
Nagios

Índex

1. Introducció	1
Per a què serveix Nagios?	1
Nagios Core vs. Nagios XI	1
2. Arquitectura i components	2
Components principals	2
3. Instal·lació	3
Requisits previs (Debian/Ubuntu)	3
Creació de l'usuari Nagios	3
Descàrrega i compilació de Nagios Core	3
Instal·lació dels plugins de Nagios	3
Configuració d'Apache	4
Inici i habilitació del servei	4
Verificació de la instal·lació	4
4. Estructura de fitxers de configuració	4
Fitxer principal: nagios.cfg	5
5. Objectes de configuració	5
Herència i plantilles	6
6. Definició d'hosts	6
Paràmetres principals d'un host	6
Definició d'un grup de hosts	7
Exemple complet: fitxer hosts.cfg	7
7. Definició de serveis	8
Paràmetres principals d'un servei	8
Serveis comuns i les seves ordres	9
Aplicar un servei a tots els membres d'un grup	10
8. Contactes i notificacions	10
Definició de contactes	10
Definició de grups de contactes	10
Períodes de temps	11
9. Plugins de Nagios	11
Plugins inclosos amb nagios-plugins	11
Com es defineixen les ordres	12
Macros estàndard de Nagios	12
Provar un plugin manualment	12
10. NRPE: monitoratge remot d'agents	13
Instal·lació de NRPE al host remot	13
Configuració de NRPE al host remot	13
Inici del servei NRPE al host remot	13
Instal·lació del client NRPE al servidor Nagios	14
Verificació de la connexió	14

Configuració al servidor Nagios per usar NRPE	14
11. Interfície web	15
Seccions principals	15
Control d'accés a la interfície web	15
12. Funcionalitats avançades	16
12.1. Escalades de notificació	16
12.2. Dependències entre serveis	16
12.3. Event Handlers (recuperació automàtica)	16
12.4. Passive Checks	17
12.5. Freshness Checking	17
12.6. PNP4Nagios: gràfiques de rendiment	18
13. Resolució de problemes	18
Verificar la sintaxi de configuració	18
Consultar els logs	18
Errors comuns i solucions	19
14. Resum d'ordres	19
Índex de figures	21

1. Introducció

Nagios és un sistema de monitoratge de codi obert que permet supervisar la disponibilitat i el rendiment de hosts, serveis i infraestructures de xarxa. Originalment creat per Ethan Galstad el 1999 com a **NetSaint**, va ser reanomenat com a Nagios el 2002.



Figura 1: Nagios logo

Per a què serveix Nagios?

- Detectar **caigudes de serveis** (HTTP, SSH, DNS, bases de dades...) i hosts.
- Enviar **alertes per correu electrònic o SMS** quan es detecta un problema.
- Generar **gràfiques de rendiment** amb dades de CPU, memòria, disc i xarxa.
- Proporcionar una **visió centralitzada** de tota la infraestructura IT.
- Permetre la **recuperació automàtica** mitjançant Event Handlers.

Nagios Core vs. Nagios XI

Característica	Nagios Core	Nagios XI
Llicència	Codi obert (GPLv2)	Propietari (de pagament)
Interfície web	Bàsica (CGI)	Avançada i moderna
Configuració	Fitxers de text	Assistent gràfic
Suport	Comunitat	Comercial
Cost	Gratuit	De pagament

NOTA

En aquest document treballarem amb **Nagios Core**, la versió de codi obert.

2. Arquitectura i components

Nagios segueix una arquitectura **client-servidor**:

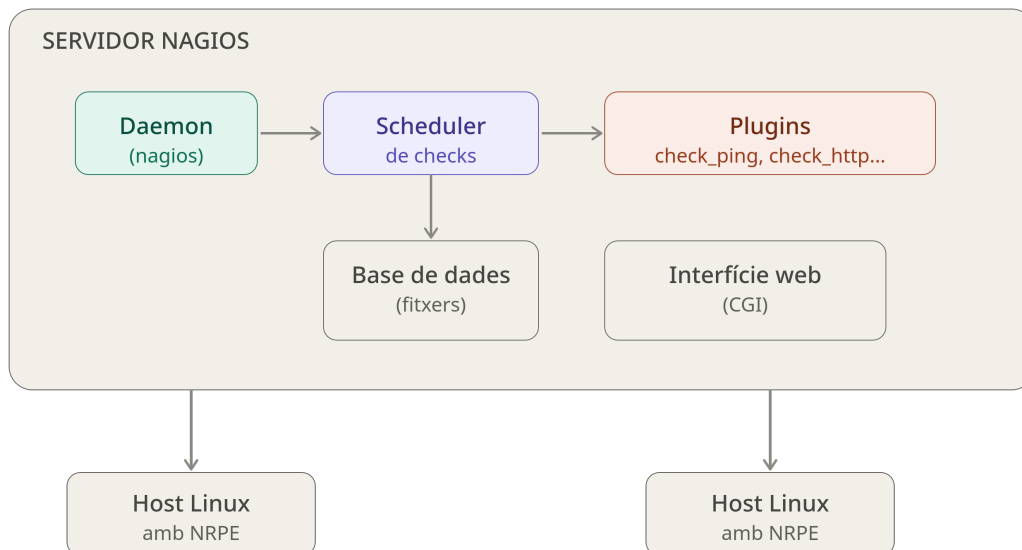


Figura 2: Arquitectura de Nagios

Components principals

Nagios Daemon (nagios) El procés principal que s'encarrega de llegir la configuració, programar els checks i processar els resultats.

Scheduler Gestor intern que determina quan s'ha d'executar cada check segons l'interval configurat.

Plugins Scripts o executables externs que realitzen les comprovacions reals. Retornen un codi de sortida estàndard:

Codi	Estat	Color
0	OK	Verd
1	WARNING	Groc
2	CRITICAL	Vermell
3	UNKNOWN	Taronja

NRPE (Nagios Remote Plugin Executor) Agent instal·lat als hosts remots que permet executar plugins localment i retornar el resultat al servidor Nagios.

3. Instal·lació

Requisits previs (Debian/Ubuntu)

```
sudo apt update
sudo apt install -y apache2 php libapache2-mod-php \
    gcc make libgd-dev libssl-dev unzip wget \
    libmcrypt-dev libnet-snmp-perl gettext
```

Creació de l'usuari Nagios

```
sudo useradd -m -s /bin/bash nagios
sudo groupadd nagcmd
sudo usermod -aG nagcmd nagios
sudo usermod -aG nagcmd www-data
```

Descàrrega i compilació de Nagios Core

```
# Descarregar l'última versió estable
cd /tmp
wget https://github.com/NagiosEnterprises/nagioscore/releases/download
↪ ad/nagios-4.4.14/nagios-4.4.14.tar.gz
tar xzf nagios-4.4.14.tar.gz
cd nagios-4.4.14

# Compilar i instal·lar
./configure --with-command-group=nagcmd
make all
sudo make install
sudo make install-init
sudo make install-commandmode
sudo make install-config
sudo make install-webconf
```

Instal·lació dels plugins de Nagios

```
cd /tmp
wget https://nagios-plugins.org/download/nagios-plugins-2.4.6.tar.gz
tar xzf nagios-plugins-2.4.6.tar.gz
cd nagios-plugins-2.4.6

./configure --with-nagios-user=nagios --with-nagios-group=nagios
make
sudo make install
```

Configuració d'Apache

```
# Crear usuari per a la interfície web
sudo htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin

# Activar mòduls d'Apache necessaris
sudo a2enmod rewrite cgi
sudo systemctl restart apache2
```

Inici i habilitació del servei

```
sudo systemctl enable nagios
sudo systemctl start nagios
sudo systemctl status nagios
```

Verificació de la instal·lació

```
# Comprovar la sintaxi de la configuració
sudo /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
```

4. Estructura de fitxers de configuració

```
/usr/local/nagios/
├── bin/
│   └── nagios                ← Executable principal
├── etc/
│   ├── nagios.cfg           ← Fitxer de configuració principal
│   ├── cgi.cfg              ← Configuració de la interfície web
│   ├── htpasswd.users       ← Usuaris web
│   ├── resource.cfg         ← Variables de macro ($USER1$...)
│   └── objects/
│       ├── commands.cfg     ← Definició d'ordres
│       ├── contacts.cfg     ← Contactes i grups
│       ├── hosts.cfg        ← Hosts monitorats (creat per l'admin)
│       ├── services.cfg     ← Serveis monitorats (creat per l'admin)
│       ├── localhost.cfg    ← Exemple de configuració local
│       ├── templates.cfg    ← Plantilles reutilitzables
│       └── timeperiods.cfg  ← Períodes de temps
├── libexec/                 ← Plugins instal·lats
├── share/                   ← Fitxers web (HTML, CGI)
├── var/
│   ├── nagios.log           ← Registre principal
│   ├── status.dat          ← Estat actual dels objectes
│   └── rw/
```

```
|      └─ nagios.cmd      ← Pipe per a ordres externes
└─ sbin/                  ← Scripts CGI de la interfície web
```

Fitxer principal: `nagios.cfg`

El fitxer `nagios.cfg` conté els paràmetres globals i les referències als fitxers d'objectes:

```
# Indica quins fitxers o directoris contenen objectes de configuració
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

# Es pot incloure un directori complet
cfg_dir=/usr/local/nagios/etc/servers

# Intervals de check (en segons)
interval_length=60

# Fitxer de log
log_file=/usr/local/nagios/var/nagios.log

# Retenció de l'estat entre reinicis
retain_state_information=1
state_retention_file=/usr/local/nagios/var/retention.dat
```

5. Objectes de configuració

Nagios utilitza un model basat en **objectes** per descriure tot allò que es vol monitorar. Els tipus d'objectes principals són:

Tipus d'objecte	Descripció
host	Equip o dispositiu de xarxa
hostgroup	Agrupació de hosts
service	Servei associat a un host
servicegroup	Agrupació de serveis
contact	Persona que rep notificacions
contactgroup	Agrupació de contactes
timeperiod	Interval de temps per a checks o notificacions
command	Ordre per executar un plugin

Herència i plantilles

Nagios permet definir **plantilles** (templates) per evitar repetir configuració. Un objecte pot heretar valors d'un altre amb el paràmetre use:

```
# Plantilla genèrica per a hosts Linux
define host {
    name                linux-server
    check_command        check-host-alive
    check_interval       5
    retry_interval       1
    max_check_attempts   3
    check_period         24x7
    notification_interval 30
    notification_period  24x7
    notification_options d,u,r
    register             0          ; No és un host real, és una
    ↪ plantilla
}

# Host que hereta la plantilla
define host {
    use                linux-server
    host_name          servidor-web
    alias              Servidor Web Principal
    address             192.168.1.10
}
```

6. Definició d'hosts

Un **host** representa qualsevol dispositiu de xarxa que volem monitorar (servidors, routers, switches, impressores...).

Paràmetres principals d'un host

```
define host {
    ; Paràmetres obligatoris
    host_name          servidor-db          ; Identificador únic
    alias              Servidor de base de dades ; Nom descriptiu
    address             192.168.1.20        ; IP o nom DNS

    ; Herència de plantilla
    use                linux-server

    ; Checks
    check_command        check-host-alive    ; Ordre per
    ↪ verificar disponibilitat
```

```

    check_interval      5                ; Cada 5 minuts en
↪ estat OK
    retry_interval      1                ; Cada minut si hi
↪ ha problema
    max_check_attempts  3                ; 3 intents abans
↪ de notificar

    ; Períodes
    check_period         24x7
    notification_period  feina           ; Notificar només
↪ en horari laboral

    ; Notificacions
    notification_options d,u,r,f         ; Down,
↪ Unreachable, Recovery, Flap
    notification_interval 60

    ; Contactes
    contacts             admin
    contact_groups        admins-ti

    ; Grups
    hostgroups            servidors-linux, servidors-bd
}

```

Definició d'un grup de hosts

```

define hostgroup {
    hostgroup_name  servidors-linux
    alias           Servidors Linux
    members         servidor-web, servidor-db, servidor-ftp
}

