

# Instalación de Nagios Core 4 + PNP4Nagios + Check\_mk + Nagvis en CentOS 7 - Redhat 7

## Introducción.

A fecha de hoy todavía no hay paquetes RPM en las distribuciones / repositorios habituales para Nagios 4 así que es necesario instalar compilando. Ya vimos en un artículo previo como instalar Nagios Core 4 y PNP4Nagios en una Centos 6.X. Por entonces check\_mk Livestatus no estaba disponible para Nagios 4. En esta ocasión vamos a ver como instalar todo el pack imprescindible en Centos 7 (Redhat 7), Nagios Core 4 + Check\_mk + PNP4Nagios + Nagvis. Instalaremos todo en sus últimas versiones a fecha de hoy.

Partimos de una instalación de Centos 7 instalada con la opción “minimal” y con SELinux deshabilitado. Necesitamos tener configurados además de los repositorios de Centos / Redhat, los repositorios de EPEL. Adicionalmente le instalamos algunos paquetes necesarios.

```
yum install perl wget httpd php
yum groupinstall "Development Tools"
```

## Software necesario.

- [Nagios Core 4](#), edición Core DIY Source. En el momento de escribir este artículo v.4.0.8.
- **Nagios Plugins.** Podemos optar por los de [nagios.org](#) o por los de [monitoring-plugins](#) que eran los standard y denominados nagios plugins hasta hace unos meses. Si tienes interés en saber el porqué de esta “oferta” de plugins puedes echar un ojo a [este artículo](#). En este caso usaremos los de monitoring-plugins pero son similares actualmente.
- PNP4Nagios. [Bajamos](#) la última versión que en el momento era la 0.6.24
- [Check\\_mk](#). En su página de [downloads](#) tenemos disponibles para descarga siempre dos versiones. La estable y la “innovate” que viene a ser la última versión alfa/beta con todas las novedades. Al gusto. Yo suelo usar la innovate ya que se actualiza muy a menudo con novedades muy interesantes y suele funcionar muy bien.
- Nagvis. En el artículo usaremos la última [versión disponible](#) 1.8rc2-

## Cuentas de usuario / grupo.

Creamos cuenta para Nagios. Creamos un grupo “nagcmd” que usaremos luego para los comandos externos y metemos en dicho grupo a nuestro usuario Nagios y al usuario que usa apache.

```
useradd nagios && passwd nagios
groupadd nagcmd
usermod -a -G nagcmd nagios
usermod -a -G nagcmd apache
```

Esto último se hace para que desde el GUI de Nagios podamos ejecutar comandos externos sobre host y servicios – usuario apache-. Para ello los ficheros donde van dichos comandos (/usr/local/nagios/var/rw/) veremos posteriormente que pertenecen a ese grupo “nagcmd”

## Instalación de Nagios Core 4.

Instalamos previamente los paquetes necesarios habituales para Nagios.

```
yum install wget httpd php gcc glibc glibc-common gd gd-devel make net-snmp libpng-devel libjpeg-turbo-devel
```

En el momento de realizar la instalación no había paquete disponible en EPEL para gd-devel así que se bajó e instaló el RPM para Centos 7 de [rpmfind](#) clear

Descomprimos el gz de Nagios Core y ejecutamos la instalación.

```
tar xzfv nagios-4.X.X.tar.gz
cd nagios-4.X.X
./configure --with-command-group=nagcmd
```

Al finalizar nos muestra el resumen.

```
...
*** Configuration summary for nagios 4.0.8 08-12-2014 ***:

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/rc.d/init.d
    Apache conf.d directory: /etc/httpd/conf.d
    Mail program:      /bin/mail
    Host OS:           linux-gnu
    IOBroker Method:   epoll

Web Interface Options:
-----
    HTML URL:  http://localhost/nagios/
    CGI URL:   http://localhost/nagios/cgi-bin/

Traceroute (used by WAP):
```

