted at /media/cdrom0. Ejecting device /dev/sr0 ...

Found VMware Tools CDRom mounted at /mnt/cdrom. Ejecting device /dev/sr0 ...
root@debian:/tmp/vmware-tools-distrib#

```

Una vegada reiniciada la màquina, podem comprovar que ja tenim les VMware Tools en funcionament.



♣ Aplicació Nagios.

Primer de tot haurem d'instal·lar una sèrie de paquets en el nostre sistema.

Abans de procedir a instal·lar-los, actualitzarem els paquets del sistema amb la següent comanda:

`apt-get -y update`

```
root@debian:/home/usuario# apt-get update
Obj http://ftp.es.debian.org wheezy Release.gpg
Des:1 http://ftp.es.debian.org wheezy-updates Release.gpg [1.554 B]
Obj http://ftp.es.debian.org wheezy Release
Des:2 http://ftp.es.debian.org wheezy-updates Release [155 kB]
Obj http://ftp.es.debian.org wheezy/main Sources
Obj http://ftp.es.debian.org wheezy/main i386 Packages
Obj http://ftp.es.debian.org wheezy/main Translation-es
Obj http://ftp.es.debian.org wheezy/main Translation-en
Des:3 http://ftp.es.debian.org wheezy-updates/main Sources [5.505 B]
Obj http://ftp.es.debian.org wheezy-updates/main i386 Packages/DiffIndex
Obj http://ftp.es.debian.org wheezy-updates/main Translation-en/DiffIndex
..
```

Ara ja podrem començar a instal·lar els paquets que necessitem. El primer que instal·larem és l'*Apache2* i ho farem amb la següent comanda:

`apt-get -y install apache2`

```
root@debian:/home/usuario# apt-get install apache2
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes extras:
  apache2-mpm-itk apache2.2-bin apache2.2-common
  libapache2-mod-php5filter php5 php5-cli php5-common php5-gd
  php5-mcrypt php5-mysql
Paquetes sugeridos:
  apache2-doc apache2-suexec apache2-suexec-custom php-pear
Los siguientes paquetes se ELIMINARÁN:
  apache2-mpm-prefork libapache2-mod-php5
Se instalarán los siguientes paquetes NUEVOS:
  apache2-mpm-itk libapache2-mod-php5filter
Se actualizarán los siguientes paquetes:
```

El segon paquet que instal·larem serà el *PHP5* i ho farem utilitzant la següent comanda:

`apt-get -y install libapache2-mod-php5`

```
root@debian:/home/usuario# apt-get install libapache2-mod-php5
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Paquetes sugeridos:
  php-pear
Los siguientes paquetes se ELIMINARÁN:
  libapache2-mod-php5filter
Se instalarán los siguientes paquetes NUEVOS:
  libapache2-mod-php5
```

Seguidament instal·larem el compilador GCC i les llibreries de desenvolupament utilitzant la següent comanda:

`apt-get -y install build-essential`

```
root@debian:~# apt-get install build-essential
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
build-essential ya está en su versión más reciente.
0 actualizados, 0 se instalarán, 0 para eliminar y 210 no actualizados.
```

Per últim, instal·larem les llibreries GD amb la comanda de sota:

```
apt-get -y install libgd2-xpm-dev
```

```
root@debian:~# apt-get install libgd2-xpm-dev
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes extras:
  fontconfig-config libexpat1 libexpat1-dev libfontconfig1
  libfontconfig1-dev libfreetype6 libfreetype6-dev libgd2-xpm
  libjpeg8-dev libpng12-dev libpthread-stubs0 libpthread-stubs0-dev
  libx11-6 libx11-dev libx11-doc libxau-dev libxcb1-dev libxdmcp-dev
  libxpm-dev libxpm4 pkg-config x11proto-core-dev x11proto-input-dev
  x11proto-kb-dev xorg-sgml-doctools xtrans-dev zlib1g-dev
Paquetes sugeridos:
  libgd-tools libxcb-doc
```

Després d'haver instal·lat els paquets crearem un usuari anomenat *nagios* i li assignarem una contrasenya utilitzant les comandes:

```
useradd -m -s /bin/bash nagios
```

```
passwd nagios
```

```
root@debian:~# useradd -m -s /bin/bash nagios
root@debian:~# passwd nagios
Introduzca la nueva contraseña de UNIX:
Vuelva a escribir la nueva contraseña de UNIX:
passwd: contraseña actualizada correctamente
root@debian:~# usermod -G nagios nagios
root@debian:~# █
```

Seguidament crearem un nou grup anomenat *nagcmd* per poder allotjar les comandes utilitzades per la interfície web, afegirem en el grup *nagcmd* a l'usuari *nagios* i a l'usuari d'apache (*www-data*) utilitzant les següents comandes respectivament:

```
groupadd nagcmd
```

```
usermod -a -G nagcmd nagios
```

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

```
root@debian:~# groupadd nagcmd
root@debian:~# usermod -a -G nagcmd nagios
root@debian:~# usermod -a -G nagcmd www-data
root@debian:~# █
```

Després d'haver acabat els passos anteriors ja podrem procedir a instal·lar el nagios utilitzant la comanda:

```
Wget      http://sourceforge.net/projects/nagios/files/nagios-4.x/nagios-4.0.6/nagios-4.0.6.tar.gz
```

```
root@debian:~# wget http://sourceforge.net/projects/nagios/files/nagios-4.x/nagios-4.0.6/nagios-4.0.6.tar.gz
--2017-03-28 08:50:04-- http://sourceforge.net/projects/nagios/files/nagios-4.x/nagios-4.0.6/nagios-4.0.6.tar.gz
Resolviendo sourceforge.net (sourceforge.net)... 216.34.181.60
Conectando con sourceforge.net (sourceforge.net)[216.34.181.60]:80... conectado.
Petición HTTP enviada, esperando respuesta... 301 Moved Permanently
Localización: https://sourceforge.net/projects/nagios/files/nagios-4.x/nagios-4.0.6/nagios-4.0.6.tar.gz [siguiendo]
--2017-03-28 08:50:10-- https://sourceforge.net/projects/nagios/files/nagios-4.x/nagios-4.0.6/nagios-4.0.6.tar.gz
Conectando con sourceforge.net (sourceforge.net)[216.34.181.60]:443... conectado.
```

Seguidament instal·larem els paquets de plugins de nagios utilitzant la següent comanda:

Wget <http://nagios-plugins.org/download/nagios-plugins-2.0.tar.gz>

```
root@debian:~# wget http://nagios-plugins.org/download/nagios-plugins-2.0.tar.gz
--2017-03-28 08:51:12-- http://nagios-plugins.org/download/nagios-plugins-2.0.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]:80.
.. conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 2501847 (2,4M) [application/x-gzip]
Grabando a: "nagios-plugins-2.0.tar.gz"

