



Alibaba Cloud | TIANCHI

Win \$210000+USD cash prize!

Join FashionAI Global Algorithm Challenge Now



How to Add Linux Host to Nagios Monitoring Using NRPE Plugin

Add Linux Host to Nagios Monitoring

In the last article, we discussed about installing Nagios server on [CentOS 7](#), [Ubuntu 16.04](#), and [Debian 9](#). Here, in this post, we will add Linux host to [Nagios Monitoring](#) tool using NRPE plugin.

I'm assuming that you have a working Nagios setup, if not, use any one of the below links and setup Nagios server.

READ: [How to Install and Configure Nagios 4.3.2 on CentOS 7 / RHEL 7](#)

READ: [Install and Configure Nagios on Ubuntu 16.04](#)

READ: [How to Install and Configure Nagios 4.3.2 on Debian 9 Stretch](#)

NRPE Plugin:

Nagios Remote Plugin Executor (abbreviated as NRPE) plugin allows you to monitor applications and services running on remote Linux / Windows hosts. This NRPE Add-on helps Nagios to monitor local resources like CPU, Memory, Disk, Swap, etc. of the remote host.

Finally, your remote system must have this NRPE plugin installed.

Steps to add Linux Host to Nagios monitoring,

On Nagios Remote Host:

1. Install NRPE Add-on & Nagios plugins
2. Configure NRPE Add-On
3. Configure Nagios Checks

On Nagios Server Host:

1. Configure Nagios Server
2. Monitor the remote machine.

Nagios Remote Host:

Install NRPE Add-on & Nagios Plugins:

NRPE and Nagios plugins are not available in the base repository. So, for CentOS 7 / RHEL 7, you should have EPEL repository configured on your machine.

READ: [How to configure EPEL repository on CentOS 7/ RHEL 7](#)

```
rpm -ivh https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
```

Use the following command to install NRPE Add-on and Nagios plugins.

```
## CentOS 7 / RHEL 7 ##
yum install -y nrpe nagios-plugins-all

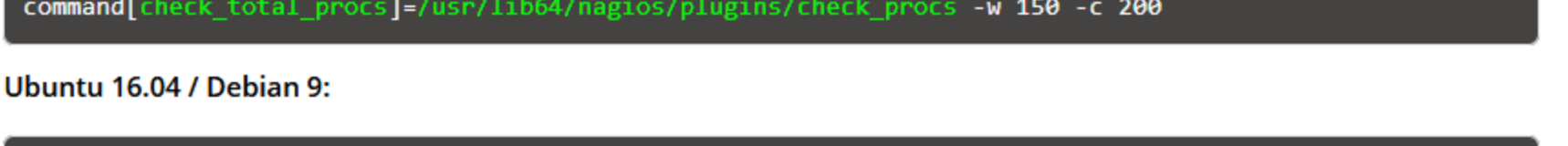
## Ubuntu 16.04 / Debian 9 ##
apt-get install -y nagios-nrpe-server nagios-plugins
```

Configure NRPE Add-on:

Modify the NRPE configuration file to accept the connection from the Nagios server, Edit the `/etc/nagios/nrpe.cfg` file.

```
vi /etc/nagios/nrpe.cfg
```

Add the Nagios servers IP address, separated by comma like below.



```
allowed_hosts=127.0.0.1,192.168.12.10
```

Configure Nagios Checks:

The `/etc/nagios/nrpe.cfg` file contains the basic commands to check the attributes (CPU, Memory, Disk, etc.architecture) and services (HTTP, FTP, etc.) on remote hosts. Below command lines lets you monitor attributes with the help of Nagios plugins.

Note: The path to the Nagios plugins may change depends on your operating system architecture (i386 or x86_64)

CentOS 7 / RHEL 7:

```
command[check_users]=usr/lib64/nagios/plugins/check_users -w 5 -c 10
command[check_load]=usr/lib64/nagios/plugins/check_load -w 15,10,5 -c 30,25,20
command[check_root]=usr/lib64/nagios/plugins/check_disk -w 20% -c 10% -p /dev/mapper/centos-root
command[check_swap]=usr/lib64/nagios/plugins/check_swap -w 20% -c 10%
command[check_total_procs]=usr/lib64/nagios/plugins/check_procs -w 150 -c 200
```

Ubuntu 16.04 / Debian 9:

```
command[check_users]=usr/lib/nagios/plugins/check_users -w 5 -c 10
command[check_load]=usr/lib/nagios/plugins/check_load -w 15,10,5 -c 30,25,20
command[check_root]=usr/lib/nagios/plugins/check_disk -w 20% -c 10% -p /dev/mapper/server--vg-root
command[check_swap]=usr/lib/nagios/plugins/check_swap -w 20% -c 10%
command[check_total_procs]=usr/lib/nagios/plugins/check_procs -w 150 -c 200
```

In the above command definition `-w` stands for **warning** and `-c` stands for **critical**.

For example, execute the below command in another terminal. Tested on **Ubuntu 16.04**:

```
/usr/lib/nagios/plugins/check_procs -w 150 -c 200
```

Output:

```
PROCS WARNING: 190 processes | procs=190;150;200;0;
```

Nagios plugin will count running processes and will warn you when the process count is more than 150, or it will report you critical when the process count is more than 200, and at the same time, the output will state OK if the count is below 150.

You can adjust the alert level as per your requirements. Change warning to 200 and critical to 250 for testing purpose. Now you can see an OK message.

```
/usr/lib/nagios/plugins/check_procs -w 200 -c 250
```

Output:

```
PROCS OK: 189 processes | procs=189;200;250;0;
```

These command definitions have to be entered on template file on the Nagios server host to enable the monitoring.

Restart the NRPE service.

```
## CentOS 7 / RHEL 7 ##
systemctl start nrpe
systemctl enable nrpe

## Ubuntu 16.04 / Debian 9 ##
/etc/init.d/nagios-nrpe-server restart
```

Nagios Server Host:

NRPE and Nagios plugins are not available in the base repository. So, for CentOS 7 / RHEL 7, you should have EPEL repository configured on your machine.

READ: [How to configure EPEL repository on CentOS 7/ RHEL 7](#)

```
rpm -ivh https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
```

Install `check_nrpe` plugin:

Use the following command to install "`check_nrpe`" plugin on your machine.

```
## CentOS 7 / RHEL 7 ##
yum -y install nagios-plugins-nrpe

## Ubuntu 16.04 / Debian 9 ##
apt-get -y install nagios-nrpe-plugin
```

Edit the Nagios configuration file to include all `*.cfg` files inside the `/usr/local/nagios/etc/servers` directory.

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

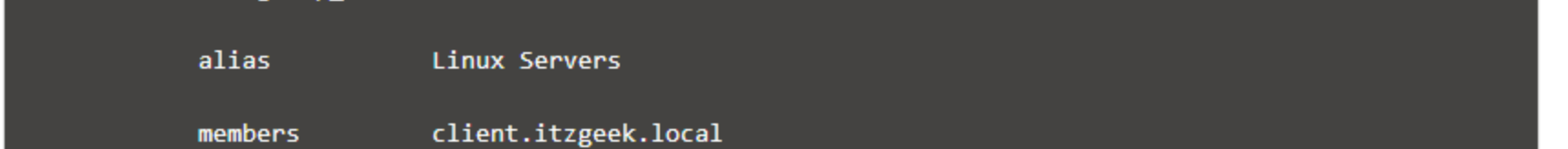
Add or uncomment the following line.

```
cfg_dir=/usr/local/nagios/etc/servers
```

Create a configuration directory.

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

Configure Nagios Server:



Now it's time to configure the Nagios server to monitor the remote client machine, and You'll need to create a command definition in Nagios object configuration file to use the `"check_nrpe"` plugin. Open the `"commands.cfg"` file....

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

Add the following Nagios command definition to the file.

```
## CentOS 7 / RHEL 7 ##

# .check_nrpe. command definition
define command{
    command_name check_nrpe
    command_line /usr/lib64/nagios/plugins/check_nrpe -H $HOSTADDRESS$ -t 30 -c $ARG1$
}

## Ubuntu 16.04 Debian 9 ##

# .check_nrpe. command definition
define command{
    command_name check_nrpe
    command_line /usr/lib/nagios/plugins/check_nrpe -H $HOSTADDRESS$ -t 30 -c $ARG1$
}
```

Add a Linux host to Nagios server:

Create a client configuration file (`/usr/local/nagios/etc/servers/client.itzgeek.local.cfg`) to define the host and service definitions of remote Linux host. You can also use the following template and modify it according to your requirement. The following template is for monitoring **"logged in users, system load, disk usage (/ - partitions), swap, and total processes**.

```
vi /usr/local/nagios/etc/servers/client.itzgeek.local.cfg
```

Copy the below content to the above file.

```
define host{
    use linux-server
    host_name client.itzgeek.local
    alias client.itzgeek.local
    address 192.168.12.20
}

define hostgroup{
    hostgroup_name linux-server
    alias Linux Servers
    members client.itzgeek.local
}

define service{
    use local-service
    host_name client.itzgeek.local
    service_description SWAP Usage
    check_command check_nrpe!check_swap
}

define service{
    use local-service
    host_name client.itzgeek.local
    service_description Root / Partition
    check_command check_nrpe!check_root
}

define service{
    use local-service
    host_name client.itzgeek.local
    service_description Current Users
    check_command check_nrpe!check_users
}

define service{
    use local-service
    host_name client.itzgeek.local
    service_description Total Processes
    check_command check_nrpe!check_total_procs
}

define service{
    use local-service
    host_name client.itzgeek.local
    service_description Current Load
    check_command check_nrpe!check_load
}
```

Verify Nagios for any errors.

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

Restart the Nagios server.

```
service nagios restart
```

Firewall:

Configure the firewall so that the Nagios server can able to reach NRPE server running on remote Linux host. Run these commands on remote Linux machine.

FirewallD:

```
firewall-cmd --permanent --add-port=5666/tcp
firewall-cmd --reload
```

UFW:

```
ufw allow 5666/tcp
ufw reload
ufw enable
```

Monitor the remote machine:

Go and check the Nagios web interface to view the new services we added just now.

The screenshot of the remote Linux box with the service we configured few steps before:

