# How To Install Nagios 4 and Monitor Your Servers on Ubuntu 14.04

**Introduction**

In this tutorial, we will cover the installation of Nagios 4, a very popular open source monitoring system, on Ubuntu 14.04. 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), which will be installed as an agent on remote hosts, to monitor their local resources.

Nagios is the most popular, open source, powerful monitoring system for any kind of infrastructure. It enables organizations to identify and resolve IT infrastructure problems before they affect critical business processes. 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 tutorial, 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

Follow below process on how To Install Linux, Apache, MySQL, PHP (LAMP) stack on Ubuntu 14.04.

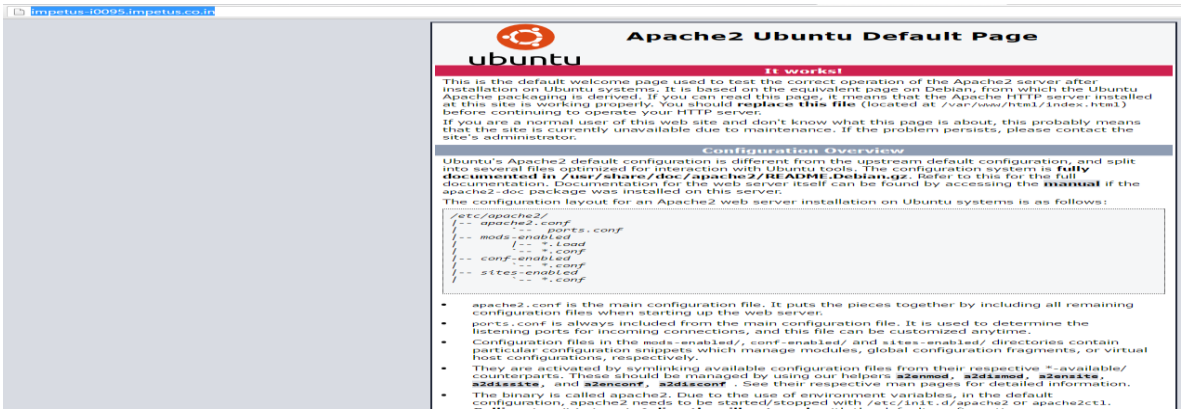
A "LAMP" stack is a group of open source software that is typically installed together to enable a server to host dynamic websites and web apps. This term is actually an acronym which represents the **L**inux operating system, with the **A**pache web server. The site data is stored in a **M**ySQL database, and dynamic content is processed by **P**HP.

**Step 1: Install Apache**

```
sudo apt-get update
sudo apt-get install apache2
```

Afterwards, your web server is installed. You can do a spot check right away to verify if it's installed correctly. http://impetus-i0095.impetus.co.in/

You will see the default Ubuntu 14.04 Apache web page, which is there for informational and testing purposes. It should look something like this



**Step 2: Install MySQL**

MySQL is a database management system. Basically, it will organize and provide access to databases where our site can store information.

```
sudo apt-get install mysql-server php5-mysql
```

During the installation, your server will ask you to select and confirm a password for the MySQL "root" user. This is an administrative account in MySQL that has increased privileges. Think of it as being similar to the root account for the server itself (the one you are configuring now is a MySQL-specific account however).

When the installation is complete, we need to run some additional commands to get our MySQL environment set up securely.

First, we need to tell MySQL to create its database directory structure where it will store its information. You can do this by typing:

```
sudo mysql_install_db
```

Afterwards, we want to run a simple security script that will remove some dangerous defaults and lock down access to our database system a little bit. Start the interactive script by running:

```
sudo mysql_secure_installation
```

You will be asked to enter the password you set for the MySQL root account. Next, it will ask you if you want to change that password. If you are happy with your current password, type "n" for "no" at the prompt.

For the rest of the questions, you should simply hit the "ENTER" key through each prompt to accept the default values. This will remove some sample users and databases, disable remote root logins, and load these new rules so that MySQL immediately respects the changes we have made.

At this point, your database system is now set up and we can move on.

**Step 3: Install PHP**

PHP is the component of our setup that will process code to display dynamic content. It can run scripts, connect to our MySQL databases to get information, and hand the processed content over to our web server to display.

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

This should install PHP without any problems. We'll test this in a moment.

In most cases, we'll want to modify the way that Apache serves files when a directory is requested. Currently, if a user requests a directory from the server, Apache will first look for a file called index.html. We want to tell our web server to prefer PHP files, so we'll make Apache look for an index.php file first.

To do this, type this command to open the dir.conf file in a text editor with root privileges:

```
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 `DirectoryIndex` specification, like this:

```
<IfModule mod_dir.c>
    DirectoryIndex index.php index.html index.cgi index.pl index.xhtml index.htm
</IfModule>
```

When you are finished, save and close the file by pressing "CTRL-X". You'll have to confirm the save by typing "Y" and then hit "ENTER" to confirm the file save location.

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
```

At this point, your LAMP stack is installed and configured. We should still test out our PHP though.

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

**Install Nagios 4**

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

**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.

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

**Install Nagios Core**

Download the source code for the latest stable release of Nagios Core. Go to the Nagios downloads page, 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.

```
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 install postfix), add --with-mail=/usr/sbin/sendmail to the following command:

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

```
Creating sample config files in sample-config/ ...
*** Configuration summary for nagios 4.1.1 08-19-2015 ***:
 General Options:
 -------------------------
        Nagios executable:  nagios
        Nagios user/group:  nagios,nagios
       Command user/group:  nagios,nagcmd
             Event Broker:  yes
        Install ${prefix}:  /usr/local/nagios
    Install ${includedir}:  /usr/local/nagios/include/nagios
                Lock file:  ${prefix}/var/nagios.lock
   Check result directory:  ${prefix}/var/spool/checkresults
          Init directory:  /etc/init.d
  Apache conf.d directory:  /etc/httpd/conf.d
             Mail program:  /bin/mail
                  Host OS:  linux-gnu
          IOBroker Method:  epoll
 Web Interface Options:
 ------------------------
                 HTML URL:  http://localhost/nagios/
                  CGI URL:  http://localhost/nagios/cgi-bin/
 Traceroute (used by WAP):
Review the options above for accuracy.  If they look okay,
type 'make all' to compile the main program and CGIs.
```

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. Copy the link address for the latest version, and copy the link address so you can download it to your Nagios server.

```
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. Download the latest version to your Nagios server.

```
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


*** Configuration summary for nrpe 2.15 09-06-2013 ***:

 General Options:
 -------------------------
 NRPE port:    5666
 NRPE user:    nagios
 NRPE group:   nagios
 Nagios user:  nagios
 Nagios group: nagios
```

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       = 127.0.0.1 172.26.60.19
```

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:

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. ( currently password is “eeteamj1”

**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
```

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

```
sudo ln -s /etc/init.d/nagios /etc/rcS.d/S99nagios
```

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

**Accessing the Nagios Web Interface**

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

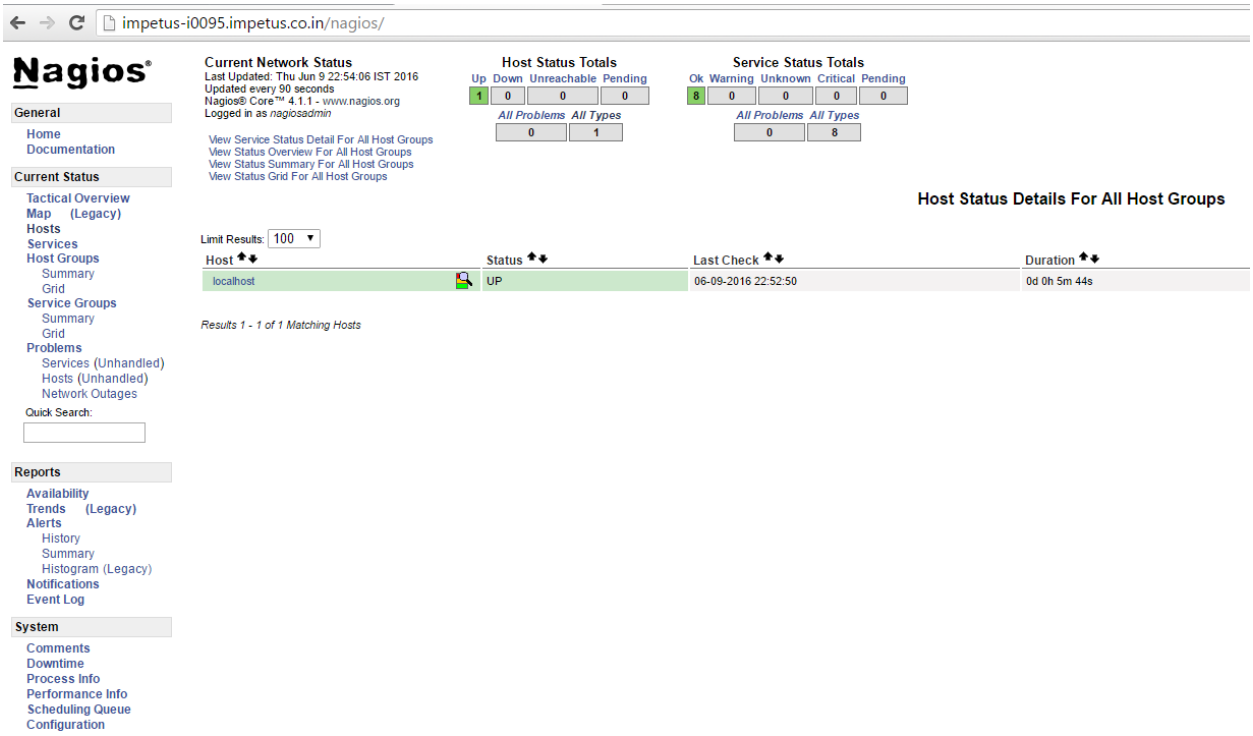
http://impetus-i0095.impetus.co.in/nagios