```

Exemple complet: fitxer hosts.cfg

```

# /usr/local/nagios/etc/objects/hosts.cfg

define host {
    use          linux-server
    host_name    portail
    alias        Servidor principal (NUC11)
    address      192.168.1.1
    hostgroups   servidors-linux
}

define host {
    use          linux-server

```

```

    host_name    portatil-debian
    alias        Portàtil Debian
    address      192.168.1.50
    hostgroups   servidors-linux, portatils
}

define hostgroup {
    hostgroup_name    servidors-linux
    alias             Servidors Linux
}

define hostgroup {
    hostgroup_name    portatils
    alias             Portàtils del departament
}

```

7. Definició de serveis

Un **servei** és una comprovació associada a un host concret. Cada host pot tenir múltiples serveis.

Paràmetres principals d'un servei

```

define service {
    ; Associació al host
    host_name          servidor-web
    service_description HTTP                               ; Nom descriptiu
    ↪ del servei

    ; Herència
    use                 generic-service

    ; Ordre de check (amb arguments)
    check_command       check_http!-p 80 -u /index.html

    ; Intervals
    check_interval      5
    retry_interval      1
    max_check_attempts  3

    ; Períodes
    check_period        24x7
    notification_period 24x7

    ; Notificacions
    notification_options w,u,c,r                          ; Warning, Unknown,
    ↪ Critical, Recovery
    notification_interval 30

```

```
; Contactes
contact_groups      admins-ti
}
```

Serveis comuns i les seves ordres

```
# Comprovar PING
define service {
    use                generic-service
    host_name          servidor-web
    service_description PING
    check_command       check_ping!100.0,20%!500.0,60%
    ; Sintaxi:
    ↪ check_ping!<warning_ms>,<warning_%>!<critical_ms>,<critical_%>
}

# Comprovar SSH (port 22)
define service {
    use                generic-service
    host_name          servidor-web
    service_description SSH
    check_command       check_ssh
}

# Comprovar espai en disc via NRPE
define service {
    use                generic-service
    host_name          servidor-web
    service_description Disc arrel (/)
    check_command       check_nrpe!check_disk
}

# Comprovar càrrega de CPU via NRPE
define service {
    use                generic-service
    host_name          servidor-web
    service_description Càrrega CPU
    check_command       check_nrpe!check_load
}

# Comprovar un port específic
define service {
    use                generic-service
    host_name          servidor-db
    service_description MySQL (3306)
    check_command       check_tcp!3306
}
```

Aplicar un servei a tots els membres d'un grup

```
define service {
    use                generic-service
    hostgroup_name     servidors-linux        ; Aplica a tots els
    ↪ membres del grup
    service_description SSH
    check_command       check_ssh
}
```

8. Contactes i notificacions

Definició de contactes

```
define contact {
    contact_name       admin-ti
    alias              Administrador de Sistemes
    email              admin@thos.local
    pager              +346000000000

    ; Quan notificar
    host_notifications_enabled 1
    service_notifications_enabled 1
    host_notification_period   24x7
    service_notification_period 24x7
    host_notification_options   d,u,r,f,s      ;
    ↪ Down,Unreachable,Recovery,Flap,Scheduled
    service_notification_options w,u,c,r,f,s

    ; Ordres de notificació
    host_notification_commands  notify-host-by-email
    service_notification_commands notify-service-by-email
}
```

Definició de grups de contactes

```
define contactgroup {
    contactgroup_name  admins-ti
    alias              Equip d'Administració TI
    members            admin-ti, responsable-ti
}
```

Períodes de temps

```
# Disponibilitat 24 hores, 7 dies
define timeperiod {
    timeperiod_name    24x7
    alias               24 Hores al dia, 7 Dies a la Setmana
    sunday              00:00-24:00
    monday              00:00-24:00
    tuesday             00:00-24:00
    wednesday           00:00-24:00
    thursday            00:00-24:00
    friday              00:00-24:00
    saturday            00:00-24:00
}

# Horari laboral
define timeperiod {
    timeperiod_name    feina
    alias              Horari laboral (L-V 8h-18h)
    monday             08:00-18:00
    tuesday            08:00-18:00
    wednesday          08:00-18:00
    thursday           08:00-18:00
    friday             08:00-18:00
}
```

9. Plugins de Nagios

Els plugins es troben a `/usr/local/nagios/libexec/`. Segueixen un estàndard: prenen arguments per línia d'ordres i retornen un codi de sortida (0, 1, 2 o 3) acompanyat d'un missatge.

