Before beginning the work on Nagios you need to install Nagios

**Section 1 – Nagios Installation**

1. You must have superuser privileges on the Ubuntu 14.04 server that will run Nagios. Ideally, you will be using a non-root user with superuser privileges.
2. A LAMP stack is also required.
3. **Install Apache**

sudo apt-get update

sudo apt-get install apache2

You can do a spot check right away to verify that everything went as planned by visiting your server's public IP address in your web browser (see the note under the next heading to find out what your public IP address is if you do not have this information already):

http://your\_server\_IP\_address

You will see the default Ubuntu 14.04 Apache web page

1. **Install MySQL**

sudo apt-get install mysql-server php5-mysql

sudo mysql\_install\_db

sudo mysql\_secure\_installation

You will be asked to enter the password you set for the MySQL root account.

1. **Install PHP**

sudo apt-get install php5 libapache2-mod-php5 php5-mcrypt

sudo nano /etc/apache2/mods-enabled/dir.conf

It will look like this:

<IfModule mod\_dir.c>

DirectoryIndex index.html index.cgi index.pl index.php index.xhtml index.htm

</IfModule>

We want to move the PHP index file highlighted above to the first position after the DirectoryIndexspecification, like this:

<IfModule mod\_dir.c>

DirectoryIndex index.php index.html index.cgi index.pl index.xhtml index.htm

</IfModule>

After this, we need to restart the Apache web server in order for our changes to be recognized. You can do this by typing this:

sudo service apache2 restart

1. **Install PHP Modules**

sudo apt-get install php5-cli

1. **Install Nagios**

### 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

### 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. While we're at it, we will also install apache2-utils, which will be used to set up the Nagios web interface.

First, update your apt-get package lists:

sudo apt-get update

Then install the required packages:

sudo apt-get install build-essential libgd2-xpm-dev openssl libssl-dev xinetd apache2-utils unzip

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. If you want to configure it to use postfix (which you can install with apt-get), add --with-mail=/usr/sbin/sendmail to the following command:

* ./configure --with-nagios-group=nagios --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 /usr/bin/install -c -m 644 sample-config/httpd.conf /etc/apache2/sites-available/nagios.conf

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

* sudo usermod -G nagcmd www-data

### 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

Enable the Apache rewrite and cgi modules:

sudo a2enmod rewrite

sudo a2enmod cgi

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 password, 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.

Now create a symbolic link of nagios.conf to the sites-enabled directory:

sudo ln -s /etc/apache2/sites-available/nagios.conf /etc/apache2/sites-enabled/

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

sudo service nagios start

sudo service apache2 restart

In case Nagios service does not start follow these steps:

sudo vi /etc/systemd/system/nagios.service

and added the following:

[Unit]

Description=Nagios

BindTo=network.target

[Install]

WantedBy=multi-user.target

[Service]

User=nagios

Group=nagios

Type=simple

ExecStart=/usr/local/nagios/bin/nagios /usr/local/nagios/etc/nagios.cfg

Then the following:

sudo systemctl enable /etc/systemd/system/nagios.service

sudo systemctl start nagios

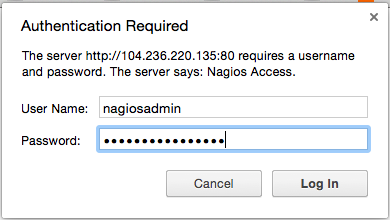
sudo systemctl restart nagios

## **Accessing the Nagios Web Interface**

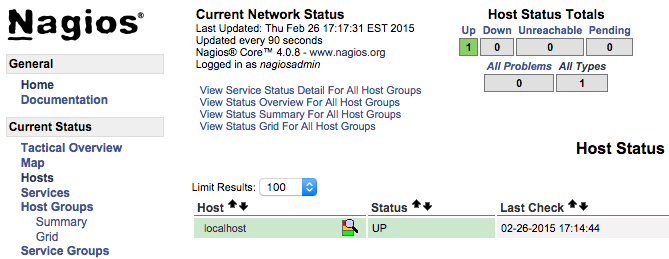
Open your favorite web browser, and go to your Nagios server (substitute the IP address or hostname for the highlighted part):

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 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 server you wish to monitor.

On a server that you want to monitor, update apt-get:

sudo apt-get update

Now install Nagios Plugins and NRPE:

sudo apt-get install nagios-plugins nagios-nrpe-server

### Configure Allowed Hosts

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

sudo vi /etc/nagios/nrpe.cfg

... and add Nagios Server IP *192.168.1.9* to the *server\_address*.

*server\_address=192.168.1.9*

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.

### Configure Allowed NRPE Commands

Look up the name of your root filesystem (because it is one of the items we want to monitor):

df -h /

We will be using the filesystem name in the NRPE configuration to monitor your disk usage (it is probably /dev/vda). Now open nrpe.cfg for editing:

sudo vi /etc/nagios/nrpe.cfg

The NRPE configuration file is very long and full of comments. There are a few lines that you will need to find and modify:

* **server\_address**: Set to the private IP address of this host
* **allowed\_hosts**: Set to the private IP address of your Nagios server
* **command[check\_hda1]**: Change /dev/hda1 to whatever your root filesystem is called

The three aforementioned lines should look like this (substitute the appropriate values):

server\_address=client\_private\_IP

allowed\_hosts=nagios\_server\_private\_IP

command[check\_hda1]=/usr/lib/nagios/plugins/check\_disk -w 20% -c 10% -p /dev/vda

Note that there are several other "commands" defined in this file that will run if the Nagios server is configured to use them. Also note that NRPE will be listening on port 5666 because server\_port=5666 is set. If you have any firewalls blocking that port, be sure to open it to your Nagios server.

Save and quit.

### Restart NRPE

Restart NRPE to put the change into effect:

sudo service nagios-nrpe-server restart

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

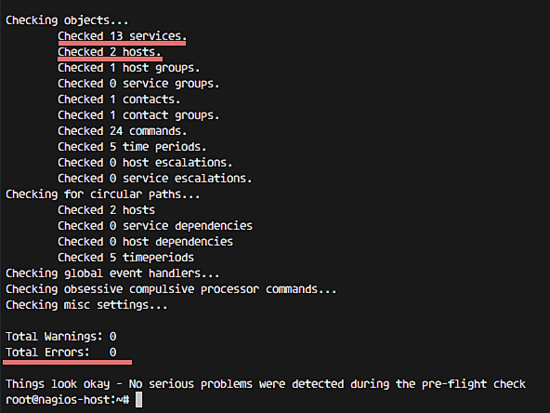
}

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.

And now check the configuration:

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

... to see if the configuration is correct.

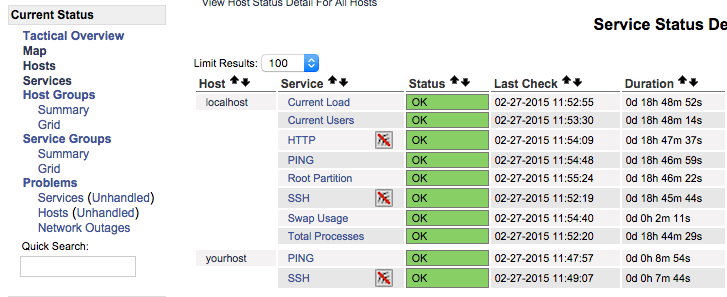
[](https://www.howtoforge.com/images/how_to_install_nagios_on_ubuntu_16_04/big/8.png)

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

*Sudo service apache2 restart  
sudo service nagios restart*

sudo service nagios reload

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:



**Section 2 – Capstone Exercise**

**Step 1**

Start your Google Compute Engine virtual machine and connect to it using SSH.

sudo ls /usr/local/nagios/etc/servers

sudo nano /usr/local/nagios/etc/servers/devops-host.cfg

**Step 2**

Add code

define service {

use generic-service

host\_name devops-host

service\_description CHECKJAVA

check\_command check\_java

notifications\_enabled 0

}

define service {

use generic-service

host\_name devops-host

service\_description CHECKTOMCAT

check\_command check\_tomcat

notifications\_enabled 0

}

define service {

use generic-service

host\_name devops-host

service\_description CHECKTOMCATAPP

check\_command check\_TomcatApplication -u admin -p admin -h http://localhost -P $

notifications\_enabled 0

}

**Step 3**

Save the file and exit.

**Step 4**

sudo service apache 2 restart

sudo service nagios restart

**Step 5**

Go to your browser and open the Nagios page

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