

# Nagios : Quick install and configuration guide

Jonathan Lyard {jonathan.lyard@bull.net}

Version 0.1

April 14, 2006

## Abstract

Nagios is an efficient tool for network and system monitoring. However, first install and configuration is pretty intricate since it requires to understand a major part of the official documentation (250 pages). This tutorial is for ones who want to quickly build a working configuration so as to monitor simple networks including NFSv4 clients and servers. This guide goes through installation and basic configuration of Nagios.

## Contents

|          |   |          |
|----------|---|----------|
| <b>1</b> | <b>Install</b>                          | <b>2</b> |
| 1.1      | Copy distribution and plug in . . . . . | 2        |
| 1.2      | Create user . . . . .                   | 2        |
| 1.3      | Configuration and compilation . . . . . | 2        |
| <b>2</b> | <b>Configuration</b>                    | <b>4</b> |
| 2.1      | Time periods . . . . .                  | 4        |
| 2.2      | Contacts . . . . .                      | 5        |
| 2.3      | Hosts . . . . .                         | 5        |
| 2.4      | Hostgroups . . . . .                    | 6        |
| 2.5      | Services . . . . .                      | 6        |
| 2.6      | Web server configuration . . . . .      | 7        |
| <b>3</b> | <b>Run Nagios</b>                       | <b>9</b> |

---

# 1 Install

## 1.1 Copy distribution and plug in

First, you must get the latest Nagios release (*nagios-x.y.tar.gz*) and the official Nagios plugins (*nagios-plugins-x.y.z*) from <http://www.nagios.org/download/>. I also advise to get the official Nagios documentation (<http://www.nagios.org/docs/>) if you plan to tune the configuration explained here.

Unpack the tarball :

```
tar xzf nagios-2.0.tar.gz tar xzf nagios-plugins-1.4.2.tar.gz 1.2
```

## 1.2 Create user

First, we need to create the user Nagios will run under :

```
adduser nagios mkdir /usr/local/nagios
chown nagios.nagios /usr/local/nagios
```

You will probably want to issue external commands from the web interface. So you need to identify the user your web-server run as. For Apache, you can get it with :

```
grep "^User" /etc/httpd/conf/httpd.conf
```

By default, the user is Apache. See */etc/passwd* file or report to you web-server installation if you are running another web server. Also note that the path to *httpd.conf* can vary depending on your system.

We need to create a group whose members include the user your web server is running as and the user Nagios is running as.

```
/usr/sbin/groupadd nagcmd
/usr/sbin/usermod -G nagcmd apache
/usr/sbin/usermod -G nagcmd nagios
```

Recall to replace *apache* by the user name you obtained at last step.

## 1.3 Configuration and compilation

We can now configure and install Nagios. Go to the directory where you untared the distribution and run :

```
./configure --prefix=/usr/local/nagios --with-nagios-user=nagios
--with-nagios-group=nagios --with-command-group=nagios
```

Depending on your system, you should get a configuration like the following :

## Configuration summary for nagios 2.0 02-07-2006 \*\*\*:

|            |                                       |
|------------|---------------------------------------|
| General    | Options:                              |
| <hr/>      |                                       |
| Nagios     | executable: nagios                    |
| Nagios     | user/group: nagios,nagios             |
| Command    | user/group: nagios,nagios             |
| Embedded   | Perl: no                              |
| Event      | Broker: yes                           |
| Install    | \${prefix}: /usr/local/nagios         |
| Lock       | file: \${prefix}/var/nagios.lock      |
| Init       | directory: /etc/rc.d/init.d           |
| Host       | OS: linux-gnu                         |
| Web        | Interface Options:                    |
| <hr/>      |                                       |
| HTML       | URL: http://localhost/nagios/         |
| CGI        | URL: http://localhost/nagios/cgi-bin/ |
| Traceroute | (used by WAP): /bin/traceroute        |

Now let's compile and install :

```
make all
make install
make install-init
make install-config
```

`make install-init` allows to run nagios at startup (`/etc/rc.d/init.d/nagios`). This does not always work (I had problems on Fedora Core 4) but we see another way to run it at startup later on.

`make install-config` installs the sample configuration files. This will be useful to setup a basic configuration.

In order for nagios to be of any use to you, you need to install some plugins. First, install the official plugins. Go to the directory where you untared `nagios-plugin-x.y.z` and run :

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

## 2 Configuration

Normally, you should already have some configuration sample files in `/usr/local/nagios/etc`. Else, try to run again `make install-config`. Files should be called `xxxx.cfg-sample`, so you need to rename them :

```
cd /usr/local/nagios/etc
mv bigger.cfg-sample bigger.cfg
(...)
```

The entry point is through `nagios.cfg`. This file is run first when nagios starts. You may not need a lot of modifications in this file.

```
log_file=/usr/local/nagios/var/nagios.log
```

indicates where the logs will go. Set the appropriate owner of the log file :

```
chown nagios.nagios /usr/local/nagios/var/nagios.log
```

Other configuration files are called from `nagios.cfg`, for instance at lines :

```
cfg_file=/usr/local/nagios/etc/checkcommands.cfg
cfg_file=/usr/local/nagios/etc/misccommands.cfg
```

Those files contain commands definition.

And finally, our own configuration will be written in a new file (`ma_conf.cfg` for instance) so we need to replace :

```
cfg_file=/usr/local/nagios/etc/minimal.cfg
```

By :

```
cfg_file=/usr/local/nagios/etc/ma_conf.cfg
```

To create our configuration, we will take `minimal.cfg` as the basis :

```
cp minimal.cfg ma_conf.cfg
```

And now let's edit it to setup a configuration for a simple network with one monitoring station and one NFSv4 server.

### 2.1 Time periods

Time periods are used to define some monitoring periods. For the time being we can keep 24\*7 but you may need to change it -for instance not to receive alarms if your NFS servers are down during night.

## 2.2 Contacts

You will certainly need to add some contacts, which are persons in charge of monitoring the network. By default they will receive notification in 24\*7 by e-mail.

```
define contact{

contact_name adminNFS

    alias Jonathan Lyard
    service_notification_period 24x7
    host_notification_period 24x7
    service_notification_options w,u,c,r
    host_notification_options d,r
    service_notification_commands notify-by-email
    host_notification_commands host-notify-by-email
    email adminNFS@localhost

}
```

Now let's create a contact group. This one will be used to indicate which contacts should be notified for each issue. Here is only one group, but feel free to add all groups you need (for instance if the ones in charge of NFS administration are different from network administrators).

```
define contactgroup{

contactgroup_name admins

alias Nagios Administrators

members nagios-admin,adminNFS

}
```

## 2.3 Hosts

In addition to localhost, you must specify all hosts your are monitoring. I assume they are all on-link (in the same LAN as your monitoring station). Else, see Nagios official documentation to get it working through routers (use *parents* option).

Here is what is required in order to monitor a machine called nfs2 with IP address 192.168.0.5. Alarms from this host will be reported to the "admins" group.

```

define host{
    use generic-host ; Name of host template to use
    host_name nfs2
    alias nfs2
    address 192.168.0.5
    check_command check-host-alive
    max_check_attempts 10
    notification_interval 120
    notification_period 24x7
    notification_options d,r
    contact_groups admins
}

```

## 2.4 Hostgroups

In this basic configuration, we will only have one hostgroup. You can define several hostgroups to group hosts together in Nagios interface.

```

define hostgroup{
    hostgroup_name nfsv4config
    alias NFSv4 test configuration
    members localhost,nfs2
}

```

## 2.5 Services

Finally, one can find services definition. Services do not necessarily mean real services that run on host (FTP, POP, HTTP...) but also some other types of metric associated with the host (response to ping, number of logged-in user, free disk space...).

For instance, to monitor connectivity for localhost and our nfs2 server, we need the following service definition :

```

define service{

use      generic-service ; Name of service template to use
host_name localhost,nfs2

```

```
service_description PING
is_volatile 0
check_period 24x7
max_check_attempts 4
normal_check_interval 5
retry_check_interval 1
contact_groups admins
notification_options w,u,c,r
notification_interval 960
notification_period 24x7
check_command check_ping!100.0,20%!500.0,60%
}
```

Later on, we will also define a service to monitor NFSv4.

## 2.6 Web server configuration

In this step, you will learn how to configure the web-server. I assume you are using Apache, but the configuration may be similar for other web-servers. Check your own web-server's documentation to put it at work.

Add the following lines in *httpd.conf* (in general */etc/httpd/conf/httpd.conf*).

```
ScriptAlias /nagios/cgi-bin /usr/local/nagios/sbin
Directory /usr/local/nagios/sbin

#       SSLRequireSSL

       Options ExecCGI
       AllowOverride None
       Order allow,deny
       Allow from all
#       Order deny,allow
#       Deny from all
#       Allow from 127.0.0.1
```

```

    AuthName Nagios Access
    AuthType Basic
    AuthUserFile /usr/local/nagios/etc/htpasswd.users
    Require valid-user
    /Directory

Alias /nagios /usr/local/nagios/share
Directory /usr/local/nagios/share

#       SSLRequireSSL

    Options None
    AllowOverride None
    Order allow,deny
    Allow from all
    #       Order deny,allow
    #       Deny from all
    #       Allow from 127.0.0.1
    AuthName Nagios Access
    AuthType Basic
    AuthUserFile /usr/local/nagios/etc/htpasswd.users
    Require valid-user
    /Directory

```

This configuration enable authentication. The first directory entry is for CGI scripts, the second for html pages. Of course, you can tune this configuration to fit your requirements.

You can also disable authentication by setting *use\_authentication=0* in */usr/local/nagios/etc/cgi.cfg* but then, you will not be allowed to run remote commands. Use it only if you have a problem with authentication.

You now need to create one or more users being able to authenticate. For the first one (-c to create the file *htpasswd.users*) :

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

and for (possible) next users :

```
htpasswd /usr/local/nagios/etc/htpasswd.users nagiosadmin2
```

Then, you need to give some rights to the users.

You can do it by editing the file `/usr/local/nagios/etc/cgi.cfg`.

For instance :

```
authorized_for_system_information=nagiosadmin,nagiosadmin2
```

And so on for all line you want to grant privilege to *nagiosadmin* and *nagiosadmin2*. I have uncommented all *authorized* line and give access to *nagiosadmin*. I advised to at least do the same if want to be able to configure everything from the web interface.

For more details on the access rights, read the chapter "Authentication and Authorization in the CGIs" in Nagios official documentation.

Restart your web browser to take modifications into account :

```
/usr/sbin/httpd -k restart
```

You should be able to point `http://localhost` in your web browser proving that your web server is up.

You should also be able to point `http://localhost/nagios/` and authenticate yourself with one of the user created previously (*htpasswd*). At this time, the CGI will not works since Nagios is not running.

### 3 Run Nagios

To run Nagios, you can use the following command to make it start as a foreground process:

```
/usr/local/nagios/bin/nagios /usr/local/nagios/etc/nagios.cfg &
```

If make install-init failed to make Nagios run at startup you can add this command to `/etc/rc.d/rc.local` (here on FC4 but path and filename can vary depending on distribution).

To check that all configuration files are alright, you can issue :

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

Finally, a useful command to restart Nagios and therefore consider modifications in the configuration files :

```
killall -SIGHUP nagios
```

Enjoy monitoring systems with Nagios!

**Jonathan Lyard {jonathan.lyard@bull.net}**