Seguimos con el resto de pasos habituales:

```
make all
make install
make install-init
make install-commandmode
make install-config
make install-webconf
```

Copiamos algunos ficheros desde dir fuentes y cambiamos algunos permisos que parece quedan pendientes:

```
cp -R contrib/eventhandlers/ /usr/local/nagios/libexec/  
chown -R nagios:nagios /usr/local/nagios/libexec/eventhandlers
```

Chequeamos que la configuración es válida, creamos usuario de acceso por apache, iniciamos Nagios y apache y los configuramos para que se arranquen al iniciar el sistema:

```
/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg  
htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin  
systemctl restart httpd.service && systemctl restart httpd.service  
chkconfig --add nagios && chkconfig --level 35 nagios on  
systemctl enable httpd.service
```

Apuntamos a la URL y listo: [http://nuestra\\_ip/nagios/](http://nuestra_ip/nagios/) (acordaros de habilitar puerto 80 en el firewall local)

Puede que sea necesario configurar Firewall para poder acceder a nuestro Apache por el puerto 80.

```
firewall-cmd --zone=public --permanent --add-port=80/tcp  
systemctl restart firewalld.service
```

## **Instalación de Nagios Plugins.**

Ya puestos a compilar, nos bajamos también la última versión disponible de los plugins para instalarlos de dicha manera (nos liaríamos más en este caso si lo instalamos desde paquetes desde EPEL como se indicaba en , pero podemos hacerlo si deseamos). Podemos optar por los plugins que publica Nagios Enterprise

### **Opción1. Instalación desde paquetes de repositorio EPEL**

Más sencillo de instalar y mantener, plugin menos actualizados.

```
yum install nagios-plugins-all nagios-plugins-nrpe
```

Los plugin se instalan en el directorio `/usr/lib64/nagios/plugins/check_ping`

Nagios espera encontrarlos en `/usr/local/nagios/libexec`

Lo más sencillo será cambiar en `/usr/local/nagios/etc/resource.cfg` la ubicación de estos.

```
#$USER1$=/usr/local/nagios/libexec  
$USER1$=/usr/lib64/nagios/plugins
```

### **Opción 2. Compilamos.**

Más complicado?. Tenemos la última versión y los plugins de nagios o de monitoringexchange a elegir.

Previamente instalamos algunos paquetes necesarios para algunos plugins:

```
yum install bind-utils net-snmp net-snmp-devel net-snmp-utils net-snmp-perl perl-Net-SNMP
```

Y otros necesarios si vamos a chequear en concreto ese tipo de servicios desde Nagios. Si no lo vamos a hacer, mejor no instalarlos. Aunque si los necesitamos posteriormente tendremos que volver a compilar.

```
yum install mariadb-libs mariadb-devel samba-client samba-common postgresql-devel  
openldap-devel
```

Descomprimos el fichero de nagios-plugins y compilamos indicándoles nuestro usuario y grupo de nagios:

```
tar xzfv monitoring-plugins-2.X.X.tar.gz  
cd monitoring-plugins-2.1.1  
# Si instalas nagios-plugins  
./configure -with-nagios-user=nagios -with-nagios-group=nagios  
# Si instalas monitoring-plugins  
./configure -with-nagios-user=nagios -with-nagios-group=nagios  
--prefix=/usr/local/nagios  
make  
make install
```

Si accedemos a nuestro Nagios veremos que tiene algunos chequeos clásicos en el localhost que ya están usando los plugins instalados.



