

Adding a Host in Nagios

Commands with green font represents master, white font terminal represents slave.
Installing NRPE on slave:

Step 1: On slave run the following command.

```
ubuntu@ip-172-31-35-225: ~
ubuntu@ip-172-31-35-225:~$ sudo apt-get update
Hit:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu bionic InRelease
Get:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Get:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 kB]
Get:4 http://us-east-2.ec2.archive.ubuntu.com/ubuntu bionic/universe Sources [9051 kB]
Get:5 http://security.ubuntu.com/ubuntu bionic-security InRelease [83.2 kB]
Get:6 http://us-east-2.ec2.archive.ubuntu.com/ubuntu bionic/main Sources [829 kB]
Get:7 http://us-east-2.ec2.archive.ubuntu.com/ubuntu bionic/multiverse Sources [181 kB]
Get:8 http://us-east-2.ec2.archive.ubuntu.com/ubuntu bionic/restricted Sources [5324 B]
```

Step 2: Install the required plugins

```
ubuntu@ip-172-31-35-225: ~
ubuntu@ip-172-31-35-225:~$ sudo apt-get install nagios-nrpe-server nagios-plugins
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libarchive13 libavahi-client3 libavahi-common-data libavahi-common3 libcups2 libdb11 libjansson4 libldb1
  libmysqlclient20 libnet-snmp-perl libpq5 libpython-stdlib libpython2.7 libpython2.7-minimal libpython2.7-
  librsensors4 libsmbclient libsnmp-base libsnmp30 libtalloc2 libtdb1 libtevent0 libtirpc1 libwbclient0
  monitoring-plugins monitoring-plugins-basic monitoring-plugins-common monitoring-plugins-standard mysql-c
  python-crypto python-ldb python-minimal python-samba python-talloc python-tdb python2.7 python2.7-minimal
  samba-common samba-common-bin samba-libs smbclient snmp
```

Step 3: Open the configuration file as shown below:

```
ubuntu@ip-172-31-35-225: ~
ubuntu@ip-172-31-35-225:~$ sudo nano /etc/nagios/nrpe.cfg
ubuntu@ip-172-31-35-225:~$
```

Find the below given line and in place of XXX add your **master IP**
allowed_hosts=127.0.0.1, XXX

```
ubuntu@ip-172-31-35-225: ~  
GNU nano 2.9.3 /etc/nagios/nrpe.cfg  
  
# NRPE GROUP  
# This determines the effective group that the NRPE daemon should run as.  
# You can either supply a group name or a GID.  
#  
# NOTE: This option is ignored if NRPE is running under either inetd or xinetd  
nrpe_group=nagios  
  
# ALLOWED HOST ADDRESSES  
# This is an optional comma-delimited list of IP address or hostnames  
# that are allowed to talk to the NRPE daemon. Network addresses with a bit mask  
# (i.e. 192.168.1.0/24) are also supported. Hostname wildcards are not currently  
# supported.  
#  
# Note: The daemon only does rudimentary checking of the client's IP  
# address. I would highly recommend adding entries in your /etc/hosts.allow  
# file to allow only the specified host to connect to the port  
# you are running this daemon on.  
#  
# NOTE: This option is ignored if NRPE is running under either inetd or xinetd  
allowed_hosts=127.0.0.1,18.225.34.109
```

Step 4: Start NRPE service as shown below:

```
ubuntu@ip-172-31-35-225: ~  
ubuntu@ip-172-31-35-225:~$ sudo /etc/init.d/nagios-nrpe-server restart  
[ ok ] Restarting nagios-nrpe-server (via systemctl): nagios-nrpe-server.service.  
ubuntu@ip-172-31-35-225:~$
```

Install check_nrpe on Master

Step 1: Now we need to install check_nrpe on Master.

```
ubuntu@ip-172-31-32-221: ~  
ubuntu@ip-172-31-32-221:~$ sudo apt-get update  
Hit:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu bionic InRelease  
Get:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]  
Get:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 kB]  
Get:4 http://us-east-2.ec2.archive.ubuntu.com/ubuntu bionic-updates/main amd64 Packages [461 kB]  
Get:5 http://security.ubuntu.com/ubuntu bionic-security InRelease [83.2 kB]  
Get:6 http://us-east-2.ec2.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 Packages [704 kB]  
Fetched 1412 kB in 1s (2502 kB/s)  
Reading package lists... Done  
ubuntu@ip-172-31-32-221:~$
```