Plugins inclosos amb nagios-plugins

Plugin	Funció
check_ping	Comprova la latència i pèrdua de paquets
check_http	Comprova la disponibilitat d'un servei web
check_ssh	Comprova la disponibilitat del servei SSH
check_tcp	Comprova si un port TCP és accessible
check_smtp	Comprova el servei de correu SMTP
check_dns	Comprova la resolució DNS
check_disk	Comprova l'ús de l'espai en disc (local)
check_load	Comprova la càrrega de la CPU (local)
check_users	Comprova el nombre d'usuaris connectats (local)
check_procs	Comprova processos en execució (local)
check_swap	Comprova l'ús de la memòria d'intercanvi (local)
check_snmp	Comprova valors via protocol SNMP
check_nrpe	Executa checks remots via agent NRPE

Com es defineixen les ordres

Les ordres es defineixen al fitxer `commands.cfg`:

```
# Exemple: ordre check_http amb arguments
define command {
    command_name    check_http
    command_line    $USER1$/check_http -H $HOSTADDRESS$ $ARG1$
}

# Exemple: ordre check_disk via NRPE
define command {
    command_name    check_nrpe
    command_line    $USER1$/check_nrpe -H $HOSTADDRESS$ -c $ARG1$
}
```

Macros estàndard de Nagios

Macro	Valor
<code>\$USER1\$</code>	Ruta als plugins (<code>/usr/local/nagios/libexec</code>)
<code>\$HOSTADDRESS\$</code>	Adreça IP o DNS del host actual
<code>\$HOSTNAME\$</code>	Nom del host actual
<code>\$SERVICEDESC\$</code>	Descripció del servei actual
<code>\$ARG1\$, \$ARG2\$...</code>	Arguments passats a l'ordre

Provar un plugin manualment

```
# Com a usuari nagios
sudo -u nagios /usr/local/nagios/libexec/check_http -H 192.168.1.10
↪ -p 80
# Sortida: HTTP OK: HTTP/1.1 200 OK - 1234 bytes in 0.123 second
↪ response time |...

sudo -u nagios /usr/local/nagios/libexec/check_ping -H 192.168.1.1 -w
↪ 100,20% -c 500,60%
# Sortida: PING OK - Packet loss = 0%, RTA = 1.23 ms
```

10. NRPE: monitoratge remot d'agents

L'**NRPE** (Nagios Remote Plugin Executor) permet executar plugins als hosts remots i retornar els resultats al servidor Nagios. Això és necessari per monitorar recursos locals dels hosts, com ara ús de disc, CPU, processos o memòria.

Instal·lació de NRPE al host remot

```
# Al host que volem monitorar (host client/agent)
sudo apt install nagios-nrpe-server nagios-plugins-basic
↳ nagios-plugins-standard
```

Configuració de NRPE al host remot

Editar /etc/nagios/nrpe.cfg:

```
sudo nano /etc/nagios/nrpe.cfg
```

```
# Adreça IP del servidor Nagios que pot fer consultes
allowed_hosts=127.0.0.1, 192.168.1.100 ; Afegir la IP del servidor
↳ Nagios

# Habilitar arguments (necessari per a checks parametrizats)
dont_blame_nrpe=0 ; Deixar a 0 per seguretat si no cal

# Definició dels checks locals disponibles
command[check_users]=/usr/lib/nagios/plugins/check_users -w 5 -c 10
command[check_load]=/usr/lib/nagios/plugins/check_load -w 15,10,5 -c
↳ 30,25,20
command[check_disk]=/usr/lib/nagios/plugins/check_disk -w 20% -c 10%
↳ -p /
command[check_swap]=/usr/lib/nagios/plugins/check_swap -w 20% -c 10%
command[check_procs]=/usr/lib/nagios/plugins/check_procs -w 150 -c
↳ 200 -s RSZDT
```

Inici del servei NRPE al host remot

```
sudo systemctl enable nagios-nrpe-server
sudo systemctl start nagios-nrpe-server
```