100%[=====>] 2.501.847 1,37M/s en 1,7s

2017-03-28 08:51:15 (1,37 MB/s) - "nagios-plugins-2.0.tar.gz" guardado [2501847/2501847]
```

Descomprimirem el paquet de nagios amb la següent comanda:

`tar xzf nagios-4.0.6.tar.gz`

I seguidament ens dirigirem a dins del directori creat amb la comanda `cd nagios-4.0.6`

```
root@debian:~# ls
nagios-4.0.6.tar.gz  nagios-plugins-2.0.tar.gz
root@debian:~# tar xzf nagios-4.0.6.tar.gz
root@debian:~# ls
nagios-4.0.6  nagios-4.0.6.tar.gz  nagios-plugins-2.0.tar.gz
root@debian:~# cd nagios-4.0.6
root@debian:~/nagios-4.0.6# ls
base          doxy.conf      Makefile.in    tap
cgi           functions      make-tarball    test
Changelog     html           mkpackage       THANKS
common        include        module          t-tap
config.guess  indent-all.sh nagios.spec      update-version
config.sub    indent.sh      nagios.sysconfig UPGRADING
configure     INSTALLING     pkginfo.in      worker
configure.in  install-sh     README          xdata
contrib       LEGAL          sample-config
daemon-init.in lib             subst.in
docs          LICENSE        t
root@debian:~/nagios-4.0.6#
```

Executarem l'script de configuració, passant-lo per el grup que hem creat (nagcmd)

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

```
root@debian:~/nagios-4.0.6# ./configure --with-command-group=nagcmd
```

A la captura de sota podem veure el final de la comanda, en la qual podem observar informació de la configuració.

```
*** Configuration summary for nagios 4.0.6 04-29-2014 ***:

General Options:
-----
    Nagios executable:  nagios
    Nagios user/group:  nagios,nagios
    Command user/group: nagios,nagcmd
    Event Broker:       yes
    Install ${prefix}:  /usr/local/nagios
    Install ${includedir}: /usr/local/nagios/include/nagios
    Lock file:          ${prefix}/var/nagios.lock
    Check result directory: ${prefix}/var/spool/checkresults
    Init directory:     /etc/init.d
    Apache conf.d directory: /etc/apache2/conf.d
    Mail program:       /usr/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/sbin/traceroute

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

Terminal (como superusuario) .6#
```

Ara compilarem el codi font amb la següent comanda:

make all

```
root@debian:~/nagios-4.0.6# make all
cd ./base && make
make[1]: se ingresa al directorio `/root/nagios-4.0.6/base'
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o nagios.o nagios.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o broker.o broker.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o nebmods.o nebmods.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o ../common/shared.o
../common/shared.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o nerd.o nerd.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o query-handler.o query-handler.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o workers.o workers.c
gcc -Wall -I.. -g -O2 -DHAVE_CONFIG_H -DNSCORE -c -o checks.o checks.c
```

Seguidament instal·larem els binaris, els scripts d'inici i els arxius d'exemple utilitzant les següents comandes:

make install-init

```
root@debian:~/nagios-4.0.6# make install-init
/usr/bin/install -c -m 755 -d -o root -g root /etc/init.d
/usr/bin/install -c -m 755 -o root -g root daemon-init /etc/init.d/nagios
*** Init script installed ***
```


make install-config

```

root@debian:~/nagios-4.0.6# 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.c
fg /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/template
-object/commands.cfg /usr/local/nagios/etc/objects/commands.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template
-object/contacts.cfg /usr/local/nagios/etc/objects/contacts.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template
-object/timeperiods.cfg /usr/local/nagios/etc/objects/timeperiods.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template
-object/localhost.cfg /usr/local/nagios/etc/objects/localhost.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template
-object/windows.cfg /usr/local/nagios/etc/objects/windows.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template
-object/printer.cfg /usr/local/nagios/etc/objects/printer.cfg
/usr/bin/install -c -b -m 664 -o nagios -g nagios sample-config/template
-object/switch.cfg /usr/local/nagios/etc/objects/switch.cfg

*** Config files installed ***

```

make install-commandmode

```

root@debian:~/nagios-4.0.6# make install-commandmode
/usr/bin/install -c -m 775 -o nagios -g nagcmd -d /usr/local/nagios/var/
rw
chmod g+s /usr/local/nagios/var/rw

*** External command directory configured ***

```

A la captura de sota podem veure que els arxius de configuració els tenim a dins de */usr/local/nagios/etc/objects*

```
root@debian:/usr/local/nagios/etc/objects# ls
commands.cfg    localhost.cfg    switch.cfg       timeperiods.cfg
commands.cfg~   localhost.cfg~   switch.cfg~      timeperiods.cfg~
contacts.cfg     printer.cfg      templates.cfg    windows.cfg
contacts.cfg~   printer.cfg~     templates.cfg~   windows.cfg~
root@debian:/usr/local/nagios/etc/objects#
```

De moment només editarem el fitxer *contacts.cfg* afegint el nostre correu electrònic. Utilitzarem la següent comanda per editar el fitxer:

```
nano /usr/local/nagios/etc/objects/contacts.cfg
```

```
root@debian:/usr/local/nagios/etc/objects# nano contacts.cfg
```

A la imatge de sota, utilitzant la comanda *cat contacts.cfg*, podem veure el contingut del fitxer *contacts.cfg* un cop modificat.

```
#####
#####
#
# CONTACTS
#
#####
#####

# Just one contact defined by default - the Nagios admin (that's you)
# This contact definition inherits a lot of default values from the 'generic-contact'
# template which is defined elsewhere.
█
define contact{
    contact_name      nagiosadmin          ; Short name of user
    use                generic-contact      ; Inherit default values from generic-contact template (defined above)
    alias              Nagios Admin         ; Full name of user

    email              marcviol@outlook.es ; <***** CHANGE THIS TO YOUR EMAIL ADDRESS *****>
}

```

Assignarem a l'usuari nagiosadmin a la interfície web de nagios. Ens demanarà una contrasenya, la qual és *linux*. Ho farem utilitzant la comanda:

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

```
root@debian:~/nagios-4.0.6# htpasswd -c /usr/local/nagios/etc/htpasswd.u
sers nagiosadmin
New password:
Re-type new password:
Adding password for user nagiosadmin
root@debian:~/nagios-4.0.6#
```

Seguidament reiniciarem el servei apache utilitzant la comanda:

```
/etc/init.d/apache2 reload
```

```
root@debian:~/nagios-4.0.6# /etc/init.d/apache2 reload
[....] Reloading web server config: apache2
apache2: Could not reliably d
etermine the server's fully qualified domain name, using 127.0.1.1 for S
erverName
. ok
root@debian:~/nagios-4.0.6#
```

Seguidament accedirem a la carpeta creada en el procés amb *cd nagios-plugins-2.0*