Step 2: Install the following package.

```
ubuntu@ip-172-31-32-221: ~  
ubuntu@ip-172-31-32-221:~$ sudo apt-get install -y autoconf automake gcc libc6 libmccrypt-dev  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
libc6 is already the newest version (2.27-3ubuntu1).  
make is already the newest version (4.1-9.1ubuntu1).  
make set to manually installed.  
gcc is already the newest version (4:7.3.0-3ubuntu2.1).
```

Step 3: Get inside tmp.

```
ubuntu@ip-172-31-32-221: /tmp  
ubuntu@ip-172-31-32-221:~$ cd /tmp  
ubuntu@ip-172-31-32-221:/tmp$
```

Step 4: Download the source.

```
ubuntu@ip-172-31-32-221: /tmp  
ubuntu@ip-172-31-32-221:/tmp$ wget --no-check-certificate -O nrpe.tar.gz https://github.com/NagiosEnterprises/  
/nrpe-3.2.1.tar.gz  
--2018-12-19 14:54:06-- https://github.com/NagiosEnterprises/nrpe/archive/nrpe-3.2.1.tar.gz  
Resolving github.com (github.com)... 192.30.253.112, 192.30.253.113  
Connecting to github.com (github.com)|192.30.253.112|:443... connected.  
HTTP request sent, awaiting response... 302 Found  
Location: https://codeload.github.com/NagiosEnterprises/nrpe/tar.gz/nrpe-3.2.1 [following]  
--2018-12-19 14:54:06-- https://codeload.github.com/NagiosEnterprises/nrpe/tar.gz/nrpe-3.2.1  
Resolving codeload.github.com (codeload.github.com)... 192.30.253.120, 192.30.253.121  
Connecting to codeload.github.com (codeload.github.com)|192.30.253.120|:443... connected.  
HTTP request sent, awaiting response... 200 OK  
Length: unspecified [application/x-gzip]  
Saving to: 'nrpe.tar.gz'
```

Step 5: Untar the file

Step 6: Then execute the following command.

```
ubuntu@ip-172-31-32-221: /tmp/nrpe-nrpe-3.2.1
```

```
ubuntu@ip-172-31-32-221:/tmp/nrpe-nrpe-3.2.1$ ./configure
```

It takes some time to configure.

```
*** Configuration summary for nrpe 3.2.1 2017-09-01 ***:
```

```
General Options:
```

```
-----
```

```
NRPE port:      5666
NRPE user:      nagios
NRPE group:     nagios
Nagios user:    nagios
Nagios group:   nagios
```

Step 7: Run the check_nrpe command.

```
ubuntu@ip-172-31-32-221: /tmp/nrpe-nrpe-3.2.1
```

```
ubuntu@ip-172-31-32-221:/tmp/nrpe-nrpe-3.2.1$ make check_nrpe
cd ./src/; make
make[1]: Entering directory '/tmp/nrpe-nrpe-3.2.1/src'
gcc -g -O2 -DHAVE_CONFIG_H -I ../include -I ../../include -o nrpe ./nrpe.c ./utils.c
./nrpe.c: In function 'read_config_dir':
./nrpe.c:1064:54: warning: '%s' directive output may be truncated writing up to 255
mat-truncation=]
    snprintf(config_file, sizeof(config_file) - 1, "%s/%s", dirname, dirfile->d_name)
                                     ^~
In file included from /usr/include/stdio.h:862:0,
                 from ../include/config.h:31,
                 from ./nrpe.c:37:
/usr/include/x86_64-linux-gnu/bits/stdio2.h:64:10: note: '__builtin__snprintf_chk'
into a destination of size 255
    return __builtin__snprintf_chk (__s, __n, __USE_FORTIFY_LEVEL - 1,
    ^~~~~~
```

Step 8: Now install plugins.

