

# Instalación y configuración de Naemon

Ignacio Vellido Expósito

Similarmente al objetivo de la práctica 3, se instalará Naemon en Ubuntu para monitorizarse tanto a sí mismo como a CentOS, concretamente a los servicios SSH y HTTP.

## Proceso

Comenzando con una instalación de Ubuntu y CentOS en la que SSH y HTTP funcionan correctamente, realizamos los siguientes pasos. Comenzando en Ubuntu, instalamos el servidor Naemon:

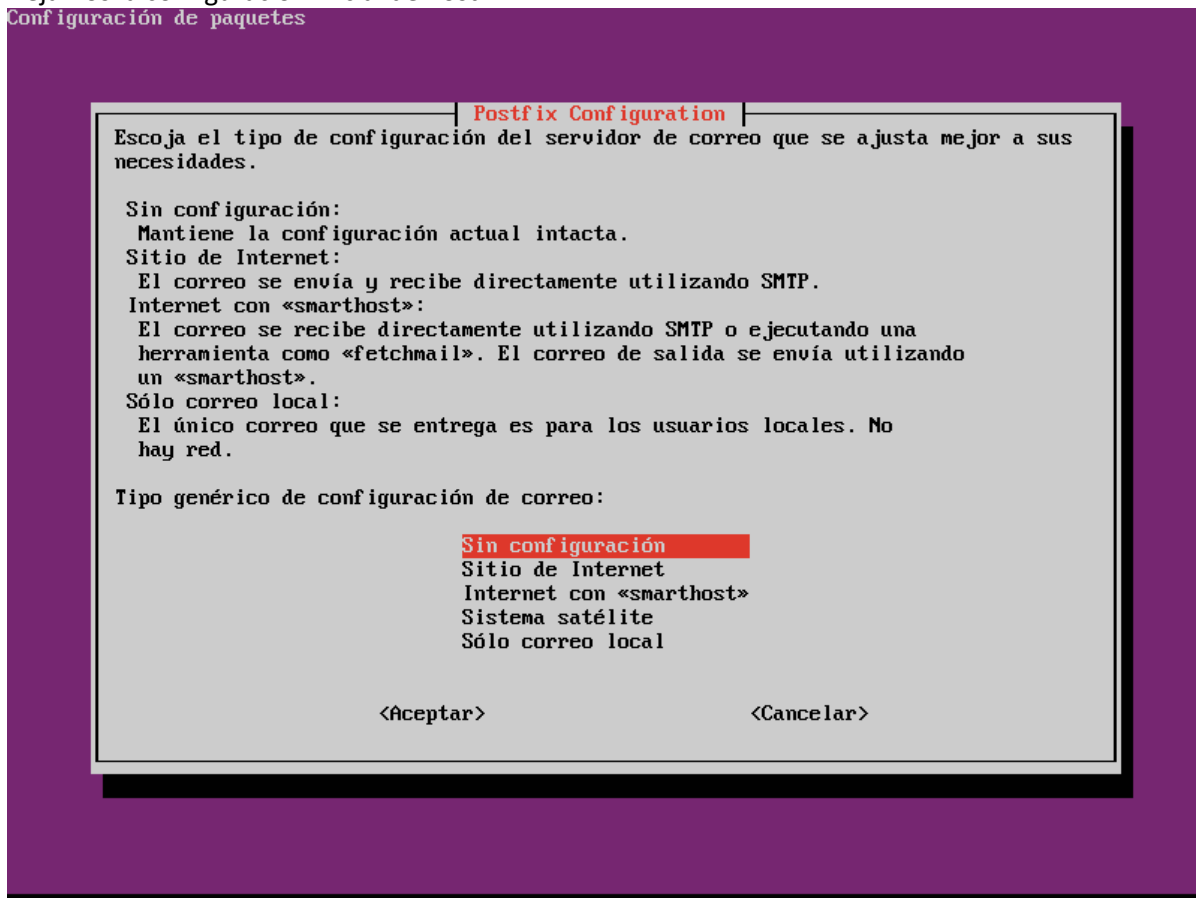
- Instalar dependencias

```
sudo apt-get install bsd-mailx apache2-utils libapache2-mod-fcgid libfontconfig1 libjpeg62 libgd3 libxpm4 xvfb libmysqlclient20
```