```
root@debian:~# ls
nagios-4.0.6  nagios-4.0.6.tar.gz  nagios-plugins-2.0.tar.gz
root@debian:~# tar xzf nagios-plugins-2.0.tar.gz
root@debian:~# ls
nagios-4.0.6          nagios-plugins-2.0
nagios-4.0.6.tar.gz  nagios-plugins-2.0.tar.gz
root@debian:~# cd nagios-plugins-2.0
root@debian:~/nagios-plugins-2.0#
```

Ara compilarem i instal·larem els plugins utilitzant les comandes:

`./configure --with-nagios-user=nagios --with-nagios-group=nagios`

```
root@debian:~/nagios-plugins-2.0# ./configure --with-nagios-user=nagios --with-nagios-group=nagios

config.status: creating plugins-scripts/Makefile
config.status: creating plugins-scripts/subst
config.status: creating plugins-scripts/utils.pm
config.status: creating plugins-scripts/utils.sh
config.status: creating perlmods/Makefile
config.status: creating test.pl
config.status: creating pkg/solaris/pkginfo
config.status: creating po/Makefile.in
config.status: creating config.h
config.status: executing depfiles commands
config.status: executing libtool commands
config.status: executing po-directories commands
config.status: creating po/POTFILES
config.status: creating po/Makefile
  --with-apt-get-command: /usr/bin/apt-get
  --with-ping6-command: /bin/ping6 -n -U -w %d -c %d %s
  --with-ping-command: /bin/ping -n -U -w %d -c %d %s
    --with-ipv6: yes
    --with-mysql: no
    --with-openssl: no
    --with-gnutls: no
  --enable-extra-opts: yes
    --with-perl: /usr/bin/perl
  --enable-perl-modules: no
    --with-cgiurl: /nagios/cgi-bin
  --with-trusted-path: /bin:/sbin:/usr/bin:/usr/sbin
  --enable-libtap: no
root@debian:~/nagios-plugins-2.0# ./configure --with-nagios-user=nagios --with-nagios-group=nagios
```

Podem veure que la comanda es troba el final ja que l'execució ha tallat el principi.

make

```
root@debian:~/nagios-plugins-2.0# make
```

A la següent captura també podem veure la comanda a sota, ja que l'execució d'ella ens ha tallat el principi.

make install

```
root@debian:~/nagios-plugins-2.0# make install
Making install in gl
make[1]: se ingresa al directorio `/root/nagios-plugins-2.0/gl'
make install-recursive
make[2]: se ingresa al directorio `/root/nagios-plugins-2.0/gl'
make[3]: se ingresa al directorio `/root/nagios-plugins-2.0/gl'
make[4]: se ingresa al directorio `/root/nagios-plugins-2.0/gl'
if test yes = no; then \
    case 'linux-gnu' in \
        darwin[56]*) \
            need_charset_alias=true ;; \
        darwin* | cygwin* | mingw* | pw32* | cegcc*) \
            need_charset_alias=false ;; \
        *) \
            need_charset_alias=true ;; \
    esac ; \
else \
    need_charset_alias=false ; \
fi ; \
if $need_charset_alias; then \
    /bin/bash ../build-aux/mkinstalldirs /usr/local/nagios/lib ; \
fi ; \
if test -f /usr/local/nagios/lib/charset.alias; then \
    sed -f ref-add.sed /usr/local/nagios/lib/charset.alias > /usr/
local/nagios/lib/charset.tmp ; \
    /usr/bin/install -c -o nagios -g nagios -m 644 /usr/local/nagi
os/lib/charset.tmp /usr/local/nagios/lib/charset.alias ; \
```

Ara iniciarem el nagios. Primer crearem un enllaç simbòlic amb la comanda *ln*, perquè d'aquesta manera s'iniciï el servei un cop engeguem la nostra màquina.

Hem utilitzat la següent comanda per fer-ho:

`ln -s /etc/init.d/nagios /etc/rcS.d/S99nagios`

```
root@debian:~/nagios-plugins-2.0# ln -s /etc/init.d/nagios /etc/rcS.d/S9
9nagios
root@debian:~/nagios-plugins-2.0# █
```

Finalment comprovarem si tota la configuració que hem fet fins ara és correcta utilitzant la comanda:

```
/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
```

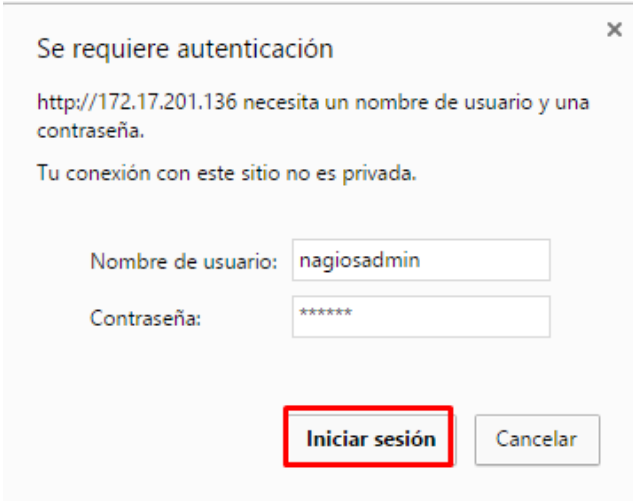
```
root@debian:~/nagios-plugins-2.0# /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
```

Si ens indica que no hi ha ningun error i que la configuració és correcta ja podrem iniciar el nagios utilitzant la comanda:

```
/etc/init.d/nagios start
```

```
root@debian:~/nagios-plugins-2.0# /etc/init.d/nagios start
Starting nagios: done.
```

Ara ja podrem accedir des del navegador al nagios posant la IP del servidor /nagios, que en el nostre cas seria 172.17.201.136/nagios. Ens demanarà un usuari i una contrasenya que és el que hem configurat anteriorment, la introduïrem i ja estarem a dins.



Se requiere autenticación

http://172.17.201.136 necesita un nombre de usuario y una contraseña.

Tu conexión con este sitio no es privada.

Nombre de usuario:

Contraseña:

Iniciar sesión Cancelar

A la imatge de sota podem veure una captura un cop hem accedit al nagios des del navegador.



Seguidament procedirem a connectar el Nagios amb un client Linux amb el plugin NRPE.

Primer de tot descarregarem l'ad-on en el servidor per poder controlar un client Linux amb la comanda següent:

Wget <http://sourceforge.net/projects/nagios/files/nrpe-2.x/nrpe-2.15/nrpe-2.15.tar.gz>