Dado que cada vez que cambiamos la configuración de ficheros de Nagios conviene chequearla (y necesitamos “recargar” la configuración) yo acostumbro a crear unos alias para tenerlo más a mano. En el fichero .bashrc de nuestro usuario podemos añadirlos.

```
# ALIAS PARA NAGIOS  
alias q1='/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg'  
alias q2='/etc/init.d/nagios reload'
```

## PNP4Nagios

Debemos instalar si o si PNP4Nagios compilando ya que si lo instalas mediante paquetes de EPEL intentará instalarte dependencias tan incomodas como un Nagios 3 entero :)

Previamente instalamos algunos paquetes necesarios.

```
yum install rrdtool perl-Time-HiRes rrdtool-perl php-gd
```

### Instalación.

En el momento de la instalación no existía en los repositorios el paquete rrdtool-perl ni localicé el equivalente así que hubo que bajarlo para Centos 7 de [rpmfind.net](http://rpmfind.net) e instalarlo:

```
yum localinstall rrdtool-perl-1.4.8-8.el7.x86_64.rpm
```

Y luego instalamos compilando pnp4nagios

```
tar zxvf pnp4nagios-0.6.XX.tar.gz
cd pnp4nagios-0.6.XX
./configure
make all
make fullinstall
```

Configuramos el demonio npcd de pnp4nagios para que se inicie automáticamente y lo iniciamos:

```
chkconfig --add npcd && chkconfig --level 35 npcd on
systemctl reload httpd.service
```

Si accedemos al interface de pnp4nagios: [http://nuestra\\_ip/pnp4nagios/](http://nuestra_ip/pnp4nagios/) deberíamos ver todos los chequeos que realiza en verde.

## PNP4Nagios Environment Tests

The following options are determined by "configure". If any of the tests have failed, consult the [documentation](#) for more information on how to correct the problem.

|                          |   |
|--------------------------|---|
| PNP4Nagios Version       | pnp4nagios-0.6.24                               |
| Prefix                   | /usr/local/pnp4nagios                           |
| Configure Arguments      | ./configure                                     |
| RRD Storage              | /usr/local/pnp4nagios/var/perfdata is readable. |
| RRDtool Binary           | /usr/bin/rrdtool is executable by PHP           |
| PHP GD extension         | Pass  |
| PHP function proc_open() | Pass  |
| PHP zlib extension       | Pass  |
| PHP session extension    | Pass  |
| PHP JSON extension       | Pass  |
| PHP magic_quotes_gpc     | Off   |
| PHP socket extension     | Pass  |
| Apache Rewrite Module    | Pass  |

## Kohana Environment Tests

The following tests have been run to determine if Kohana will work in your environment. If any of the tests have failed, consult the [documentation](#) for more information on how to correct the problem.

|                        |  |
|------------------------|--|
| PHP Version            | 5.4.16                                   |
| System Directory       | /usr/local/pnp4nagios/lib/kohana/system/ |
| Application Directory  | /usr/local/pnp4nagios/share/application/ |
| Reflection Enabled     | Pass                                     |
| Iconv Extension Loaded | Pass                                     |
| URI Determination      | Pass                                     |

Your environment passed all requirements. Remove or rename the /usr/local/pnp4nagios/share/install.php file NOW.

Procedemos como nos indica a renombrar el fichero /usr/local/pnp4nagios/share/install.php para que se dé por enterado de que está correctamente instalado:

```
mv /usr/local/pnp4nagios/share/install.php /usr/local/pnp4nagios/share/install.php.ORI
```

Volvemos a cargar la página y veremos que ahora nos da un error: **Please check the documentation for information about the following error. perfdata directory "/usr/local/pnp4nagios/var/perfdata/" is empty. Please check your Nagios config. Read FAQ online**

De momento es normal, nos faltan configurar la integración con nagios.

## Integración con Nagios.

PNP4Nagios tiene varias formas de configurar la integración con Nagios. Usaremos el método bulk-mode. Si estás pensando en usar el más sencillo e igual de eficaz “Bulk Mode with npcdmod” olvídalos de momento. Como comentaba, la API “event broker” de Nagios Core 4 ha cambiado con respecto a la versión previa, con lo cual el broker nocdmod no va a funcionar (al menos no el de la última versión probada en este artículo – pnp4nagios 0.6.24-, en la [página de documentación de pnp4nagios](#), indica claramente “**PNP4Nagios Broker Module npcdmod.o is not compatible with Nagios Core 4.x**”).

En el directorio /usr/local/pnp4nagios/etc tenemos dos ficheros de ejemplo para hacer copy/paste.

Del fichero **nagios.cfg-sample** copiamos la siguiente configuración a el fichero **/usr/local/nagios/etc/nagios.cfg**

```
#
# Bulk / NPCD mode
#
process_performance_data=1
# *** the template definition differs from the one in the original nagios.cfg
#
service_perfdata_file=/usr/local/pnp4nagios/var/service-perfdata
service_perfdata_file_template=DATATYPE::SERVICEPERFDATA\tTIMET::$TIMET$\tHOSTNAME::$
$HOSTNAME$\tSERVICEDESC::$SERVICEDESC$\tSERVICEPERFDATA::$
$SERVICEPERFDATA$\tSERVICECHECKCOMMAND::$SERVICECHECKCOMMAND$\tHOSTSTATE::$
$HOSTSTATES$\tHOSTSTATETYPE::$HOSTSTATETYPE$\tSERVICESTATE::$
$SERVICESTATE$\tSERVICESTATETYPE::$SERVICESTATETYPE$
service_perfdata_file_mode=a
service_perfdata_file_processing_interval=15
service_perfdata_file_processing_command=process-service-perfdata-file
# *** the template definition differs from the one in the original nagios.cfg
#
host_perfdata_file=/usr/local/pnp4nagios/var/host-perfdata
host_perfdata_file_template=DATATYPE::HOSTPERFDATA\tTIMET::$TIMET$\tHOSTNAME::$
$HOSTNAME$\tHOSTPERFDATA::$HOSTPERFDATA$\tHOSTCHECKCOMMAND::$
$HOSTCHECKCOMMAND$\tHOSTSTATE::$HOSTSTATE$\tHOSTSTATETYPE::$HOSTSTATETYPE$
host_perfdata_file_mode=a
host_perfdata_file_processing_interval=15
host_perfdata_file_processing_command=process-host-perfdata-file
```

Del fichero **misccommands.cfg-sample** copiamos la siguiente configuración a `/usr/local/nagios/etc/objects/commands.cfg`

```
#
# Bulk with NPCD mode
#
define command {
    command_name process-service-perfdata-file
    command_line /bin/mv /usr/local/pnp4nagios/var/service-perfdata /usr/local/pnp4nagios/var/spool/service-perfdata.$TIMET$
}
define command {
    command_name process-host-perfdata-file
    command_line /bin/mv /usr/local/pnp4nagios/var/host-perfdata /usr/local/pnp4nagios/var/spool/host-perfdata.$TIMET$
}
```

Reiniciamos el demonio `npcd` de `pnp4nagios` y el propio Nagios.

```
service npcđ restart && service nagios restart
```

Y ya podemos conectarnos a la URL de PNP4Nagios. Desaparece el error y veremos al rato que empieza a pintar nuestras gráficas.

Nos faltaría ahora una cosa importante. Poder acceder directamente desde el Host / Servicio en Nagios a las gráficas correspondientes de este sin tener que ir a el interface de PNP4 para localizarlo. Siguiendo las instrucciones de la documentación nos vamos directamente a configurarlo para POPUPs que queda muy mono. Creamos unos templates para aplicarlos luego a los objetos.

