

**Project : “SERVERS MONITORING SYSTEM USING NAGIOS CORE 4.1.1 (Full Organizational Deployment for monitoring all the development and prod servers of an organization)”**

  Made by : Apurv Agarwal

**Scope:**  
1. Deploy a Nagios Server for centralized view of the entire monitored IT infrastructure through web interface.  
  
2. Install Nagios Client on different servers covering below functionalities :--  
  
- Metrics capture every scheduled interval like for CPU/RAM utilization or Free disk space etc.  
- Detection of infrastructure outages like ping response failure etc.  
- Detection of outages for custom Application/Apis running on server.  
- Mail Alerts via Email/Sms for outages/disruptions.  
- Service Monitoring like for Mysql, Tomcat, Nginx and other services running on the servers.  
- Event handlers that allow automatic restart of failed applications and services.  
- Scheduled downtime mechanism for alert suppression during infrastructure upgrades.

**Introduction**

In this Documentation, we will cover the installation of Nagios 4, a very popular open source monitoring system, on CentOS 7 or RHEL 7. We will cover some basic configuration, so you will be able to monitor host resources via the web interface. We will also utilize the Nagios Remote Plugin Executor (NRPE), that will be installed as an agent on remote hosts, to monitor their local resources.

Nagios is useful for keeping an inventory of your servers, and making sure your critical services are up and running. Using a monitoring system, like Nagios, is an essential tool for any production server environment.

**Prerequisites**

To follow this doc, you must have root privileges on the CentOS 7 server that will run Nagios. Ideally, you will be using a root user with all privileges.

Now that we have the prerequisites sorted out, let’s move on to getting Nagios 4 installed.

Here we are using two servers, one as the main monitoring server and other as the client:

Monitoring server IP : 139.59.95.156

Client IP : 142.93.212.56

**Basic Installation**

**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:

#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:

useradd nagios

groupadd nagcmd

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. 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:

make install

make install-commandmode

make install-init

make install-config

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:

* 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
* make install
* make install-xinetd
* make install-daemon-config

Open the xinetd startup script in an editor:

* 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 = 127.0.0.1 139.59.95.156

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

Restart the xinetd service to start NRPE:

* 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:

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:

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:

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:

* 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:

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:

systemctl daemon-reload

systemctl start nagios.service

systemctl restart httpd.service

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

chkconfig nagios on

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

systemctl restart nagios.service

systemctl restart httpd.service

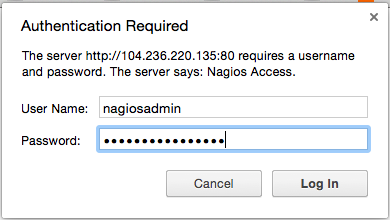
Nagios is now running, so let’s try and log in.

## Accessing the Nagios Web Interface

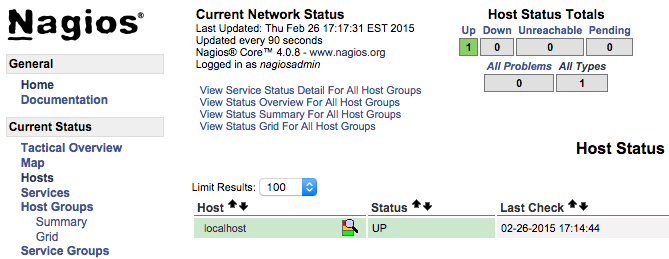
Open your favorite web browser, and go to your Nagios server (substitute the IP address or hostname for the highlighted part):

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 Client 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.

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

yum install epel-release

Now install Nagios Plugins and NRPE:

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=127.0.0.1,139.59.95.156

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.

### Add Client to Nagios Configuration

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

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 142.93.212.56

max\_check\_attempts 2

check\_period 24x7

notification\_interval 1

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

}

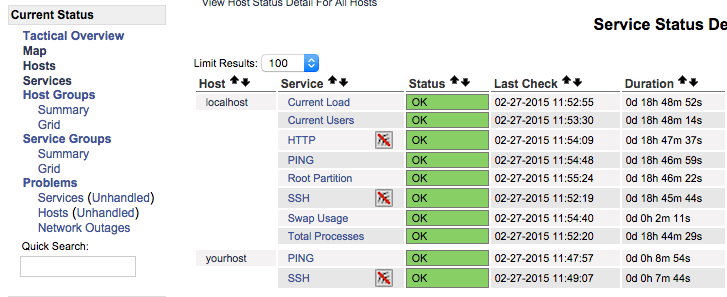
If you’re not sure what use generic-service means, it is simply inheriting the values of a service template called “generic-service” that is defined by default. It is being defined at

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

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:



The Basic installation has been successfully done till here.

The **Server & the Client** are now visible on the Nagios core Web UI.

Now we will look upon the deployment of various essential services to be monitored on the client by the nagios server.