The screenshot shows a terminal window titled 'UbuntuSE (P2 - S3) [Running] - Oracle VM VirtualBox'. The terminal output shows the command `sudo apt-get install bsd-mailx apache2-utils libapache2-mod-fcgid libfontconfig1 libjpeg62 libgd3 libxpm4 xvfb libmysqlclient20` being executed. The window has a menu bar with 'File', 'Machine', 'View', 'Input', 'Devices', and 'Help'.

- Dejamos la configuración inicial de Postfix



- Descargamos el repositorio de Naemon

```
ive@ubuntuISEC3:~$ gpg --keyserver keys.gnupg.net --recv-keys F8C1CA08A57B9ED7
gpg: /home/ive/.gnupg: directorio creado
gpg: creado un nuevo archivo de configuración '/home/ive/.gnupg/gpg.conf'
gpg: ATENCIÓN: aún no se han activado en esta ejecución las opciones en '/home/ive/.gnupg/gpg.conf'
gpg: anillo «/home/ive/.gnupg/secring.gpg» creado
gpg: anillo «/home/ive/.gnupg/pubring.gpg» creado
gpg: solicitando clave A57B9ED7 de hkp servidor keys.gnupg.net
gpg: /home/ive/.gnupg/trustdb.gpg: se ha creado base de datos de confianza
gpg: clave A57B9ED7: clave pública "Sven Nierlein <sven.nierlein@consol.de>" importada
gpg: Cantidad total procesada: 1
gpg:      importadas: 1 (RSA: 1)
ive@ubuntuISEC3:~$ gpg --armor --export F8C1CA08A57B9ED7 | apt-key add -
ERROR: This command can only be used by root.
ive@ubuntuISEC3:~$ sudo gpg --armor --export F8C1CA08A57B9ED7 | apt-key add -
gpg: AVISO: propiedad insegura del archivo de configuración '/home/ive/.gnupg/gpg.conf'
ERROR: This command can only be used by root.
ive@ubuntuISEC3:~$ sudo -i
root@ubuntuISEC3:~# gpg --armor --export F8C1CA08A57B9ED7 | apt-key add -
gpg: /root/.gnupg: directorio creado
gpg: creado un nuevo archivo de configuración '/root/.gnupg/gpg.conf'
gpg: ATENCIÓN: aún no se han activado en esta ejecución las opciones en '/root/.gnupg/gpg.conf'
gpg: anillo «/root/.gnupg/secring.gpg» creado
gpg: anillo «/root/.gnupg/pubring.gpg» creado
gpg: ATENCIÓN: no se ha exportado nada
gpg: no se han encontrado datos OpenPGP válidos
root@ubuntuISEC3:~# echo "deb http://labs.consol.de/repo/stable/ubuntu $(lsb_release -cs) main" > /etc/apt/sources.list.d/labs-consol-stable.list
root@ubuntuISEC3:~#
```

Y seguimos con los comandos:

```
cd /tmp/
```

```
wget https://github.com/naemon/naemon-core/archive/v1.0.6.tar.gz
```

```
tar xfv v1.0.6.tar.gz
```

```
cd naemon-core-1.0.6/
```

- Actualizamos e instalamos Naemon junto a los plugins de Nagios (estos van incluidos)

```
sudo apt-get update
```

```
sudo apt-get install naemon
```