ubuntu@ip-172-31-32-221: /tmp/nrpe-nrpe-3.2.1

```
ubuntu@ip-172-31-32-221:/tmp/nrpe-nrpe-3.2.1$ sudo make install-plugin
cd ./src/; make install-plugin
make[1]: Entering directory '/tmp/nrpe-nrpe-3.2.1/src'
/usr/bin/install -c -m 755 -d /usr/local/nagios/bin
/usr/bin/install -c -m 755 ../uninstall /usr/local/nagios/bin/nrpe-uninstall
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/libexec
/usr/bin/install -c -m 775 -o nagios -g nagios -d /usr/local/nagios/libexec
/usr/bin/install -c -m 775 -o nagios -g nagios check_nrpe /usr/local/nagios/libexec
make[1]: Leaving directory '/tmp/nrpe-nrpe-3.2.1/src'
ubuntu@ip-172-31-32-221:/tmp/nrpe-nrpe-3.2.1$
```

Step 9: We will check the connection as shown below:

ubuntu@ip-172-31-32-221: ~

```
ubuntu@ip-172-31-32-221:~$ /usr/local/nagios/libexec/check_nrpe -H 18.220.227.9
NRPE v3.2.1
ubuntu@ip-172-31-32-221:~$
```

Now we need to make the following configuration file before we observe the connection.

Step 10: Make a configuration file as shown:

ubuntu@ip-172-31-32-221: ~

```
ubuntu@ip-172-31-32-221:~$ sudo nano /usr/local/nagios/etc/servers/Host1.cfg
```

Add the below given content there.

After adding the content, the configuration file should look like this:

ubuntu@ip-172-31-27-173: ~

```
GNU nano 2.9.3 /usr/local/nagios/etc/servers/MyHost01.cfg
#####
# Linux Host 001 configuration file
#####

define host {
    use                Linux-server
    host_name          Linux_Host_01
    alias              Linux_Host_01
    address            18.224.140.24
    register           1
}
define service{
    host_name          Linux_Host_01
    service_description PING
    check_command      check_ping!100.0,20%!500.0,60%
    max_check_attempts 2
    check_interval     2
    retry_interval     2
    check_period       24x7
    check_freshness    1
    contact_groups     admins
    notification_interval 2
    notification_period 24x7
    notifications_enabled 1
    register           1
}
```

Step 11: Now verify the configuration again.

 ubuntu@ip-172-31-32-221: ~

```
ubuntu@ip-172-31-32-221:~$ sudo /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg


Nagios Core 4.4.2
Copyright (c) 2009-present Nagios Core Development Team and Community Contributors
Copyright (c) 1999-2009 Ethan Galstad
Last Modified: 2018-08-16
License: GPL
```

```
Checking objects...
    Checked 9 services.
    Checked 2 hosts.
    Checked 1 host groups.
    Checked 0 service groups.
    Checked 1 contacts.
    Checked 1 contact groups.
    Checked 24 commands.
    Checked 5 time periods.
    Checked 0 host escalations.
    Checked 0 service escalations.
Checking for circular paths...
    Checked 2 hosts
    Checked 0 service dependencies
    Checked 0 host dependencies
    Checked 5 timeperiods
Checking global event handlers...
Checking obsessive compulsive processor commands...
Checking misc settings...

Total Warnings: 0
Total Errors: 0
```

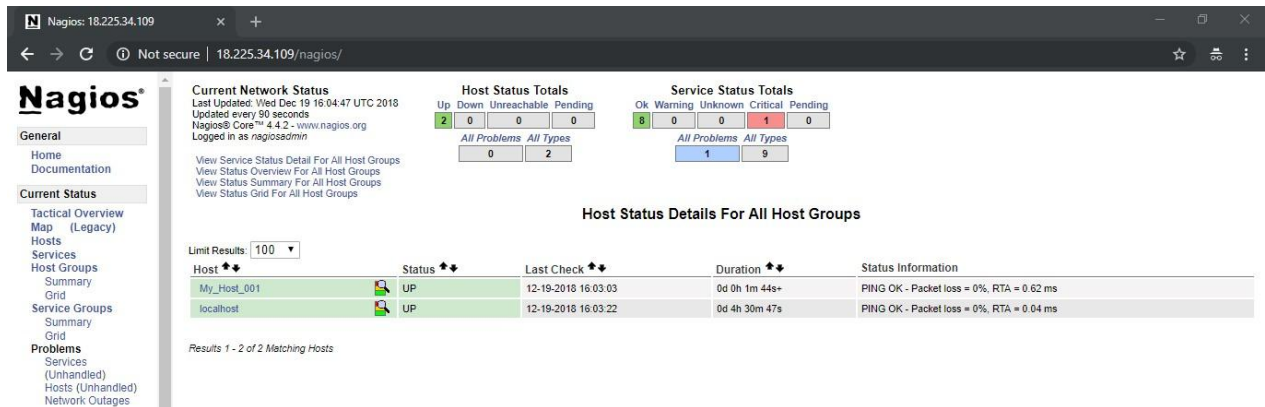
Everything looks fine!