En el fichero `/usr/local/nagios/etc/objects/templates.cfg` (p.e.) añadimos:

```
define host {
    name host-pnp
    action_url /pnp4nagios/index.php/graph?host=$HOSTNAME$&srv=_HOST_' class='tips'
    rel='/pnp4nagios/index.php/popup?host=$HOSTNAME$&srv=_HOST_'
    register 0
}
define service {
    name srv-pnp
    action_url /pnp4nagios/index.php/graph?host=$HOSTNAME$&srv=$SERVICEDESC$' class='tips'
    rel='/pnp4nagios/index.php/popup?host=$HOSTNAME$&srv=$SERVICEDESC$
    register 0
}
```



Y modificamos nuestros objetos Host / Servicios para que hereden dicha plantilla. P.e.

```
define host{
use linux-server,host-pnp
host_name localhost
alias localhost
address 127.0.0.1
}
define service{
use local-service,svr-pnp
host_name localhost
service_description PING
check_command check_ping!100.0,20%!500.0,60%
}
```

Aunque realmente si vas a usar luego `check_mk` puedes despreocuparte de esta última configuración de templates y asignación de estas a objetos. CMK hace estas cosas por ti.

Si no controlas bien el tema de los ficheros de configuración de Nagios es que no leíste el estupendo artículo [Nagios Core. Ficheros de configuración](#) . Si no te aclaras con los ficheros de configuración de objetos debes leer el artículo de [Nagios, Introducción a los objetos](#).

Podemos añadirle la plantilla a todos aquellos host / servicios que queramos tengan gráfica. Las estrategias pueden ser diferentes; uno a uno, definición de servicios asignadas a grupos de host en lugar de a hosts (lo normal), asignar la plantilla a la plantillas a otra plantilla (local-service p.e.), ....

OJO. Es necesario copiar un fichero a la config web de Nagios y las rutas de la doc de PNP4Nagios no coinciden con las nuestras. Copiamos el fichero correctamente de los fuentes de pnp4nagios:

```
cp contrib/ssi/status-header.ssi /usr/local/nagios/share/ssi/
```

Reiniciamos nagios y ncmd (módulo de pnp):

```
service npcd restart && service nagios restart
```

Veremos que nos han aparecido los iconos para acceder pinchando directamente a las gráficas del Host / Servicio directamente desde este y además, si pasamos solo el ratón por el icono nos muestra ya un popup con la gráfica. Bonito ¿no?.

| Host      | Service         | Status  | Last Check          | Duration      | Attempt | Status Information  |
|-----------|-----------------|---------|---------------------|---------------|---------|---|
| localhost | Current Load    | OK      | 12-03-2014 00:17:55 | 0d 0h 58m 39s | 1/4     | OK - load average: 0.43, 0.33, 0.29   |
|           | Current Users   | OK      | 12-03-2014 00:17:45 | 0d 0h 58m 49s | 1/4     | USERS OK - 3 users currently logged in  |
|           | HTTP            | WARNING | 12-03-2014 00:18:22 | 0d 0h 58m 12s | 4/4     | HTTP WARNING: HTTP/1.1 403 Forbidden - 5162 bytes in 0,015 second response time |
|           | PING            | OK      |                     |               |         | PING OK - Packet loss = 0%, RTA = 0.29 ms                                       |
|           | Root Partition  | OK      |                     |               |         | DISK OK - free space: / 16618 MB (87% inode=99%):                               |
|           | SSH             | OK      |                     |               |         | SSH OK - OpenSSH_6.4 (protocol 2.0)   |
|           | Swap Usage      | OK      |                     |               |         | SWAP OK - 99% free (1035 MB out of 1055 MB)                                     |
|           | Total Processes | OK      | 12-03-2014 00:16:30 | 0d 0h 55m 4s  | 1/4     | PROCS OK: 238 processes with STATE = RSZDT                                      |

Results 1 - 8 of 8 Matching Services

Si quieres saber más de PNP4Nagios échale un ojo al artículo de [Características de PNP4Nagios](#).

## Check\_mk

Continuamos con la instalación de check\_mk.

### Prerequisitos

Necesitamos un módulo para Apache del repositorio de EPEL. El problema es que este módulo ya no va a estar disponible más en EPEL ya que hay otro módulo python para Apache con funcionalidad similar y parece que bastante mejor (mod\_wgsi). Hasta que CMK migre a este (que parece están en ello) debemos buscarnos la vida para instalarlo. Podemos seguir el artículo [“Compilar módulo de Apache mod\\_python para Centos 7 / Redhat 7”](#) para compilarlo e instalarlo.

