

How to Install & Configure Nagios Core in Remote Server ::

Ref. Links:

<https://www.digitalocean.com/community/tutorials/how-to-install-nagios-4-and-monitor-your-servers-on-centos-7>

In this tutorial, we will cover the installation of Nagios 4, a very popular open source monitoring system, on CentOS 7 or RHEL 7.

Prerequisites:

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

Note: Before install Nagios, we have to install **LAMP Stack** on the Server.

```
$ sudo yum install -y httpd24 mysql56-server php70 php70-mysqld  
$ sudo service httpd start
```

Install Nagios 4:

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

Step:1: 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
```

Step:2: 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, and 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.

Step:3: Install Nagios Core

Download Nagios 4.1.1. (latest stable release) to your home directory with curl and extract

```
$ cd ~
$ curl -L -O https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.1.1.tar.gz
$ tar xvf nagios-*.tar.gz
```

Before building Nagios, we must configure it with this command

```
$ cd nagios-*
$ ./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
```

Step:4: Install Nagios Plugins:

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
$ tar xvf nagios-plugins-*.tar.gz
```

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

```
$ cd nagios-plugins-*
$ ./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
```

Step:5: Install NRPE:

At the time of this writing, the latest release of NRPE 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  
$ tar xvf nrpe-*.tar.gz
```

Configure NRPE with these commands:

```
$ 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
```

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 your Nagios server to the end (substitute in the actual IP address of your server):

```
$ only_from = 127.0.0.1 (replace your server Private IP Address)
```

Restart the xinetd service to start NRPE:

```
$ sudo service xinetd restart
```

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

===== Nagios Installation Over =====

Configure Nagios:

Now let's perform the initial Nagios configuration on your Nagios server.

Step:1: 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
```

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

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

Now create new file(servername.cfg) and add the server info. :

```
$ sudo vi /usr/local/nagios/etc/servers/servername.cfg
```

if you want add a new server in nagios : add the following code into WAC-mynewserver.cfg

```
define host {  
    use                linux-server  
    host_name          WAC-mynewserver  
    alias              WAC-mynewserver  
    address            54.161.194.240    #Public IP of Your Agent  
    max_check_attempts 5  
    check_period       24x7  
    notification_interval 30  
    notification_period 24x7  
}
```

Step:2: 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 *****
```

if you want to add more than one contacts add a tag like below ::

```
define contact{
    contact_name      nagiosadmin_3          ; Short name of user
    use                generic-contact        ; Inherit default values from generic-contact template (defined above)
    alias              Nagios Admin           ; Full name of user
    email              christopher@webappclouds.com
    service_notification_commands  notify-service-by-email
    host_notification_commands      notify-host-by-email
}
```

Step:3: 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.

Step:4: 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 service nagios start  
$ sudo service httpd start
```

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

```
$ sudo chkconfig nagios on
```

Step:5: Accessing the Nagios Web Interface:

```
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 seeing the default Nagios home page. Click on the Hosts link, in the left navigation bar, to see which hosts Nagios is monitoring:

Step:6: Add the services, which you want to monitor to Nagios Server:

If you need alerts for HTTP and SSH please comment below line in command.cfg

```
$ sudo vi /usr/local/nagios/etc/objects/commands.cfg
```

if you want add the services to monitor, add the below content to commands.cfg

```
$ define service {
    use                generic-service
    host_name          WAC-mynewserver
    service_description PING
    check_command       check_ping!100.0,20%!500.0,60%
}

define service {
    use                generic-service
    host_name          WAC-mynewserver
    service_description SSH
    check_command       check_ssh
#    notifications_enabled    0
}

define service{
    use                local-service        ; Name of service template to use
    host_name          WAC-mynewserver
    service_description HTTP
    check_command       check_http
#    notifications_enabled    0
}
```

To remember the paths:

Main configuration files of nagios:

```
Add servers in Nagios server (Nagios server ) :: /usr/local/nagios/etc/servers/servername.cfg

Add services to check in host server(Nagios server) : vi /usr/local/nagios/etc/objects/commands.cfg

Add your metrics to check remote server and restart service :: /usr/local/nagios/etc/objects/commands.cfg

Add contacts in :: /usr/local/nagios/etc/objects/contacts.cfg

/usr/local/nagios/etc/objects : contacts , localhost, commands

Nagios log file : /usr/local/nagios/var/nagios.log
```

Errors:

<https://stackoverflow.com/questions/23673549/nagios-http-warning-http-1-1-403-forbidden>

Nagios - HTTP WARNING: HTTP/1.1 403 Forbidden

```
$ sudo /usr/local/nagios/libexec/check_http -H 54.89.9.28 -e 'HTTP' --authorization=username:password
```