- Indicamos la ruta donde están los plugins al archivo de configuración de Naemon  
`sudo vi /etc/naemon/resources.cfg`

```
#####
#
# RESOURCE.CFG - Sample Resource File for Naemon 1.0.8
#
#
# You can define $USERx$ macros in this file, which can in turn be used
# in command definitions in your host config file(s). $USERx$ macros are
# useful for storing sensitive information such as usernames, passwords,
# etc. They are also handy for specifying the path to plugins and
# event handlers - if you decide to move the plugins or event handlers to
# a different directory in the future, you can just update one or two
# $USERx$ macros, instead of modifying a lot of command definitions.
#
# Naemon supports up to 256 $USERx$ macros ($USER1$ through $USER256$)
#
# Resource files may also be used to store configuration directives for
# external data sources like MySQL...
#
#####
# Sets $USER1$ to be the path to the plugins
$USER1$=/usr/lib/nagios/plugins

# Sets $USER2$ to be the path to event handlers
$USER2$=/usr/lib/naemon/plugins/eventhandlers

# Store some usernames and passwords (hidden from the CGIs)
$USER3$=someuser
$USER4$=somepassword
#
#
```

- Modificamos la contraseña por defecto del administrador a la nuestra propia

```
ive@ubuntuISEC3:~$ sudo htpasswd /etc/thruk/htpasswd thrukadmin
[sudo] password for ive:
New password:
Re-type new password:
Updating password for user thrukadmin
ive@ubuntuISEC3:~$ _
```

Reiniciamos los servicios apache2 y naemon y ya podemos acceder desde la página web

```
ive@ubuntuISEC3:~$ sudo systemctl status naemon.service
• naemon.service - LSB: start and stop Naemon monitoring server
  Loaded: loaded (/etc/init.d/naemon; bad; vendor preset: enabled)
  Active: active (running) since lun 2018-11-26 12:20:54 CET; 44s ago
  Docs: man:systemd-sysv-generator(8)
```



El acceso es a la ruta **192.168.56.105/naemon**

Usuario administrador: **thrukadmin**

Contraseña: **practicass,ISE**

**Thruk**

General

Home

Documentation

Logout

Panorama View

Current Status

Tactical Overview

Map

Hosts

Services

Host Groups

Summary (Grid)

Service Groups

Summary (Grid)

Mine Map

Problems

Services (Unhandled)

Hosts (Unhandled)

Network Outages

Reports

Availability

Trends

Alerts

History (Summary)

Notifications

Event Log

Business Process

Reporting

System

Comments

Downtimes

Recurring Downtimes

Process Info

Performance Info

Scheduling Queue

Configuration

Broadcasts

Config Tool

**Thruk**

Thruk Monitoring Webinterface

Version 2.24-2

November 06, 2018

Check for updates

Read what's new in Thruk

Copyright © 2009-present Thruk Developer Team.

Copyright © 2009 Nagios Core Development Team and Community Contributors.

Copyright © 1999-2009 Ethan Galstad.

Thruk Monitoring Webinterface is licensed under the GNU General Public License and is provided AS IS with NO WARRANTY OF ANY KIND, INCLUDING THE WARRANTY OF DESIGN, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE.

**Thruk**

**github**

SOCIAL CODING

Con esto tenemos instalado el servidor Naemon en Ubuntu

- Instalamos Naemon en CentOS como cliente

Nos aseguramos de que esté actualizado el sistema con **sudo yum update**  
 Instalamos los repositorios EPEL con **sudo yum install epel-release -y**

Instalamos los plugins de Nagios, OpenSSL y NRPE

```
[live@localhost ~]$ sudo yum install nrpe nagios-plugins-all openssl_
```

Modificamos el archivo **/etc/nagios/nrpe.cfg** para que incluya la IP del servidor

```
allowed_hosts=127.0.0.1,::1,192.168.56.105
```

Iniciamos NPPE con **sudo systemctl start nrpe**

Y añadimos una excepción en el firewall sobre el puerto en el que se trabaja (5666 por defecto)

```
[live@localhost ~]$ sudo firewall-cmd --permanent --add-port=5666/tcp
success
[live@localhost ~]$ sudo firewall-cmd --reload
success
[live@localhost ~]$ _
```

- Configuramos los servicios a monitorizar (SSH y HTTP) de Ubuntu

Primero deberíamos crear un service para HTTP y otro para SSH, pero con la instalación ya vienen configurados ciertos servicios básicos, entre los que se encuentran los que necesitamos.

Service Status Details For Host 'localhost'						
<small>Select hosts / services with leftclick to send multiple commands. Select multiple with shift + mouse.            select all (hosts) - unselect all - all problems - all with downtime</small>						
Host ▲▼	Service ▲▼	Status ▲▼	Last Check ▲▼	Duration ▲▼	Attempt ▲▼	Status Information ▲▼
localhost	Current Load	OK	12:01:35	2d 23h 43m 29s	1/4	OK - load average: 0.13, 0.10, 0.12
	Current Users	OK	11:59:13	30d 1h 8m 24s	1/4	USERS OK - 1 users currently logged in
	HTTP	OK	12:01:13	30d 1h 9m 3s	1/4	HTTP OK: HTTP/1.1 302 Found - 493 bytes in 0,001 second response time
	PING	OK	12:01:59	30d 1h 8m 3s	1/4	PING OK - Packet loss = 0%, RTA = 0.13 ms
	Root Partition	OK	12:01:43	2d 23h 43m 16s	1/4	DISK OK - free space: / 3504 MB (61% inode=80%):
	SSH	CRITICAL	12:00:10	2d 23h 43m 25s	4/4	connect to address 127.0.0.1 and port 22: Conexión rehusada
	Swap Usage	CRITICAL	12:00:28	30d 1h 9m 23s	4/4 #2	SWAP CRITICAL - 0% free (0 MB out of 0 MB)
	Total Processes	OK	12:00:33	2d 23h 41m 28s	1/4	PROCS OK: 50 processes with STATE = RSZDT
<small>select all (hosts) - unselect all - all problems - all with downtime</small>						

Como vemos, aparece un problema en SSH, y es que debemos cambiar el puerto por el que hemos fijado en las prácticas ya que por defecto escucha en el 22 y no en el 22022.

Por tanto, modificamos en **/etc/naemon/conf.d/commands.cfg** indicando que el primer argumento será **"-p 22022"**, de esa manera escuchará en el puerto que queremos.

```
# 'check_ssh' command definition
define command {
    command_name      check_ssh
    command_line       $USER1$/check_ssh -p 22022 $ARG1$ $HOSTADDRESS$
```

Host ▲▼	Service ▲▼	Status ▲▼	Last Check ▲▼	Duration ▲▼	Attempt ▲▼	Status Information ▲▼
localhost	Current Load	OK	12:21:33	3d 0h 4m 7s	1/4	OK - load average: 0.00, 0.00, 0.02
	Current Users	OK	12:17:00	30d 1h 29m 2s	1/4	USERS OK - 1 users currently logged in
	HTTP	OK	12:20:29	30d 1h 29m 41s	1/4	HTTP OK: HTTP/1.1 302 Found - 493 bytes in 0,001 second response time
	PING	OK	12:19:50	30d 1h 28m 41s	1/4	PING OK - Packet loss = 0%, RTA = 0.09 ms
	Root Partition	OK	12:22:16	3d 0h 3m 54s	1/4	DISK OK - free space: / 3503 MB (61% inode=80%):
	SSH	OK	12:22:51	0d 0h 0m 0s	1/4	SSH OK - OpenSSH_7.2p2 Ubuntu-4ubuntu2.4 (protocol 2.0)
	Swap Usage	CRITICAL	12:22:14	30d 1h 30m 1s	4/4 #2	SWAP CRITICAL - 0% free (0 MB out of 0 MB)
	Total Processes	OK	12:21:30	3d 0h 2m 6s	1/4	PROCS OK: 51 processes with STATE = RSZDT



#### Service Status Details For Host 'localhost'

Select hosts / services with leftclick to send multiple commands. Select multiple with shift + mouse.  
select all (hosts) - unselect all - all problems - all with downtime

Host ▲▼	Service ▲▼	Status ▲▼	Last Check ▲▼	Duration ▲▼	Attempt ▲▼	Status Information ▲▼
localhost	Current Load	OK	12:23:58	3d 0h 5m 40s	1/4	OK - load average: 0.00, 0.00, 0.01
	Current Users	OK	12:17:00	30d 1h 30m 35s	1/4	USERS OK - 1 users currently logged in
	HTTP	OK	12:20:29	30d 1h 31m 14s	1/4	HTTP OK: HTTP/1.1 302 Found - 493 bytes in 0,001 second response time
	PING	OK	12:19:50	30d 1h 30m 14s	1/4	PING OK - Packet loss = 0%, RTA = 0.09 ms
	Root Partition	OK	12:23:54	3d 0h 5m 27s	1/4	DISK OK - free space: / 3503 MB (61% inode=80%):
	SSH	CRITICAL	12:24:24	0d 0h 0m 0s	1/4	connect to address 127.0.0.1 and port 22022: Conexión rehusada
	Swap Usage	CRITICAL	12:22:14	30d 1h 31m 34s	4/4 #2	SWAP CRITICAL - 0% free (0 MB out of 0 MB)
	Total Processes	OK	12:21:30	3d 0h 3m 39s	1/4	PROCS OK: 51 processes with STATE = RSZDT

Vemos que para localhost (Ubuntu) monitoriza correctamente, y si paramos el servicio SSH se nos notifica de la caída

- Configuramos monitorización en CentOS

Para ello debemos crear un host nuevo en el directorio de configuración:

**sudo vi /etc/naemon/conf.d/centos.cfg**

En el que se indica la máquina y los servicios a monitorizar:


```
define host {
    use                linux-server
    host_name          centos
    alias              centos
    address            192.168.56.110
}

define service {
    use                generic-service
    host_name          centos
    service_description HTTP
    check_command      check_http
}

define service {
    use                generic-service
    host_name          centos
    service_description SSH
    check_command      check_ssh
}


# El comando está modificado en el archivo de configuración para que utilice el puerto 22022
```

Reiniciamos Naemon y vemos que funciona



Select hosts / services with leftclick to send multiple commands. Select multiple with shift + mouse. select all (hosts) - unselect all - all problems - all with downtime						
Host ▲▼	Service ▲▼	Status ▲▼	Last Check ▲▼	Duration ▲▼	Attempt ▲▼	Status Information ▲▼
centos	HTTP	OK	12:12:07	0d 0h 0m 0s	1/3	HTTP OK: HTTP/1.1 200 OK - 343 bytes in 0,003 second response time
	SSH	OK	12:07:36	0d 0h 4m 31s	1/3	SSH OK - OpenSSH_7.4 (protocol 2.0)
select all (hosts) - unselect all - all problems - all with downtime						
2 of 2 Matching Service Entries Displayed						

Y si paramos los servicios nos muestra que han caído



Select hosts / services with leftclick to send multiple commands. Select multiple with shift + mouse. select all (hosts) - unselect all - all problems - all with downtime						
Host ▲▼	Service ▲▼	Status ▲▼	Last Check ▲▼	Duration ▲▼	Attempt ▲▼	Status Information ▲▼
centos	HTTP	CRITICAL	12:15:02	0d 0h 0m 1s	1/3	connect to address 192.168.56.110 and port 80: Conexión rehusada
	SSH	CRITICAL	12:15:02	0d 0h 0m 1s	1/3	connect to address 192.168.56.110 and port 22022: Conexión rehusada
select all (hosts) - unselect all - all problems - all with downtime						

### Bibliografía:

Guía de instalación oficial de Naemon.

<http://www.naemon.org/documentation/usersguide/quickstart.html>

<https://github.com/naemon/naemon>

Como la documentación oficial no se indican los paquetes necesarios para Ubuntu 16.04, se han buscado otras guías en internet y se ha contrastado con la oficial.

<https://linuxide.com/monitoring-2/setup-naemon-networking-monitoring-tool-linux/>

Guía de configuración oficial de Naemon.

<http://www.naemon.org/documentation/usersguide/monitoring-networkservices.html>