```
root@debian:~# wget http://sourceforge.net/projects/nagios/files/nrpe-2.x/nrpe-2.15/nrpe-2.15.tar.gz
--2017-03-28 09:56:00-- http://sourceforge.net/projects/nagios/files/nrpe-2.x/nrpe-2.15/nrpe-2.15.tar.gz
Resolviendo sourceforge.net (sourceforge.net)... 216.34.181.60
Conectando con sourceforge.net (sourceforge.net)[216.34.181.60]:80... conectado.
Petición HTTP enviada, esperando respuesta... 301 Moved Permanently
Localización: https://sourceforge.net/projects/nagios/files/nrpe-2.x/nrpe-2.15/nrpe-2.15.tar.gz [siguiendo]
--2017-03-28 09:56:00-- https://sourceforge.net/projects/nagios/files/nrpe-2.x/nrpe-2.15/nrpe-2.15.tar.gz
Conectando con sourceforge.net (sourceforge.net)[216.34.181.60]:443... conectado.
Petición HTTP enviada, esperando respuesta... 302 Found
Localización: https://sourceforge.net/projects/nagios/files/nrpe-2.x/nrpe-2.15/nrpe-2.15.tar.gz/download [siguiendo]
--2017-03-28 09:56:10-- https://sourceforge.net/projects/nagios/files/nrpe-2.x/nrpe-2.15/nrpe-2.15.tar.gz/download
```

Una vegada hem descarregat l'ad-on, el descomprimirem utilitzant la següent comanda:

tar xvf nrpe-2.15.tar.gz

```
nrpe-2.15/nrpe.spec.in
nrpe-2.15/package/
nrpe-2.15/package/solaris/
nrpe-2.15/package/solaris/Makefile.in
nrpe-2.15/package/solaris/pkg/
nrpe-2.15/package/solaris/pkg/i.config
nrpe-2.15/package/solaris/pkg/nrpe
nrpe-2.15/package/solaris/pkg/nrpe.xml
nrpe-2.15/package/solaris/pkg/postinstall
nrpe-2.15/package/solaris/pkg/preinstall
nrpe-2.15/package/solaris/pkg/r.config
nrpe-2.15/sample-config/
nrpe-2.15/sample-config/nrpe.cfg.in
nrpe-2.15/sample-config/nrpe.xinetd.in
nrpe-2.15/src/
nrpe-2.15/src/Makefile.in
nrpe-2.15/src/acl.c
nrpe-2.15/src/check_nrpe.c
nrpe-2.15/src/nrpe.c
nrpe-2.15/src/snprintf.c
nrpe-2.15/src/utls.c
nrpe-2.15/subst.in
nrpe-2.15/update-version
root@debian:~# tar xvf nrpe-2.15.tar.gz
```

Seguidament executarem l'script de configuració, que es troba a dins del directori `~/nrpe-2.15` utilitzant la comanda següent:

`.configure`

```
root@debian:~/nrpe-2.15# ./configure
checking for a BSD-compatible install... /usr/bin/install -c
checking build system type... i686-pc-linux-gnu
checking host system type... i686-pc-linux-gnu
```

A la imatge de sota podem veure que en el final de l'script de configuració ens dona un error.


```

checking for long... yes
checking size of long... 4
checking for uint32_t... yes
checking for u_int32_t... yes
checking for int32_t... yes
checking for va_copy... yes
checking for vsnprintf... yes
checking for snprintf... yes
checking for asprintf... yes
checking for vasprintf... yes
checking for C99 vsnprintf... yes
checking for getopt_long... yes
checking for main in -lnsl... yes
checking for socket in -lsocket... no
checking for main in -lwrap... no
checking for strdup... yes
checking for strstr... yes
checking for strtoul... yes
checking for initgroups... yes
checking for closesocket... no
checking for socklen_t... yes
checking for type of socket size... size_t
checking for SSL headers... configure: error: Cannot find ssl headers
root@debian:~/nrpe-2.15#

```

Per poder evitar aquest error tindrem que instal·lar un paquet anomenat *libssl-dev* per tal de poder evitar-lo. Per instal·lar el paquet utilitzarem la següent comanda:

`apt-get install libssl-dev`

```

root@debian:~/nrpe-2.15# apt-get install libssl-dev
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes extras:
  libssl-doc libssl1.0.0
Se instalarán los siguientes paquetes NUEVOS:
  libssl-dev libssl-doc
Se actualizarán los siguientes paquetes:
  libssl1.0.0
1 actualizados, 2 se instalarán, 0 para eliminar y 202 no actualizados.
Necesito descargar 5.924 kB de archivos.
Se utilizarán 6.827 kB de espacio de disco adicional después de esta operación.
¿Desea continuar [S/n]?

```

Tot seguit, tornarem a executar l'script de configuració que es troba en el directori `~/nrpe-2.15` amb la comanda:

`./configure --with-ssl=/usr/bin/openssl --with-ssl-lib=/usr/lib/x86_64-linux-gnu`

```

root@debian:~/nrpe-2.15# ./configure --with-ssl=/usr/bin/openssl --with
ssl-lib=/usr/lib/x86_64-linux-gnu
checking for a BSD-compatible install... /usr/bin/install -c
checking build system type... i686-pc-linux-gnu
checking host system type... i686-pc-linux-gnu
checking for gcc... gcc
checking for C compiler default output file name... a.out
checking whether the C compiler works... yes
checking whether we are cross compiling... no
checking for suffix of executables...
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
checking for gcc option to accept ANSI C... none needed
checking whether make sets $(MAKE)... yes
checking how to run the C preprocessor... gcc -E
checking for egrep... grep -E
checking for ANSI C header files... yes

```

A la imatge de sota podem veure que efectivament, aquesta vegada l'script de configuració no ens dona l'error i acaba correctament.

```

*** Configuration summary for nrpe 2.15 09-06-2013 ***:

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

Review the options above for accuracy.  If they look okay,
type 'make all' to compile the NRPE daemon and client.

root@debian:~/nrpe-2.15# ./configure --with-ssl=/usr/bin/openssl --with-
ssl-lib=/usr/lib/i386-linux-gnu/

```

Ara compilarem (estant dins del directori `~/nrpe-2.15` amb la següent comanda:

```
make all
```

```

root@debian:~/nrpe-2.15# make all
cd ./src/; make ; cd ..
make[1]: se ingresa al directorio `/root/nrpe-2.15/src'
gcc -g -O2 -I/usr/include/openssl -I/usr/include -DHAVE_CONFIG_H -I ../include -I ../../include -o nrpe ./nrpe.c ./utils.c ./acl.c -L/usr/lib/i386-linux-gnu/ -lssl -lcrypto -lnsl
gcc -g -O2 -I/usr/include/openssl -I/usr/include -DHAVE_CONFIG_H -I ../include -I ../../include -o check_nrpe ./check_nrpe.c ./utils.c -L/usr/lib/i386-linux-gnu/ -lssl -lcrypto -lnsl
make[1]: se sale del directorio `/root/nrpe-2.15/src'

*** Compile finished ***

If the NRPE daemon and client compiled without any errors, you
can continue with the installation or upgrade process.

Read the PDF documentation (NRPE.pdf) for information on the next
steps you should take to complete the installation or upgrade.

root@debian:~/nrpe-2.15#

```

Finalment instal·larem el plugin utilitzant la comanda de sota:

make install-plugin

```

root@debian:~/nrpe-2.15# make install-plugin
cd ./src/ && make install-plugin
make[1]: se ingresa al directorio `/root/nrpe-2.15/src'
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/libexec
/usr/bin/install -c -m 775 -o nagios -g nagios check_nrpe /usr/local/nagios/libexec
make[1]: se sale del directorio `/root/nrpe-2.15/src'

```

Un cop hem instal·lat el plugin en el servidor, procedirem a fer el mateix en el client.

Amb el *SERVER-WEB*, abans d'instal·lar res, primer de tot crearem un usuari anomenat nagios i li assignarem una contrasenya igual que el nom. Ho farem amb les comandes:

useradd nagios

passwd nagios

```
root@debian:/home/usuario# useradd nagios
root@debian:/home/usuario# passwd nagios
Introduzca la nueva contraseña de UNIX:
Vuelva a escribir la nueva contraseña de UNIX:
passwd: contraseña actualizada correctamente
root@debian:/home/usuario#
```

Seguidament descarregarem el el plugin *nagios-plugin* utilitzant la comanda de sota:

wget <http://nagios-plugins.org/download/nagios-plugins-2.0.3.tar.gz>

```
root@debian:/home/usuario# wget http://nagios-plugins.org/download/nagios-plugins-2.0.3.tar.gz
```

Un cop descarregat el paquet, procedirem a descomprimir-lo utilitzant la comanda següent:

tar xvfz nagios-plugins-2.0.3.tar.gz

```

usuari@debian: ~
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
root@debian:/home/usuari# tar xvfz nagios-plugins-2.0.3.tar.gz

```

Ens dirigirem al directori *nagios-plugins-2.0.3* amb *cd* i seguidament compilarem amb la comanda de sota:

```
export LDFLAGS=-ldl
```

```
./configure --with-nagios-user=nagios --with-nagios-group=nagios --enable-redhat-
pthread-workaround
```

```

root@debian:/home/usuari/nagios-plugins-2.0.3# export LDFLAGS=-ldl
root@debian:/home/usuari/nagios-plugins-2.0.3# ./configure --with-na
nagios --with-nagios-group=nagios --enable-redhat-pthread-workaround
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /bin/mkdir -p
checking for gawk... no
checking for mawk... mawk
checking whether make sets $(MAKE)... yes
checking whether to disable maintainer-specific portions of Makefiles
checking build system type... █

```

Tot seguit compilarem utilitzant la comanda *make*.

```
-install -lresolv -lpinread -ldl
make[2]: se sale del directorio `/nagios-plugins-2.1.4/plug
Making all in po
make[2]: se ingresa al directorio `/nagios-plugins-2.1.4/pc
make[2]: No se hace nada para `all'.
make[2]: se sale del directorio `/nagios-plugins-2.1.4/po'
make[2]: se ingresa al directorio `/nagios-plugins-2.1.4'
make[2]: se sale del directorio `/nagios-plugins-2.1.4'
make[1]: se sale del directorio `/nagios-plugins-2.1.4'
root@debian:/nagios-plugins-2.1.4# make
```

Acabarem d'instal·lar utilitzant la comanda *make install*.

```
usuario@debian: ~
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
root@debian:/nagios-plugins-2.1.4# make install
```

Ara canviarem els propietaris de les carpetes de nagios en el *SERVER-WEB* utilitzant les comandes:

```
chown nagios.nagios /usr/local/nagios
```

```
chown -R nagios.nagios /usr/local/nagios/libexec/
```

```
root@debian:/nagios-plugins-2.1.4# chown nagios.nagios /usr/local/nagios
root@debian:/nagios-plugins-2.1.4# chown -R nagios.nagios /usr/local/nagios/libexec
```

Seguidament procedirem a descarregar l'add-on NRPE en el *SERVER-WEB* amb la comanda:

wget <http://sourceforge.net/projects/nagios/files/nrpe-2.x/nrpe-2.15/nrpe-2.15.tar.gz>

```

usuari@debian: ~
Archivo Editar Ver Buscar Terminal Ayuda
root@debian:/nagios-plugins-2.1.4# cd
root@debian:~# wget http://sourceforge.net/projects/nagios/files/nrpe-2.x/nrpe-2.15/nrpe-2.15.tar.gz

```

Ara descomprimim el paquet que acabem de descarregar en el *SERVER-WEB* amb la comanda:

tar xvfz nrpe-2.15.tar.gz

```

nagios-plugins-2.0.3/      nrpe-2.15.tar.gz
nagios-plugins-2.0.3.tar.gz
root@debian:~# tar xvfz nrpe-2.15.tar.gz

```

Ens dirigirem al directori *nrpe-2.15* amb la comanda *cd* i compilarem el codi font utilitzant la comanda:

./configure

```

usuari@debian
Archivo Editar Ver Buscar Terminal Ayuda
root@debian:~# cd nrpe-2.15
root@debian:~/nrpe-2.15# ./configure

```

Ens donarà un error, així que tindrem que instal·lar les llibreries SSL (les mateixes que hem instal·lat en el *SERVER-MONITOR* prèviament) per poder evitar-lo amb la comanda:

apt-get install libssl-dev

```

Archivo  Editar  Ver  Buscar  Terminal  Ayuda
root@debian:~/nrpe-2.15# apt-get install libssl-dev
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... 0%

```

Ara tornarem a fer el configure indicant-li les llibreries. La comanda quedaria així:

`./configure --with-ssl=/usr/bin/openssl --with-ssl-lib=/usr/lib/i386-linux-gnu`

```

debian:~/nrpe-2.15# ./configure --with-ssl=/usr/bin/openssl --with-ssl-lib=/usr/lib/i386-linux-gnu

```

Seguidament compilarem utilitzant la comanda:

`make all`

```

Archivo  Editar  Ver  Buscar  Terminal  Ayuda
root@debian:~/nrpe-2.15# make all
cd ./src/; make ; cd ..
make[1]: se ingresa al directorio `/root/nrpe-2.15/s
cc -g -O2 -I/usr/include/openssl -I/usr/include -Df
./../include -o nrpe ./nrpe.c ./utils.c ./acl.c -l
-lssl -lcrypto -lnsl

```

Seguirem compilant utilitzant la comanda

`make install-plugin`


```

Archivo  Editar  Ver  Buscar  Terminal  Ayuda
root@debian:~/nrpe-2.15# make install-plugin
d ./src/ && make install-plugin
make[1]: se ingresa al directorio `/root/nrpe-2
usr/bin/install -c -m 775 -o nagios -g nagios
usr/bin/install -c -m 775 -o nagios -g nagios

```

Ara instal·larem el dimoni xinetd utilitzant la comanda:

`apt-get install xinetd`