### Instalación de check\_mk y Livestatus

Descomprimos como siempre y lanzamos la instalación que en este se basa en un script.

```
tar xzfv check_mk-1.X.XiX.tar.gz
cd check_mk-1.X.XiX
./setup.sh
```

Durante la instalación nos va haciendo preguntas de localización de ficheros pero nos facilita casi siempre la tarea ofreciéndonos un valor por defecto normalmente válido y detectando la mayoría de los valores.

Debemos rectificar en lo siguientes valores:

#### Nagios command pipe

( default --> /var/log/nagios/rw/nagios.cmd): /usr/local/nagios/var/rw/nagios.cmd

#### Path to check\_icmp (depende que plugins instales cambiará el camino)

( autodetected --> /usr/lib64/nagios/plugins/check\_icmp): /usr/local/libexec/check\_icmp

#### HTTP authentication file

( default --> /etc/nagios/htpasswd.users): /usr/local/nagios/etc/htpasswd.users

#### PNP4Nagios templates

( default --> /usr/share/check\_mk/pnp-templates): /usr/local/pnp4nagios/share/templates

RRD files








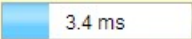











( default --> /var/lib/nagios/rrd): /usr/local/pnp4nagios/var/perfdata

Si algo va mal puedes repetir la instalación las veces que quieras y rectificar. CMK recuerda las respuestas previas ya que las guardá en un fichero “.check\_mk\_setup.conf” en nuestro home. De hecho cuando actualicemos check\_mk será tan sencillo como ejecutar la nueva versión y aceptar todos los valores ya correctos de la instalación previa.

Una vez finalizado reiniciamos los servicios.

```
systemctl restart httpd.service && service nagios restart
```

Y probamos a acceder a cmk → [http://server\\_nagios/check\\_mk](http://server_nagios/check_mk)

| localhost |                 |   |  |       |         |  |
|-----------|-----------------|---|--|-------|---------|--|
| State     | Service         | Status detail   | Icons  | Age   | Checked | Perf-O-Meter   |
| OK        | Current Load    | OK - load average: 0.04, 0.73, 0.77   |     | 3 hrs | 4 min   |  |
| OK        | Current Users   | USERS OK - 2 users currently logged in  |     | 4 hrs | 18 sec  |  |
| WARN      | HTTP            | HTTP WARNING: HTTP/1.1 403 Forbidden - 5237 bytes in 0.003 second response time |    | 4 hrs | 9 sec   |  3.4 ms |
| OK        | PING            | PING OK - Packet loss = 0%, RTA = 0.04 ms                                       |     | 4 hrs | 5 min   |  |
| OK        | Root Partition  | DISK OK - free space: / 14565 MB (86% inode=94%):                               |     | 4 hrs | 57 sec  |  |
| OK        | SSH             | SSH OK - OpenSSH_5.3 (protocol 2.0)   |    | 4 hrs | 18 sec  |  |
| OK        | Swap Usage      | SWAP OK - 100% free (2047 MB out of 2047 MB)                                    |     | 4 hrs | 4 min   |  |
| OK        | Total Processes | PROCS OK: 66 processes with STATE = RSZDT                                       |     | 4 hrs | 5 min   |  |

Todo debería funcionar correctamente. Si no es así verificar que la instalación añadió en el nagios.cfg la carga del módulo / broker de cmk livestatus:

```
# Load Livestatus Module
broker_module=/usr/lib/check_mk/livestatus.o /usr/local/nagios/var/rw/live
event_broker_options=-1
# added by setup.sh of check_mk
cfg_dir=/usr/local/nagios/etc/check_mk.d
```

Y luego revisar el fichero nagios.log para verificar que al reiniciar Nagios no existan problemas con la carga del broker de cmk livestatus.

