**Nagios Installation**

**Step 1:-** Install Apache and Php first.

**Step 2:-** Install pre-required packages and libraries

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

**Step 3:-** Create nagios user and group

#useradd nagios

#groupadd nagcmd

#usermod -a -G nagcmd nagios

**Step 4:-** Install Nagios core

#cd ~

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

#tar xvf nagios-\*.tar.gz

#cd nagios-\*

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

#make all

#make install

#make install-commandmode

#make install-init

#make install-config

#make install-webconf

**Step 5:-** Add apache user to nadcmd group

#usermod -G nagcmd apache

**Step 6:-** Now install nagios plugin

#cd ~

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

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

#cd nagios-plugins-\*

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

#make

#make install

**Step 7:-** Install nrpe

#cd ~

#curl -L -O <http://downloads.sourceforge.net/project/nagios/nrpe-2.x/nrpe-2.15/nrpe-2.15.tar.gz>

#tar xvf nrpe-\*.tar.gz

#cd nrpe\*

#./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

#make all

#make install

#make install-xinetd

#make install-daemon-config

#sudo vi /etc/xinetd.d/nrpe

**Note:-** change the option “allow from” and give the nagios server IP

**only\_from = 127.0.0.1 10.132.224.168**

#service xinetd restart

All has been done now configure the nagion accordingly

**Step 8:-** Configure nagios

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

uncomment the below line...

**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:

#mkdir /usr/local/nagios/etc/servers

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

#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**

save and exit the file.

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

#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.

**Step 9:-** Configure Apache

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.

Now create a file which will make nagios service start/stop/restart

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

Add the following lines.

**[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**

save and close the file.

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

#systemctl start nagios.service

#systemctl enable nagios.service

#systemctl restart httpd.service

#chkconfig nagios on

**Step 10:-**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:

#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:

#systemctl restart nagios.service

#systemctl restart httpd.service

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

**Step 11:-** Accessing the Nagios Web Interface

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

**Note:-** It's time configure clients Linux/Windows

**Step 12:-** Monitor a CentOS 7 Host with NRPE

#yum install epel-release

#yum install nrpe nagios-plugins-all

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

#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:

#systemctl start nrpe.service

#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.

Note:- Now again come to the nagios server and make the clients.

**Step 13:-**Add Host to Nagios Configuration

#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:

#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:

Note:- With the following procedure we can add the clients.

#vi /usr/local/nagios/etc/servers/clients.cfg

add the following lines.

**define host{**

**use linux-server**

**host\_name client**

**alias client**

**address 192.168.1.152**

**max\_check\_attempts 5**

**check\_period 24x7**

**notification\_interval 30**

**notification\_period 24x7**

**}**

Here, **192.168.1.152** is my nagios client IP address. Finally restart nagios service.

#systemctl restart nagios.service

**Step 14:-** We have just defined the monitoring host. Now, let us add some services of the monitoring host. For example, to monitor the **ssh** service, add the following lines shown in bold in the“**/usr/local/nagios/etc/servers/clients.cfg”** file.

#vi /usr/local/nagios/etc/servers/clients.cfg

Add the following lines shown in bold:

**define service {**

**use generic-service**

**host\_name client**

**service\_description SSH**

**check\_command check\_ssh**

**notifications\_enabled 0**

**}**

Save and close the file. Restart Nagios.

#systemctl restart nagios.service

Wait for few seconds, and check for the added services (i.e ssh) in the nagios web interface. Navigate to **Services** section on the left side bar, you’ll see the **ssh** service there.