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
```

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=127.0.0.1,172.26.60.19
```

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/sda). 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/sda1
```

Below as per our cluster for one of our host

```
server_address=172.26.60.16
allowed_hosts=172.16.60.19

command[check_hda1]=/usr/lib/nagios/plugins/check_disk -w 20% -c 10% -p
/dev/sda1
```

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
}
```

Added the following for our cluster

**# sudo vi /usr/local/nagios/etc/servers/impetus-i0161.cfg**

```
define host {
        use                             linux-server
        host_name                       impetus-i0161.impetus.co.in
        alias                           impetus-i0161 Ambari Ganglia Server
        address                         172.26.60.16
        max_check_attempts              5
        check_period                    24x7
        notification_interval           30
        notification_period             24x7
}
```

**# sudo vi /usr/local/nagios/etc/servers/impetus-I0163.cfg**

```
define host {
        use                             linux-server
        host_name                       impetus-I0163.impetus.co.in
        alias                           impetus-I0163 NameNode
```

```
        address                         172.26.60.17
        max_check_attempts              5
        check_period                    24x7
        notification_interval           30
        notification_period             24x7
}
```

**# sudo vi /usr/local/nagios/etc/servers/impetus-i0203.cfg**

```
define host {
        use                             linux-server
        host_name                       impetus-i0203.impetus.co.in
        alias                           impetus-i0203 DataNode JournalNode NodeManager
ResourceManager
        address                         172.26.60.18
        max_check_attempts              5
        check_period                    24x7
        notification_interval           30
        notification_period             24x7
}
```

**# sudo vi /usr/local/nagios/etc/servers/impetus-i0095.cfg**

```
define host {
        use                             linux-server
        host_name                       impetus-i0095.impetus.co.in
        alias                           impetus-i0095 JournalNode Grafana Metrics
Monitor NameNode
        address                         172.26.60.19
        max_check_attempts              5
        check_period                    24x7
        notification_interval           30
        notification_period             24x7
}
```

If you want to monitor particular services, then follow the below process

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 same 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 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:

The below process is for setting up Nagios to monitor Hadoop components where ambari-metrics is already installed.

**Install the Configuration Files**

There are several configuration files that must be set up for Nagios.

**Extract the Nagios Configuration Files**

From the file you downloaded in Download Companion Files, open the configuration_files.zip and copy the files in the nagios folder to a temporary directory. The nagios folder contains two sub-folders, objects and plugins.7

```
# wget http://public-repo-
1.hortonworks.com/HDP/tools/2.0.6.0/hdp_manual_install_rpm_helper_files-
2.0.6.76.tar.gz
```

```
# tar -xvf hdp_manual_install_rpm_helper_files-2.0.6.76.tar.gz
```

**Create the Nagios Directories**

Make the following Nagios directory:

```
mkdir -p /var/run/nagios
```

Change ownership on those directories to the Nagios user:

```
chown -R nagios:nagios /var/run/nagios
```

**Copy the Configuration Files**

Copy the contents of the objects folder into place:

```
cp /home/nagios/hdp_manual_install_rpm_helper_files-
2.0.6.76/configuration_files/nagios/objects/*.*
/usr/local/nagios/etc/objects/
```