```

iones  Lugares
a documentos, carpetas y lugares en la red io@debian: ~
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
root@debian:~/nrpe-2.15# apt-get install xinetd
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes NUEVOS:
  xinetd
Se descargados, 1 se instalarán, 0 para eliminar y 218 no
necesito descargar 146 kB de archivos.
Se utilizarán 252 kB de espacio de disco adicional después
es:1 http://ftp.es.debian.org/debian/ wheezy/main xinetd
1 [146 kB]
Se descargados 146 kB en 5seg. (28,0 kB/s)
Seleccionando el paquete xinetd previamente no selecciona
Leyendo la base de datos ... 145038 ficheros o directorio
e.)
Se empaquetando xinetd (de .../xinetd_1%3a2.3.14-7.1+deb7u
Procesando disparadores para man-db ...
Configurando xinetd (1:2.3.14-7.1+deb7u1) ...
ok ] Stopping internet superserver: xinetd.
ok ] Starting internet superserver: xinetd.
root@debian:~/nrpe-2.15#

```

Seguidament editarem l'arxiu *nrpe* que es troba a dins de */etc/xinetd.d* per poder configurar l'NRPE en el dimoni del client afegint la IP del servidor nagios en la línia *only_from* per admetre les seves connexions.

```

Archivo  Editar  Ver  Buscar  Terminal  Ayuda
GNU nano 2.2.6  Fichero: /etc/xinetd.d/nrpe
default: on
description: NRPE (Nagios Remote Plugin Executor)
service nrpe
    flags            = REUSE
    socket_type      = stream
    port             = 5666
    wait             = no
    user             = nagios
    group            = nagios
    server           = /usr/local/nagios/bin/nrpe
    server_args      = -c /usr/local/nagios/etc/nrpe.cfg --inetd
    log_on_failure   += USERID
    disable          = no
    only_from        = 127.0.0.1 172.17.201.136

```

Ara ens dirigirem a l'arxiu `/etc/services` i al final del fitxer afegirem la línia:

`nrpe 5666/tcp # NRPE`

```

usuari@debian: ~
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
GNU nano 2.2.6  Fichero: /etc/services
xpidot      15345/udp
sgi-cmsd    17001/udp      # Cluster
sgi-crsd    17002/udp
sgi-gcd     17003/udp      #
sgi-cad     17004/tcp      #
isdnlod     20011/tcp      #
isdnlod     20011/udp
vboxd       20012/tcp      #
vboxd       20012/udp
binkp       24554/tcp      #
asp         27374/tcp      #
asp         27374/udp
csync2      30865/tcp      #
dirproxy    57000/tcp      #
tfido       60177/tcp      #
fido        60179/tcp      #
# Local services
nrpe 5666/tcp # NRPE

```

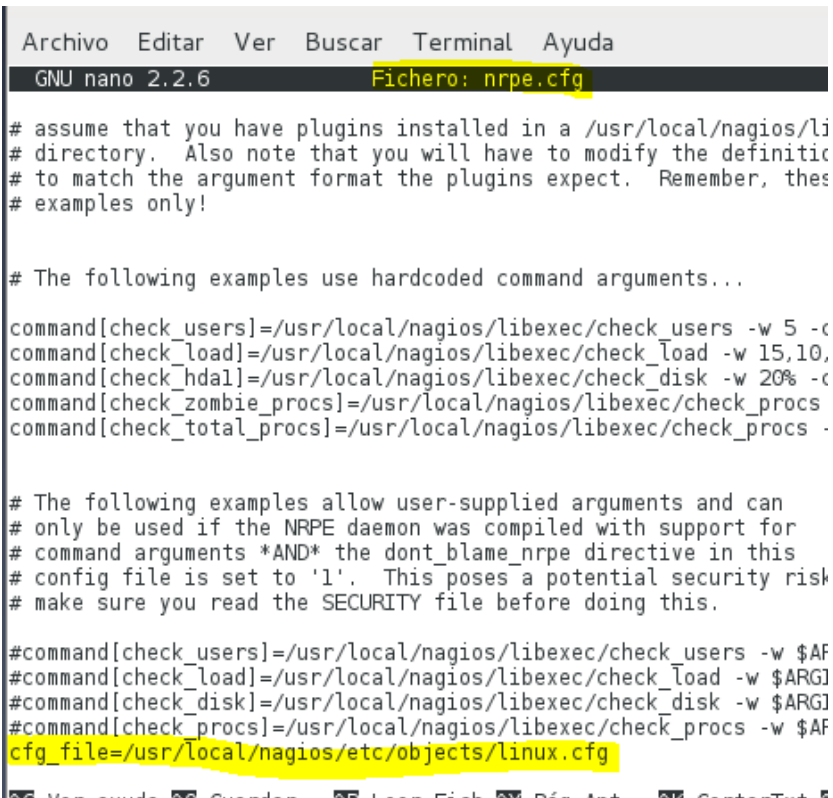
Accedim a dins de la carpeta `/usr/local/nagios` i podem observar com hi tenim l'arxiu "nrpe.cfg" que hem de modificar

Cd /usr/local/nagios

```
root@debian:~# cd /usr/local/nagios/
root@debian:/usr/local/nagios# ls
bin  etc  include  libexec  nrpe-2.15.tar.gz  s
root@debian:/usr/local/nagios# cd etc
root@debian:/usr/local/nagios/etc# ls
nrpe.cfg
root@debian:/usr/local/nagios/etc#
```

Procedim a editar l'arxiu "nrpe.cfg" i afegim la línia:

cfg_file=/usr/local/nagios/etc/objects/linux.cfg



```

Archivo  Editar  Ver  Buscar  Terminal  Ayuda
GNU nano 2.2.6  Fichero: nrpe.cfg

# assume that you have plugins installed in a /usr/local/nagios/li
# directory. Also note that you will have to modify the definitio
# to match the argument format the plugins expect. Remember, thes
# examples only!

# The following examples use hardcoded command arguments...

command[check_users]=/usr/local/nagios/libexec/check_users -w 5 -c
command[check_load]=/usr/local/nagios/libexec/check_load -w 15,10,
command[check_hdall]=/usr/local/nagios/libexec/check_disk -w 20% -c
command[check_zombie_procs]=/usr/local/nagios/libexec/check_procs
command[check_total_procs]=/usr/local/nagios/libexec/check_procs -

# 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
# make sure you read the SECURITY file before doing this.

#command[check_users]=/usr/local/nagios/libexec/check_users -w $AF
#command[check_load]=/usr/local/nagios/libexec/check_load -w $ARG1
#command[check_disk]=/usr/local/nagios/libexec/check_disk -w $ARG1
#command[check_procs]=/usr/local/nagios/libexec/check_procs -w $AF
cfg_file=/usr/local/nagios/etc/objects/linux.cfg

```

Perquè pugui agafar el "WEB-SERVER", hem de buscar la línia on posa "allowed_hosts", i afegir la IP del web

```
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
GNU nano 2.2.6  Fichero: nrpe.cfg

# (i.e. 192.168.1.0/24) are also supported. Hostname wildcards are
# supported.
#
# Note: The daemon only does rudimentary checking of the client's IP
# address. I would highly recommend adding entries in your /etc/hosts
# 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 inet
allowed_hosts=127.0.0.1 172.17.201.136

# COMMAND ARGUMENT PROCESSING
# This option determines whether or not the NRPE daemon will allow
# to specify arguments to commands that are executed. This option
```

♣ Disposarà d'una IP pròpia.