**Steps to configure Nagiosgraph with Nagios Core**

**Step 1:-** First up all we have to download the latest version of Nagiosgraph from its website at http://nagiosgraph.sourceforge.net/, directly onto your monitoring server, using a tool such as wget:

#cd ~

#wget <http://downloads.sourceforge.net/project/nagiosgraph/nagiosgraph/1.5.2/nagiosgraph-1.5.2.tar.gz>

**Step 2:-** After the download completes then inflate the .tar.gz file and change to the directory within it:

**#tar -xf nagiosgraph-1.5.2.tar.gz**

**#cd nagiosgraph-1.5.2**

**Step 3:-** Install the pre required packages

**#yum install perl-CGI perl-rrd\* perl-GD\***

**#cpan Digest::MD5**

**#cpan Module::Build**

**#cpan Nagios::Config**

**Step 4:-** Nagiosgraph requires some dependency application to run. As the root user, run the install.pl script with the --check-prereq option. This will give you a survey of any dependencies you may need to install via packages or CPAN. When you have installed all the prerequisites, the output should look similar to the following code snippet:

**#./install.pl –check-prereq**

It will show the following:-

checking required PERL modules

Carp...1.11

CGI...3.43

Data::Dumper...2.124

File::Basename...2.77

File::Find...1.14

MIME::Base64...3.08

POSIX...1.17

RRDs...1.4003

Time::HiRes...1.9719

checking optional PERL modules

GD...2.39

checking nagios installation

found nagios at /usr/local/nagios/bin/nagios

checking web server installation

found apache at /usr/sbin/apache2

These are all reasonably standard Perl libraries, so don’t forget to check if packages are available for them before you resort to using CPAN. For example, I was able to install the RRDs and GD modules on my Debian system as follows:

**# apt-get install librrds-perl libgd-gd2-per**

If you are having trouble getting **install.pl** to find your Nagios Core or Apache HTTPD instances, then take a look at the output of **install.pl** –help to run an installation specific to your kind of system. This is documented in more detail in the INSTALL file.

Step 5:-After getting the sucessfull message from **./install.pl --check-prereq** command now we will move forwar to install the nagiosgraph on the machine. As the root user, run the **install.pl** script with the **--install**  argument. You will be prompted many times for directory layout options. The default is shown in square brackets and should be correct for a typical Nagios Core installation, so to start with, simply press Enter on each option. If you are an advance user of Nagios or you have installed Nagios in any other location then you can modify the path values during the nagiosgraph installation.

**# ./install.pl --install**

...

Destination directory (prefix)? [/usr/local/nagiosgraph]

Location of configuration files (etc-dir)? [/usr/local/

nagiosgraph/etc]

Location of executables? [/usr/local/nagiosgraph/bin]

Location of CGI scripts? [/usr/local/nagiosgraph/cgi]

Location of documentation (doc-dir)? [/usr/local/nagiosgraph/doc]

Location of examples? [/usr/local/nagiosgraph/examples]

Location of CSS and JavaScript files? [/usr/local/nagiosgraph/

share]

Location of utilities? [/usr/local/nagiosgraph/util]

Location of state files (var-dir)? [/usr/local/nagiosgraph/var]

Location of RRD files? [/usr/local/nagiosgraph/var/rrd]

Location of log files (log-dir)? [/usr/local/nagiosgraph/var]

Path of log file? [/usr/local/nagiosgraph/var/nagiosgraph.log]

Path of CGI log file? [/usr/local/nagiosgraph/var/nagiosgraph-cgi.

log]

URL of CGI scripts? [/nagiosgraph/cgi-bin]

URL of CSS file? [/nagiosgraph/nagiosgraph.css]

URL of JavaScript file? [/nagiosgraph/nagiosgraph.js]

Path of Nagios performance data file? [/tmp/perfdata.log]

URL of Nagios CGI scripts? [/nagios/cgi-bin]

username or userid of Nagios user? [nagios]

username or userid of web server user? [www-data]

Modify the Nagios configuration? [n]

Modify the Apache configuration? [n]

…

After the preceding selections are all made, the files should be installed with appropriate permissions set. The final part of the output gives instructions for adding configuration to Nagios Core and Apache HTTPD, which we’ll do next.

**Step 6:-** Now we will edit some nagios configuration files to make nagiosgraph functional. Change to the Nagios Core configuration directory. As per the quick start guide installation the default nagios configuration location should be : **/usr/local/nagios/etc** .

**# cd /usr/local/nagios/etc**

**Step 7:-** Now we will edit the core configuration file of nagios to append some values to it. Edit the core configuration file **nagios.cfg**, and add the following directives at the end of the file:

# process nagios performance data using nagiosgraph

**process\_performance\_data=1**

**service\_perfdata\_file=/tmp/perfdata.log**

**service\_perfdata\_file\_template=$LASTSERVICECHECK$||$HOSTNAME$||$SERVICEDESC$||$SERVICEOUTPUT$||$SERVICEPERFDATA$**

**service\_perfdata\_file\_mode=a**

**service\_perfdata\_file\_processing\_interval=30**

**service\_perfdata\_file\_processing\_command=process-service-perfdata-for-nagiosgraph**

**Step 8:-** Now move to the Nagios Core objects configuration directory. As per the quick start guide installation the default nagios core object configuration location should be : **/usr/local/nagios/etc/objects.**

# cd /usr/local/nagios/etc/objects

**Step 9:-** Edit the commands.cfg file, and add the following command definition:

# command to process nagios performance data for nagiosgraph

**define command {**

**command\_name process-service-perfdata-for-nagiosgraph**

**command\_line /usr/local/nagiosgraph/bin/insert.pl**

**}**

Step 10:- After installing the nagiosgraph you should have an Apache configuration file for nagiosgraph inside the **/usr/local/nagiosgraph/etc/** directory. We have to include that file to our Apache configuration file to make it functional. Edit the httpd.conf file for your Apache HTTPD server to include the following line at the end:

**Include /usr/local/nagiosgraph/etc/nagiosgraph-apache.conf**

In a local install of Apache HTTPD, this file is normally in **/usr/local/apache/conf/httpd.conf**, but its location varies by system. On Debian-derived systems it may be **/etc/apache2/apache2.conf.**

**Step 11:-** Validate the configuration of both the Apache HTTPD server and the Nagios Core server, and restart them both:

**# /usr/local/apache/bin/apachectl configtest**

**# /usr/local/apache/bin/apachectl restart**

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

**# /etc/init.d/nagios restart**

Make sure that your SELiux should be in Disabled or Permissive mode to work nagiosgraph properly.

Note:- Make the following changes in **“vi /usr/local/nagiosgraph/etc/nagiosgraph-apache.conf”** at both the places

**Allow from all**

**Require all granted**

**Step 12:-** Visit **http://<Your Nagioshost>/nagiosgraph/cgi-bin/showconfig.cgi** in your browser, substituting your own Nagios Core server’s hostname, to test that everything’s working. You should see a long page with configuration information for Nagiosgraph:

**Step 13:-** If everything is working up to this point, the only thing left to do is to define an action URL for the services that you want to graph, so that you can click to go directly to the graphs for that service from the Nagios Core web interface. The tidiest and most straightforward way to do this is to define a service template. Edit the templetes.cfg file to append the new service template inside it.

**#vi /usr/local/nagios/etc/objects/templates.cfg**

Append the below code to the bottom of that file.

**define service {**

**name nagiosgraph**

**action\_url /nagiosgraph/cgi-bin/show.cgi?host=$HOSTNAME$&service=$SERVICEDESC$**

**register 0**

**}**

Then, you can have the services you want graphed inherit from it, as well as from any other templates they use, by adding nagiosgraph to the value for the use directive:

**define service {**

**use generic-service,nagiosgraph**

**host\_name corinth.naginet**

**service\_description PING**

**check\_command check\_ping!100,10%!200,20%**

**}**

You should do this for all the services for which you want graphing.

**Step 14:-** Validate the configuration and restart the Nagios Core server again:

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

**# /etc/init.d/nagios restart**

With this done, visiting the Service section of the web interface should include action icons after each graphed service:

#####The END#####