Copy the contents of the plugins folder into place:

```
cp /home/nagios/hdp_manual_install_rpm_helper_files-
2.0.6.76/configuration_files/nagios/plugins/*.* /usr/lib/nagios/plugins/
```

**Set the Nagios Admin Password** (You can skip this as we had already set it earlier)

1. Choose a Nagios administrator password, for example, “admin”.
2. Set the password. Use the following command:

```
htpasswd -c -b  /etc/nagios/htpasswd.users nagiosadmin eeteamj1
```

**Set the Nagios Admin Email Contact Address** (You can skip this as we had already set it earlier)

1. Open vi /usr/local/nagios/etc/objects/contacts.cfg with a text editor.
2. Change the nagios@localhost value to the admin email address so it can receive alerts.

**Register the Hadoop Configuration Files**

1. Open /usr/local/nagios/etc/nagios.cfg with a text editor.
2. In the section OBJECT CONFIGURATION FILE(S), add the following:

   ```
   # Definitions for hadoop servers
   cfg_file=/usr/local/nagios/etc/objects/hadoop-commands.cfg
   cfg_file=/usr/local/nagios/etc/objects/hadoop-hosts.cfg
   cfg_file=/usr/local/nagios/etc/objects/hadoop-hostgroups.cfg
   cfg_file=/usr/local/nagios/etc/objects/hadoop-services.cfg
   cfg_file=/usr/local/nagios/etc/objects/hadoop-servicegroups.cfg
   ```

3. Change the `command-file` directive to /usr/local/nagios/var/rw/nagios.cmd:

   ```
   command_file=/usr/local/nagios/var/rw/nagios.cmd
   ```

**Set Hosts**

1. Open /`usr/local/nagios/etc/objects/hadoop-hosts.cfg` with a text editor.
2. Create a "`define host { … }`" entry for each host in your cluster using the following format:

```
define host {
        alias        impetus-i0161
        host_name    impetus-i0161.impetus.co.in
        use          linux-server
        address      172.26.60.16
        check_interval         0.25
        retry_interval         0.25
        max_check_attempts     4
        notifications_enabled     1
        first_notification_delay  0  # Send notification soon after change in
the hard state
        notification_interval     0  # Send the notification once
        notification_options      d,u,r
}

define host {
        alias        impetus-I0163
        host_name    impetus-I0163.impetus.co.in
        use          linux-server
        address      172.26.60.17
        check_interval         0.25
        retry_interval         0.25
        max_check_attempts     4
        notifications_enabled     1
        first_notification_delay  0     # Send notification soon after change
in the hard state
        notification_interval     0     # Send the notification once
        notification_options      d,u,r
}

define host {
        alias        impetus-i0203
        host_name    impetus-i0203.impetus.co.in
        use          linux-server
        address      172.26.60.18
        check_interval         0.25
        retry_interval         0.25
        max_check_attempts     4
        notifications_enabled     1
        first_notification_delay  0     # Send notification soon after change
in the hard state
        notification_interval     0     # Send the notification once
        notification_options      d,u,r
}

define host {
        alias        impetus-i0095
        host_name    impetus-i0095.impetus.co.in
        use          linux-server
        address      172.26.60.19
        check_interval         0.25
        retry_interval         0.25
        max_check_attempts     4
        notifications_enabled     1
        first_notification_delay  0     # Send notification soon after change
in the hard state
```

```
        notification_interval     0     # Send the notification once
        notification_options      d,u,r
}
```

**Set Host Groups**

1. Open /usr/local/nagios/etc/objects /hadoop-hostgroups.cfg with a text editor.
2. Create host groups based on all the hosts and services you have installed in your cluster. Each host group entry should follow this format:

```
define hostgroup {
       hostgroup_name  @NAME@
       alias           @ALIAS@
       members         @MEMBERS@
   }
```

Where

**Table 15.1. Host Group Parameters**

| Parameter | Description |
|---|---|
| @NAME@ | The host group name |
| @ALIAS@ | The host group alias |
| @MEMBERS@ | A comma-separated list of hosts in the group |

3. The following table lists the core and monitoring host groups:

**Table 15.2. Core and Monitoring Hosts**

| Service | Component | Name | Alias | Members |
|---|---|---|---|---|
| All servers in the cluster | | all-servers | All Servers | List all servers in the cluster |
| HDFS | NameNode | namenode | namenode | The NameNode host |
| HDFS | SecondaryNameNode | snamenode | snamenode | The Secondary NameNode host |

| Service | Component | Name | Alias | Members |
|---|---|---|---|---|
| MapReduce | JobTracker | jobtracker | jobtracker | The Job Tracker host |
| HDFS, MapReduce | Slaves | slaves | slaves | List all hosts running DataNode and TaskTrackers |
| Nagios | | nagios-server | nagios-server | The Nagios server host |
| Ganglia | | ganglia-server | ganglia-server | The Ganglia server host |