```

RX bytes:0 (0.0 B)  TX bytes:571727 (558.3 KiB)
Interrupt:19 Base address:0x2024

eth1      Link encap:Ethernet  HWaddr 00:0c:29:61:70:52
          inet addr:172.17.201.136  Bcast:172.17.255.255  Mask:255.255.0
          .0
          inet6 addr: fe80::20c:29ff:fe61:7052/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:577086 errors:1 dropped:0 overruns:0 frame:0
          TX packets:5988 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:73256912 (69.8 MiB)  TX bytes:940414 (918.3 KiB)
          Interrupt:19 Base address:0x2080

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:2354 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2354 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:923595 (901.9 KiB)  TX bytes:923595 (901.9 KiB)

```

L'arxiu de configuració del host de nagios es troba a `/usr/local/nagios/etc/objects` i anomenat en el meu cas `Linux.cfg`. Aquest contindrà informació sobre la màquina a monitoritzar. Cada cop que es fa un canvi es reiniciarà el servei de nagios amb la comanda: `service nagios restart`. Les línies amb les quals aconseguirem presentar el nostre host en el servidor nagios, són les següents:

```
define host{
```

<i>name</i>	<i>SERVER-WEB</i>
<i>use</i>	<i>generic-host</i>
<i>check_period</i>	<i>24x7</i>
<i>check_interval</i>	<i>5</i>
<i>retry_interval</i>	<i>1</i>
<i>max_check_attempts</i>	<i>10</i>
<i>check_command</i>	<i>check-host-alive</i>
<i>notification_period</i>	<i>24x7</i>
<i>notification_interval</i>	<i>30</i>
<i>notification_options</i>	<i>d,r</i>
<i>contact_groups</i>	<i>admins</i>

```

    register                                0 ; ES UNA PLANTILLA
}
define host{
    use                                    SERVER-WEB
    host_name                             debian 7
    alias                                 Debian 7
    address                               192.168.1.132
}

```

Seguidament, en cada objecte a monitoritzar, plantejarem les següents línies que formaran l'arxiu de configuració del servidor web.

```

GNU nano 2.2.6      Fichero: commands.cfg

# 'process-service-perfdata' command definition
define command{
    command_name    process-service-perfdata
    command_line    /usr/bin/printf "%b" "$LASTSERVICECHECK$\t$HOST$
}

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

```

[^]G Ver ayuda [^]O Guardar [^]R Leer Fich [^]Y Pág Ant [^]K CortarTxt [^]C Pos actua
[^]X Salir [^]J Justifica [^]W Buscar [^]V Pág Sig [^]U PegarTxt [^]T Ortografí

Anem al directori `/usr/local/nagios/libexec`, i fem un `ls` per poder veure el que tenim dintre.

```
root@debian:/usr/local/nagios/etc/objects# cd /usr/local/nagios/libexec/
root@debian:/usr/local/nagios/libexec# ls
check_apd          check_ide_smart    check_ntp          check_ssmtp
check_breeze       check_ifoperstatus check_ntp_peer     check_swap
check_by_ssh       check_ifstatus     check_ntp_time     check_tcp
check_clamd        check_imap         check_nwstat       check_time
check_cluster      check_ircd         check_oracle       check_udp
check_dhcp         check_jabber       check_overcr       check_ups
check_dig          check_load         check_ping         check_uptime
check_disk         check_log          check_pop          check_users
check_disk_smb     check_mailq        check_procs        check_wave
check_dns          check_mrtg         check_real         negate
check_dummy        check_mrtgtraf     check_rpc          urlize
check_file_age     check_nagios       check_sensors      utils.pm
check_flexlm       check_nntp         check_simap        utils.sh
check_ftp          check_nntp         check_smtp
check_nttp         check_nrpe         check_spop
check_icmp         check_nt           check_ssh
root@debian:/usr/local/nagios/libexec#
```

A continuació, anem dintre del directori `/usr/local/nagios/etc/objects`, i fem un `ls` per veure el que tenim dintre d'aquest directori.

Tot seguit, modifiquem els arxius `localhost.cfg` i `linux.cfg`.

```
root@debian:~# cd /usr/local/nagios/etc/objects/
root@debian:/usr/local/nagios/etc/objects# ls
commands.cfg  localhost.cfg  switch.cfg     timeperiods.cfg
contacts.cfg  printer.cfg   templates.cfg  windows.cfg
root@debian:/usr/local/nagios/etc/objects# nano localhost.cfg
root@debian:/usr/local/nagios/etc/objects# nano linux.cfg
```

Un cop fet el pas anterior, anem a l'arxiu nagios.cfg i afegim la següent línia.

```

Terminal (como superusuario)
GNU nano 2.2.6 Fichero: nagios.cfg Modificado

#cfg_file=/usr/local/nagios/etc/objects/switch.cfg

# Definitions for monitoring a network printer
#cfg_file=/usr/local/nagios/etc/objects/printer.cfg

# 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
#cfg_dir=/usr/local/nagios/etc/printer
#cfg_dir=/usr/local/nagios/etc/switches
#cfg_dir=/usr/local/nagios/etc/routers

# OBJECT CACHE FILE

^G Ver ayuda ^O Guardar ^R Leer Fich ^Y Pág Ant ^K CortarTxt ^C Pos actual
^X Salir ^J Justificar ^W Buscar ^V Pág Sig ^U PegarTxt ^T Ortografía

root@debian:/usr/local/nagios/etc# nano nagios.cfg
root@debian:/usr/local/nagios/etc# nano nagios.cfg
root@debian:/usr/local/nagios/etc# mkdir /usr/local/nagios/etc/servers

```

Per aconseguir que el client pugi ser monitoritzat, modificarem el següent arxiu: /usr/local/nagios/etc/configuracion.cfg. Afegirem a la línia allowed_hosts, la IP de la màquina nagios:

allowed_hosts=127.0.0.1,172.17.201.142