Instal·lació del client NRPE al servidor Nagios

```
# Al servidor Nagios
sudo apt install nagios-nrpe-plugin
```

Verificació de la connexió

```
# Des del servidor Nagios, provar la connexió al host remot
/usr/local/nagios/libexec/check_nrpe -H 192.168.1.20
# Sortida esperada: NRPE v4.1.0

# Executar un check remot
/usr/local/nagios/libexec/check_nrpe -H 192.168.1.20 -c check_disk
# Sortida: DISK OK - free space: / 15432 MB (78% inode=90%)
```

Configuració al servidor Nagios per usar NRPE

```
# A commands.cfg
define command {
    command_name    check_nrpe
    command_line    $USER1$/check_nrpe -H $HOSTADDRESS$ -c $ARG1$
}

# A services.cfg
define service {
    use              generic-service
    host_name        servidor-web
    service_description Disc arrel (/)
    check_command    check_nrpe!check_disk
}

define service {
    use              generic-service
    host_name        servidor-web
    service_description Càrrega CPU
    check_command    check_nrpe!check_load
}
```

11. Interfície web

L'accés a la interfície web de Nagios Core es fa des d'un navegador:

```
http://<ip-servidor-nagios>/nagios/
```

Les credencials s'han configurat durant la instal·lació amb `htpasswd`.

Seccions principals

Tactical Overview Resum general de l'estat de tots els hosts i serveis: quants estan en OK, WARNING, CRITICAL, UNKNOWN i quants checks han fallat.

Hosts Llistat de tots els hosts monitorats amb el seu estat actual, última comprovació i informació addicional.

Services Llistat de tots els serveis amb el seu estat, l'hora de l'última comprovació i el missatge retornat pel plugin.

Host Groups / Service Groups Vista agrupada per grups definits a la configuració.

Problems Vista filtrada que mostra únicament els problemes actuals (CRITICAL, WARNING o UNKNOWN).

Event Log Registre cronològic de tots els canvis d'estat i notificacions.

Status Maps Mapa visual de la infraestructura monitorada (requereix el paquet `libgd`).

Control d'accés a la interfície web

La configuració d'accés es troba a `cgi.cfg`:

```
# Usuari que té accés d'administrador complet
authorized_for_all_services=nagiosadmin
authorized_for_all_hosts=nagiosadmin
authorized_for_system_information=nagiosadmin
authorized_for_configuration_information=nagiosadmin
authorized_for_system_commands=nagiosadmin
authorized_for_all_service_commands=nagiosadmin
authorized_for_all_host_commands=nagiosadmin
```

12. Funcionalitats avançades

12.1. Escalades de notificació

Permeten enviar alertes a contactes addicionals si un problema no es resol en un temps determinat:

```
define serviceescalation {
    host_name                servidor-web
    service_description      HTTP
    first_notification        3          ; Escalar a partir de la 3a
    ↪ notificació
    last_notification         0          ; 0 = sense límit
    notification_interval    15
    contact_groups            direccio-ti
}
```

12.2. Dependències entre serveis

Permet definir que un check no s'executi si un servei del qual depèn ja ha fallat:

```
define servicedependency {
    host_name                servidor-web
    service_description      Xarxa
    dependent_host_name      servidor-web
    dependent_service_description HTTP
    execution_failure_criteria c,u      ; No executar check HTTP
    ↪ si Xarxa és CRITICAL o UNKNOWN
    notification_failure_criteria c,u
}
```

12.3. Event Handlers (recuperació automàtica)

Permet executar una acció automàtica quan un servei canvia d'estat. Per exemple, reiniciar Apache si cau:

```
# Script: /usr/local/nagios/libexec/event_handler_apache.sh
#!/bin/bash
case "$1" in
    WARNING) ;;
    UNKNOWN) ;;
    CRITICAL) /usr/bin/sudo /bin/systemctl restart apache2 ;;
    OK)      ;;
esac
exit 0
```

```

define service {
    use                generic-service
    host_name          servidor-web
    service_description HTTP
    check_command       check_http
    event_handler_enabled 1
    event_handler       event_handler_apache!$SERVICESTATE$
}