# Nagvis.

## Prerequisitos.

Instalamos los paquetes necesarios que nos instalarán a su vez bastantes dependencias.

```
yum install php-mbstring php-pdo graphviz
```

## Instalación.

Descomprimos e instalamos. Nagvis se instala también mediante un script.

```
tar xfv nagvis-1.8xx.tar.gz  
cd nagvis-1.8xx  
./install.sh
```

La instalación nos realiza preguntas acerca de rutas de nuestro software, usuarios, Backends a usar (seleccionaremos solo mklivestatus -por defecto-). Si hemos seguido paso a paso este artículo nos valdrán las opciones por defecto.

Por último editamos el fichero de configuración que nos crea para apache en /etc/httpd/conf.d/nagvis.conf y hacemos un par de cambios para que funcione correctamente en esta versión de apache (2.4):

```
#AllowOverride None  
Require all granted
```

Una vez finalizada la instalación reiniciamos / recargamos Apache como siempre y ya podremos acceder Nagvis → <http://server.nagios/nagvis>. (admin / admin)

The screenshot shows the Nagvis web interface. At the top, there is a navigation bar with the Nagvis logo, an 'Open' dropdown, a 'User menu' dropdown, a 'Choose Language' dropdown, and a '¿Necesita Ayuda?' dropdown. Below the navigation bar is a section titled 'Índice de Mapa' (Map Index) containing a grid of demo links. Each link is accompanied by a status icon: a red 'X' for failed or unavailable, a green checkmark for successful, and a globe icon for a specific demo. The demos listed are: Demo: 5 Automap (red X), Demo: 2 Overview System Load (green checkmark), Demo: 3 Server - muc-srv1 (green checkmark), Demo: 1 Datacenter Hamburg (globe icon), Demo: 0 Overview Germany (red X), Demo: 4 Geomap (green checkmark), and Demo Overview (red X). Below the map index is a section titled 'Pools de Rotación' (Rotation Pools) with a table listing various demo pools under the 'demo' category: demo-germany, demo-ham-racks, demo-load, demo-muc-srv1, demo-geomap, and demo-automap.

Lo primero que tendremos que hacer en Nagvis será probar que funciona el acceso a los datos de nuestro nagios a través de mklivestatus.

Ir a un mapa existente (p.e. “Demo1. Datacenter Hamburg”).

- Menu Edut map / lock-unlock all /
- Menu Edit Map / Map Options / listbox “backend\_id” y selecciona live\_1 (save).
- Menu Edit Map / Add Icon / Host, situa el puntero en el map, selecciona un un host en el “listbox” (tendrás al menos tu localhost) y listo.
- Menu Edut map / lock-unlock all para quitar el modo edición.

Ahora ya podemos crear nuevos mapas usando imágenes que subamos y empezar a situar nuestros objetos en estas. Al crear nuestro mapa usaremos el backend “live\_1”

Nagvis incluye muchos mapas que están bien para echarle un ojo inicialmente pero que generan mucha confusión así que mejor que los borres.

## **Artículos relacionados**

Otros artículos que te pueden ser de interés:

- [PNP4Nagios. Características - guía rápida.](#)
- [Nagios Core. Archivos de configuración.](#)
- [Nagios. Introducción a los objetos.](#)
- [Nagios. Macros globales.](#)
- [Nagios. Introducción a los plugins standard.](#)
- [Check\\_mk. Funcionalidades. La navaja Suiza para Nagios.](#) Introducción a las principales funcionalidades de check\_mk.
- [Check\\_mk. Chequeos. Primeros pasos.](#) Instalación de clientes y configuración mínima para el chequeo de estos.
- [Check\\_mk. Chequeos. Primeros pasos \(II\).](#) Configuración de chequeos adicionales básicos

O directamente puedes buscar encontrar lo que necesites en la [Guía de artículos](#).