```

GNU nano 2.2.6 Fichero: configuracion.cfg

define host{
    use linux-server ; Name of host template$
    ; This host definition $
    ; in (or inherited by) $

    host_name debian
    alias debian
    address 172.17.201.142
}

```


Podem veure en el frontal web com en la pestanya Hosts de l'esquerra trobem ara apart del nostre servidor nagios, el server-web.

Nagios®

General
Home
Documentation
Current Status
Tactical Overview
Map
Hosts
Services
Host Groups
Summary
Grid
Service Groups
Summary
Grid
Problems
Services
(Unhandled)
Hosts (Unhandled)

Current Network Status
Last Updated: Thu Mar 30 13:04:22 CEST 2017
Updated every 90 seconds
Nagios® Core™ 4.0.6 - www.nagios.org
Logged in as nagiosadmin

Host Status Totals
Up Down Unreachable Pending
2 0 0 0
All Problems All Types
0 2

Service Status Totals
Ok Warning Unknown Critical Pending
8 0 0 0 0
All Problems All Types
0 8

Host Status Details For All Host Groups

Host	Status	Last Check	Duration	Status Information
debian	UP	03-30-2017 13:03:45	0d 0h 0m 37s+	PING OK - Packet loss = 0%, RTA = 1.85 ms
localhost	UP	03-30-2017 13:00:40	0d 3h 40m 57s	PING OK - Packet loss = 0%, RTA = 0.05 ms

Results 1 - 2 of 2 Matching Hosts

Després, al final del document nagios.cfg, afegim la següent línia.

```
GNU nano 2.2.6          Fichero: nagios.cfg

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

# 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/servers/configuracion.cfg
```

Executem la següent comanda:

```
root@debian:~# /usr/local/nagios/libexec/check_swap -w 20% -c 10%
SWAP OK - 100% free (879 MB out of 879 MB) |swap=879MB;175;87;0;879
```

Seguidament, en cada objecte a monitoritzar, plantejarem les següents línies que formaran l'arxiu de configuració del servidor web.

```
GNU nano 2.2.6          Fichero: commands.cfg          Modificado
: 'check_nt' command definition
define command{
    command_name    check_nt
    command_line    $USER1$/check_nt -H $HOSTADDRESS$ -p 12489 -v $ARG1$ $ARG2$
}
: check_mem
define command {
    command_name check_mem
    command_line /usr/local/nagios/libexec/check_mem -H $HOSTADDRESS$ -c memory
}

: check_mysql
define command {
    command_name check_mysql
    command_line /usr/local/nagios/libexec/check_mysql.pl -H $HOSTADDRESS$
}

: check_http
define command {
    command_name check_http
    command_line /usr/local/nagios/libexec/check_http -H $HOSTADDRESS$ $ARG1$
}

: check_ssh
define command {
    command_name check_ssh
    command_line $USER1$/check_ssh $ARG1$ $HOSTADDRESS$
}

G Ver ayuda      ^C Guardar      ^F Leer Fich    ^Y Pág Ant      ^K CortarTxt    ^C Pos actual
X Salir          ^J Justificar   ^W Buscar      ^V Pág Sig      ^U PegarTxt     ^T Ortografia
```

Des de una màquina que no sigui 'SERVER-WEB' ni 'SERVER-MONITOR', a través del sistema de monitorització, es podrà controlar mitjançant components gràfics (icones i gràfiques) els següents elements de la màquina 'SERVER-WEB',

- El processador (gràfica temporal de la càrrega)

La comanda per monitoritzar la càrrega del processador es troba a /usr/local/nagios/libexec/ check_load. Així doncs, afegirem les línies en l'arxiu de configuració:

```
define service{
    use                               generic-service
    host_name                         SERVER-WEB
    service_description               Carrega del processador
    check_command                     check_nrpe!check_Load
}
```

- La memòria (gràfica temporal de la memòria disponible).

Per tal de monitoritzar la memòria RAM hem buscat un script per internet i l'hem afegit a la carpeta /usr/local/nagios/libexec.

Descàrrega: https://exchange.nagios.org/directory/Plugins/System-Metrics/Memory/check_mem-2Esh/details

```
define service{
use                                generic-service
host_name                        SERVER-WEB
service_description              Carrega memoria
check_command                    check_nrpe!check_mem
}
```

- El espai de disc (gràfiques temporals de l'espai disponible i l'ocupat).

Per fer-ho usarem la comanda check_disk trobada per defecte a /usr/local/nagios/libexec.

```
define service{
use                                generic-service
host_name                        SERVER-WEB
service_description              /dev/sda1 Free Space
check_command                    check_nrpe!check_disk
}
```

- El servei Apache.

Per fer-ho usarem la comanda check_http disponible dins de /usr/local/nagios/libexec.

```
define service{
use                                generic-service
host_name                        SERVER-WEB
service_description              Apache check
check_command                    check_nrpe!check_http
}
```

- El servei MySQL.

Per internet hem trobat un script per tal de monitoritzar el servei sql. L'hem guardat dins de /usr/local/nagios/libexec:

```
define service{
    use                generic-service
    host_name          SERVER-WEB
    service_description Check mysql
    check_command       check_nrpe!check_mysql.pl
}
```

El servei SSH.

Per fer-ho usarem la comanda check_ssh disponible dins de /usr/local/nagios/libexec.

```
define service{
    use                generic-service
    host_name          SERVER-WEB
    service_description SSH check
    check_command       check_nrpe!check_ssh
}
```

També afegirem a l'arxiu /usr/local/nagios/etc/objects/commands.cfg el següent per fer que les comandes funcionin.

```
# check_mem
define command {
    command_name check_mem
    command_line /usr/local/nagios/libexec/check_mem -H
$HOSTADDRESS$ -c memory
}

# check_mysql
define command {
    command_name check_mysql
```

```

        command_line /usr/local/nagios/libexec/check_mysql.pl -
H $HOSTADDRESS$
    }
    # 'check_http'
    #define command{
        #command_name check_http
        #command_line $USER1$/check_http -I $HOSTADDRESS$ $ARG1$
    #}

# check_ssh
define command{
    command_name check_ssh
    command_line $USER1$/check_ssh $ARG1$ $HOSTADDRESS$
}

```

Podem observar que, una vegada reiniciat el servei “nagios”, ens monitoritza la majoria de serveis, no ens mostra el servei “apache” ni el “mysql”

Current Network Status

Last Updated: Sat Feb 6 06:12:28 EST 2016
Updated every 90 seconds
Nagios® Core™ 4.1.1 - www.nagios.org
Logged in as nagiosadmin

[View History For This Host](#)
[View Notifications For This Host](#)
[View Service Status Detail For All Hosts](#)

Host Status Totals

Up	Down	Unreachable	Pending
1	0	0	0

All Problems: All Types

0	1
---	---

Limit Results: 100 ▼

Host	Service	Status
WEB-SERVER	C:\ Drive Space	OK
	CPU Load	OK
	Memory Usage	OK
	Uptime	OK
	Current Load	OK
	Current Users	OK
	HTTP	OK
	PING	OK
	Root Partition	OK
	SSH	OK
	Swap Usage	OK
	Total Processes	OK