```

12.4. Passive Checks

En els **checks actius** (per defecte), el servidor Nagios inicia el check. En els **checks passius**, és el propi host qui envia l'estat a Nagios. Útil per a dispositius darrere de tallafocs o per a tasques programades:

```

# Enviar un resultat passiu via ordre externa
echo "[$(date +%s)]
↳ PROCESS_SERVICE_CHECK_RESULT;servidor-web;HTTP;0;HTTP OK -
↳ Resultat passiu" \
  | sudo tee /usr/local/nagios/var/rw/nagios.cmd

```

12.5. Freshness Checking

Permet detectar si un host o servei no ha enviat un check passiu en un temps màxim:

```

define service {
    use                generic-service
    host_name          servidor-backup
    service_description Backup nocturn
    check_freshness    1
    freshness_threshold 86400          ; 24 hores en segons
    check_command       check_dummy!2!No s'ha rebut resultat del
↳ backup
    active_checks_enabled 0            ; Desactivar checks actius
    passive_checks_enabled 1
}

```

12.6. PNP4Nagios: gràfiques de rendiment

PNP4Nagios és un complement per generar gràfiques de rendiment a partir de les dades de performance data que retornen els plugins:

```
sudo apt install pnp4nagios
```

Els plugins que suporten performance data retornen dades com:

```
HTTP OK: HTTP/1.1 200 OK - 1234 bytes in 0.123 second response time  
↪ |time=0.123s;;;0.000 size=1234B;;;0
```

Les dades després del | s'emmagatzemen i es representen gràficament.

13. Resolució de problemes

Verificar la sintaxi de configuració

Sempre executar aquest pas abans de reiniciar el servei:

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

Consultar els logs

```
# Log principal de Nagios  
sudo tail -f /usr/local/nagios/var/nagios.log  
  
# Log d'Apache (per a problemes amb la interfície web)  
sudo tail -f /var/log/apache2/error.log  
  
# Journal del sistema  
sudo journalctl -u nagios -f
```

Errors comuns i solucions

El daemon no arrenca: error a la configuració

```
Error: Could not open config file
↳ '/usr/local/nagios/etc/objects/hosts.cfg' for reading!
```

Solució: El fitxer referenciat a `nagios.cfg` no existeix o no té permisos correctes.

```
sudo chown nagios:nagios /usr/local/nagios/etc/objects/hosts.cfg
```

El check NRPE falla: “Connection refused”

```
# Comprovar que el servei NRPE s'executa al host remot
sudo systemctl status nagios-nrpe-server

# Comprovar el tallafoc al host remot
sudo ufw allow 5666/tcp    # Port NRPE per defecte

# Verificar que el servidor Nagios és a la llista allowed_hosts
grep allowed_hosts /etc/nagios/nrpe.cfg
```

“Return code of 127 for check...” (plugin no trobat)

El plugin no existeix o no és executable:

```
ls -la /usr/local/nagios/libexec/check_nrpe
sudo chmod +x /usr/local/nagios/libexec/check_nrpe
```

Notificacions per correu no s'envien

```
# Comprovar que el MTA (Postfix, sendmail) funciona
echo "Test" | mail -s "Prova Nagios" admin@thos.local

# Verificar que les notificacions estan habilitades
grep notifications_enabled /usr/local/nagios/etc/nagios.cfg
```

14. Resum d'ordres

```
# Verificar configuració
sudo /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg

# Gestió del servei
sudo systemctl start|stop|restart|status nagios

# Recarregar configuració sense aturar el servei
sudo systemctl reload nagios
```

```
# Provar un plugin manualment
sudo -u nagios /usr/local/nagios/libexec/check_http -H <ip> -p 80

# Provar un check NRPE
sudo -u nagios /usr/local/nagios/libexec/check_nrpe -H
↳ <ip-host-remot> -c check_disk

# Consultar el log en temps real
sudo tail -f /usr/local/nagios/var/nagios.log
```

Índex de figures

1	Nagios logo	1
2	Arquitectura de Nagios	2

Versions d'aquest document

- [HTML](#)
- [PDF](#)
- [ODT](#)
- [MD](#)

[Domini Públic \(CC0\)](#)