4. The following table lists the ecosystem project host groups:

**Table 15.3. Ecosystem Hosts**

| Service | Component | Name | Alias | Members |
|---|---|---|---|---|
| HBase | Master | hbasemaster | hbasemaster | List the master server |
| HBase | Region | regions-servers | region-servers | List all region servers |
| ZooKeeper | | zookeeper-servers | zookeeper-servers | List all ZooKeeper servers |
| Oozie | | oozie-server | oozie-server | The Oozie server |
| Hive | | hiveserver | hiverserver | The Hive metastore server |
| WebHCat | | webhcat-server | webhcat-server | The WebHCat server |
| Templeton | | templeton-server | templeton-server | The Templeton server |

Following entries were made for our cluster in /usr/local/nagios/etc/objects /hadoop-hostgroups.cfg

```
define hostgroup {
        hostgroup_name  EETeamJ1
        alias           All Servers in the Cluster
        members         impetus-i0161.impetus.co.in,impetus-
I0163.impetus.co.in,impetus-i0203.impetus.co.in,impetus-i0095.impetus.co.in
}

define hostgroup {
        hostgroup_name  namenode
        alias           namenode
        members         impetus-i0095.impetus.co.in
}

define hostgroup {
        hostgroup_name  snamenode
        alias           snamenode
        members         impetus-I0163.impetus.co.in
}

define hostgroup {
        hostgroup_name  slaves
        alias           slaves
        members         impetus-i0203.impetus.co.in
}

define hostgroup {
        hostgroup_name  resourcemanager
        alias           resourcemanager
        members         impetus-i0161.impetus.co.in
}

define hostgroup {
        hostgroup_name  nodemanagers
        alias           nodemanagers
        members         impetus-i0203.impetus.co.in
}
define hostgroup {
        hostgroup_name  nagios-server
        alias           nagios-server
        members         impetus-i0095.impetus.co.in
}

define hostgroup {
        hostgroup_name  ganglia-server
        alias           ganglia-server
        members         impetus-i0161.impetus.co.in
}

define hostgroup {
        hostgroup_name  historyserver2
        alias           historyserver2
        members         impetus-I0163.impetus.co.in
}


define hostgroup {
        hostgroup_name  region-servers
        alias           region-servers
        members         dn2.localdomain,dn0.localdomain
}
```

```
define hostgroup {
        hostgroup_name  zookeeper-servers
        alias           zookeeper-servers
        members         impetus-i0161.impetus.co.in,impetus-
I0163.impetus.co.in,impetus-i0095.impetus.co.in
}


define hostgroup {
        hostgroup_name  oozie-server
        alias           oozie-server
        members         impetus-i0161.impetus.co.in
}
define hostgroup {
        hostgroup_name  hiveserver
        alias           hiveserver
        members         impetus-i0161.impetus.co.in
}

define hostgroup {
        hostgroup_name  webhcat-server
        alias           webhcat-server
        members         impetus-i0161.impetus.co.in
}
```

**Set Services**

1. Open /usr/local/nagios/etc/objects/hadoop-services.cfg with a text editor.
2. This file contains service definitions for the following services: Ganglia, HBase (Master and Region), ZooKeeper, Hive, Templeton and Oozie
3. Remove any services definitions for services you have not installed.
4. Replace the parameter @NAGIOS_BIN@ and @STATUS_DAT@ parameters based on the operating system.

```
@STATUS_DAT@ = /usr/local/nagios/var/status.dat
@NAGIOS_BIN@ = /usr/local/nagios/bin/nagios
```

5. If you have installed Hive or Oozie services, replace the parameter @JAVA_HOME@ with the path to the Java home. For example, /usr/java/default.

**Set Status**

1. Open /usr/local/nagios/etc/objects /hadoop-commands.cfg with a text editor.
2. Replace the @STATUS_DAT@ parameter with the location of the Nagios status file. The file is located: /usr/local/nagios/var/status.dat

**Validate the Installation**

Use these steps to validate your installation.

**3.1. Validate the Nagios Installation. Rectify issue if you see any.**

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