Step 12: Start Nagios.

 ubuntu@ip-172-31-32-221: ~

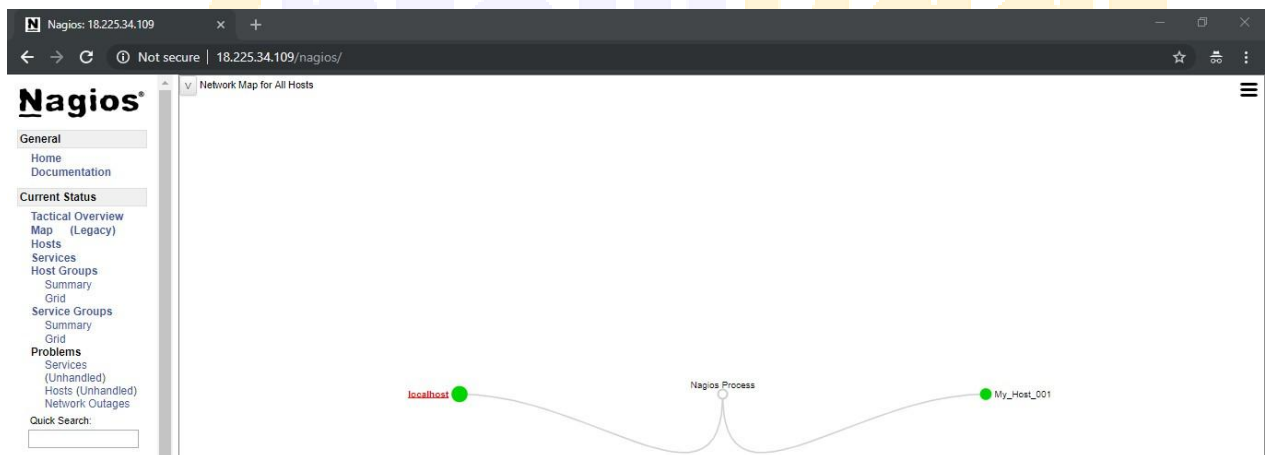
```
ubuntu@ip-172-31-32-221:~$ sudo service nagios restart
ubuntu@ip-172-31-32-221:~$
```

Step 13: Go to the browser. And you can see the slave is added to the connection with a PING service running.



The screenshot shows the Nagios web interface at 18.225.34.109. The interface includes a sidebar with navigation links like General, Current Status, Tactical Overview, Map, Hosts, Services, Host Groups, Service Groups, Problems, and Network Outages. The main content area displays 'Current Network Status' (Last Updated: Wed Dec 19 16:04:47 UTC 2018), 'Host Status Totals' (Up: 2, Down: 0, Unreachable: 0, Pending: 0), and 'Service Status Totals' (Ok: 8, Warning: 0, Unknown: 0, Critical: 1, Pending: 0). Below this is a table titled 'Host Status Details For All Host Groups' showing two hosts: 'My_Host_001' and 'localhost', both with a status of 'UP' and a last check time of 12-19-2018 16:03:03. The status information for both hosts is 'PING OK - Packet loss = 0%, RTA = 0.62 ms' and 'PING OK - Packet loss = 0%, RTA = 0.04 ms' respectively.

To see the connection map, click on **Map**.



The screenshot shows the Nagios web interface at 18.225.34.109, specifically the 'Network Map for All Hosts' view. The map displays a central node labeled 'Nagios Process' connected to two other nodes: 'localhost' on the left and 'My_Host_001' on the right. The nodes are represented by green circles, and the connections are shown as lines.

Congratulations! You have successfully setup Nagios Master-Slave Cluster.