Install Nagios 4

This section will cover how to install Nagios 4 on your monitoring server. You only need to complete this section once.

**Install Build Dependencies**

Because we are building Nagios Core from source, we must install a few development libraries that will allow us to complete the build.

First, install the required packages:

sudo yum install gcc glibc glibc-common gd gd-devel make net-snmp openssl-devel xinetd unzip

**Create Nagios User and Group**

We must create a user and group that will run the Nagios process. Create a "nagios" user and "nagcmd" group, then add the user to the group with these commands:

sudo useradd nagios

sudo groupadd nagcmd

sudo usermod -a -G nagcmd nagios

Let's install Nagios now.

**Install Nagios Core**

Download the source code for the latest stable release of Nagios Core. Go to the [Nagios downloads page](http://www.nagios.org/download/core-stay-informed), and click the **Skip to download** link below the form. Copy the link address for the latest stable release so you can download it to your Nagios server.

At the time of this writing, the latest stable release is Nagios 4.1.1. Download it to your home directory with curl:

cd ~

curl -L -O https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.1.1.tar.gz

Extract the Nagios archive with this command:

tar xvf nagios-\*.tar.gz

Then change to the extracted directory:

cd nagios-\*

Before building Nagios, we must configure it with this command:

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

Now compile Nagios with this command:

make all

Now we can run these make commands to install Nagios, init scripts, and sample configuration files:

sudo make install

sudo make install-commandmode

sudo make install-init

sudo make install-config

sudo make install-webconf

In order to issue external commands via the web interface to Nagios, we must add the web server user, apache, to the nagcmd group:

* sudo usermod -G nagcmd apache

**Install Nagios Plugins**

Find the latest release of Nagios Plugins here: [Nagios Plugins Download](http://nagios-plugins.org/download/?C=M;O=D). Copy the link address for the latest version, and copy the link address so you can download it to your Nagios server.

At the time of this writing, the latest version is Nagios Plugins 2.1.1. Download it to your home directory with curl:

cd ~

curl -L -O http://nagios-plugins.org/download/nagios-plugins-2.1.1.tar.gz

Extract Nagios Plugins archive with this command:

tar xvf nagios-plugins-\*.tar.gz

Then change to the extracted directory:

cd nagios-plugins-\*

Before building Nagios Plugins, we must configure it. Use this command:

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

Now compile Nagios Plugins with this command:

make

Then install it with this command:

sudo make install

**Install NRPE**

Find the source code for the latest stable release of NRPE at the [NRPE downloads page](http://sourceforge.net/projects/nagios/files/nrpe-2.x/). Download the latest version to your Nagios server.

At the time of this writing, the latest release is 2.15. Download it to your home directory with curl:

* cd ~
* curl -L -O http://downloads.sourceforge.net/project/nagios/nrpe-2.x/nrpe-2.15/nrpe-2.15.tar.gz

Extract the NRPE archive with this command:

* tar xvf nrpe-\*.tar.gz

Then change to the extracted directory:

* cd nrpe-\*

Configure NRPE with these commands:

* ./configure --enable-command-args --with-nagios-user=nagios --with-nagios-group=nagios --with-ssl=/usr/bin/openssl --with-ssl-lib=/usr/lib/x86\_64-linux-gnu

Now build and install NRPE and its xinetd startup script with these commands:

* make all
* sudo make install
* sudo make install-xinetd
* sudo make install-daemon-config

Open the xinetd startup script in an editor:

* sudo vi /etc/xinetd.d/nrpe

Modify the only\_from line by adding the private IP address of the your Nagios server to the end (substitute in the actual IP address of your server):

only\_from = 127.0.0.1 10.132.224.168

Save and exit. Only the Nagios server will be allowed to communicate with NRPE.

Restart the xinetd service to start NRPE:

* sudo service xinetd restart

Now that Nagios 4 is installed, we need to configure it.

Configure Nagios

Now let's perform the initial Nagios configuration. You only need to perform this section once, on your Nagios server.

**Organize Nagios Configuration**

Open the main Nagios configuration file in your favorite text editor. We'll use vi to edit the file:

sudo vi /usr/local/nagios/etc/nagios.cfg

Now find an uncomment this line by deleting the #:

#cfg\_dir=/usr/local/nagios/etc/servers

Save and exit.

Now create the directory that will store the configuration file for each server that you will monitor:

sudo mkdir /usr/local/nagios/etc/servers

**Configure Nagios Contacts**

Open the Nagios contacts configuration in your favorite text editor. We'll use vi to edit the file:

sudo vi /usr/local/nagios/etc/objects/contacts.cfg

Find the email directive, and replace its value (the highlighted part) with your own email address:

email nagios@localhost ; <<\*\*\*\*\* CHANGE THIS TO YOUR EMAIL ADDRESS \*\*\*\*\*\*

Save and exit.

**Configure check\_nrpe Command**

Let's add a new command to our Nagios configuration:

* sudo vi /usr/local/nagios/etc/objects/commands.cfg

Add the following to the end of the file:

define command{

command\_name check\_nrpe

command\_line $USER1$/check\_nrpe -H $HOSTADDRESS$ -c $ARG1$

}

Save and exit. This allows you to use the check\_nrpe command in your Nagios service definitions.

**Configure Apache**

Use htpasswd to create an admin user, called "nagiosadmin", that can access the Nagios web interface:

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

Enter a password at the prompt. Remember this login, as you will need it to access the Nagios web interface.

**Note:** If you create a user that is not named "nagiosadmin", you will need to edit /usr/local/nagios/etc/cgi.cfg and change all the "nagiosadmin" references to the user you created.

Nagios is ready to be started. Let's do that, and restart Apache:

sudo systemctl daemon-reload

sudo systemctl start nagios.service

sudo systemctl restart httpd.service

To enable Nagios to start on server boot, run this command:

sudo chkconfig nagios on

Optional: Restrict Access by IP Address

If you want to restrict the IP addresses that can access the Nagios web interface, you will want to edit the Apache configuration file:

sudo vi /etc/httpd/conf.d/nagios.conf

Find and comment the following two lines by adding # symbols in front of them:

Order allow,deny

Allow from all

Then uncomment the following lines, by deleting the # symbols, and add the IP addresses or ranges (space delimited) that you want to allow to in the Allow from line:

# Order deny,allow

# Deny from all

# Allow from 127.0.0.1

As these lines will appear twice in the configuration file, so you will need to perform these steps once more.

Save and exit.

Now start Nagios and restart Apache to put the change into effect:

sudo systemctl restart nagios.service

sudo systemctl restart httpd.service

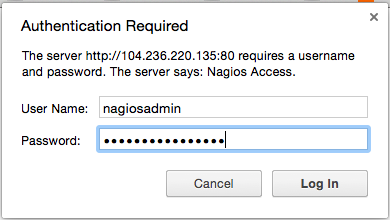
Nagios is now running, so let's try and log in.

Accessing the Nagios Web Interface

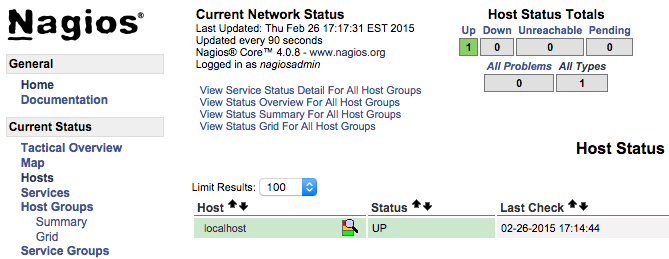
Open your favorite web browser, and go to your Nagios server (substitute the IP address or hostname for the highlighted part):

http://nagios\_server\_public\_ip/nagios

Because we configured Apache to use htpasswd, you must enter the login credentials that you created earlier. We used "nagiosadmin" as the username:



After authenticating, you will be see the default Nagios home page. Click on the **Hosts** link, in the left navigation bar, to see which hosts Nagios is monitoring:



As you can see, Nagios is monitoring only "localhost", or itself.

Let's monitor another host with Nagios!

Monitor a CentOS 7 Host with NRPE

In this section, we'll show you how to add a new host to Nagios, so it will be monitored. Repeat this section for each CentOS or RHEL server you wish to monitor.

**Note:** If you want to monitor an Ubuntu or Debian server, follow the instructions in this link: [Monitor an Ubuntu Host with NRPE](https://www.digitalocean.com/community/tutorials/how-to-install-nagios-4-and-monitor-your-servers-on-ubuntu-14-04#monitor-an-ubuntu-host-with-nrpe).

On a server that you want to monitor, install the EPEL repository:

sudo yum install epel-release

Now install Nagios Plugins and NRPE:

sudo yum install nrpe nagios-plugins-all

Now, let's update the NRPE configuration file. Open it in your favorite editor (we're using vi):

sudo vi /etc/nagios/nrpe.cfg

Find the allowed\_hosts directive, and add the private IP address of your Nagios server to the comma-delimited list (substitute it in place of the highlighted example):

allowed\_hosts=127.0.0.1,10.132.224.168

Save and exit. This configures NRPE to accept requests from your Nagios server, via its private IP address.

Restart NRPE to put the change into effect:

sudo systemctl start nrpe.service

sudo systemctl enable nrpe.service

Once you are done installing and configuring NRPE on the hosts that you want to monitor, you will have to add these hosts to your Nagios server configuration before it will start monitoring them.

**Add Host to Nagios Configuration**

On your Nagios server, create a new configuration file for each of the remote hosts that you want to monitor in /usr/local/nagios/etc/servers/. Replace the highlighted word, "yourhost", with the name of your host:

sudo vi /usr/local/nagios/etc/servers/yourhost.cfg

Add in the following host definition, replacing the host\_name value with your remote hostname ("web-1" in the example), the alias value with a description of the host, and the address value with the private IP address of the remote host:

define host {

use linux-server

host\_name yourhost

alias My first Apache server

address 10.132.234.52

max\_check\_attempts 5

check\_period 24x7

notification\_interval 30

notification\_period 24x7

}

With the configuration file above, Nagios will only monitor if the host is up or down. If this is sufficient for you, save and exit then restart Nagios. If you want to monitor particular services, read on.

Add any of these service blocks for services you want to monitor. Note that the value of check\_command determines what will be monitored, including status threshold values. Here are some examples that you can add to your host's configuration file:

Ping:

define service {

use generic-service

host\_name yourhost

service\_description PING

check\_command check\_ping!100.0,20%!500.0,60%

}

SSH (notifications\_enabled set to 0 disables notifications for a service):

define service {

use generic-service

host\_name yourhost

service\_description SSH

check\_command check\_ssh

notifications\_enabled 0

}

If you're not sure what use generic-service means, it is simply inheriting the values of a service template called "generic-service" that is defined by default.

Now save and quit. Reload your Nagios configuration to put any changes into effect:

sudo systemctl reload nagios.service

Once you are done configuring Nagios to monitor all of your remote hosts, you should be set. Be sure to access your Nagios web interface, and check out the **Services** page to see all of your monitored hosts and services:

