



## Purpose

This document describes how to automatically restart problematic services on Linux servers using NRPE.

## Target Audience

This document is intended for use by Nagios XI Administrators who want to automate starting, stopping, and restarting of problematic services on their Linux servers. A basic knowledge of NRPE is recommended.

## Prerequisites

You should already have the NRPE server configured on the Linux machine you would like to restart services on, please refer to the following documentation:

[Installing The Nagios XI Linux Agent](#)

## Background Information

In this guide you will be shown how to use an event handler to restart a service on a Linux server. This guide will be using a CentOS 6.x Linux server, which uses the command `service` to perform the service restart.

## Configuring NRPE Command

First we'll create a command in the `/usr/local/nagios/etc/nrpe.cfg` file that will perform the restart command. Establish a terminal or SSH session to your Linux server as the root user and execute the following command:

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

When using the vi editor, to make changes press `i` on the keyboard first to enter insert mode. Press `Esc` to exit insert mode.

Go to the end of the file by pressing Shift + G and add the following line:

```
command[service_restart]=sudo service $ARG1$ restart 2>&1
```

```
command[service_restart]=sudo service $ARG1$ restart
```

When you have finished, save the changes in vi by typing:

```
:wq
```

and press Enter.

Remain logged into your Linux server as you'll need to perform more steps.

## Granting NRPE Permission to Restart Services

The `nagios` user will also need to be granted permissions to execute the service command. Execute the following command as root to give NRPE permission to restart services:

```
echo "nagios ALL = NOPASSWD: `which service`" >> /etc/sudoers
```

It's very important to use the **back-tick** key on your keyboard around the ``which service`` words above, this key is commonly located to the left of the 1 key.

## Testing the Commands from Nagios XI Server

Now we will test from the Nagios XI server that the command you just added to the NRPE client on the Linux server is working. This example is going to restart the `crond` service as it is unlikely to cause any issues.

Establish a terminal session to your Nagios XI server and execute the following command:

```
/usr/local/nagios/libexec/check_nrpe -H 10.25.13.34 -p 5666 -c service_restart -a crond
```

```
[root@xi-r5x-x86 libexec]# /usr/local/nagios/libexec/check_nrpe -H 10.25.13.34 -p 5666 -c service_restart -a crond
Stopping crond: [ OK ]
Starting crond: [ OK ]
```

You can see from the screenshot on the previous page that we received back the results from the `service_restart` command, it appears to be working.

## Create Event Handler Script

Next we need to create a script that will be used by Nagios XI for the event handler. The script will be called `service_restart.sh` and will be located in the `/usr/local/nagios/libexec/` directory on the Nagios XI server. Execute the following command:

```
vi /usr/local/nagios/libexec/service_restart.sh
```

Paste the following into the terminal session:

```
#!/bin/sh
case "$1" in
    OK)
        ;;
    WARNING)
        ;;
    UNKNOWN)
        ;;
    CRITICAL)
        /usr/local/nagios/libexec/check_nrpe -H "$2" -p 5666 -c service_restart -a "$3"
        ;;
esac
exit 0
```

When you have finished, save the changes in vi by typing:

```
:wq
```

and press Enter.

Now execute the following commands to set the correction permissions:

```
chown apache:nagios /usr/local/nagios/libexec/service_restart.sh
chmod 775 /usr/local/nagios/libexec/service_restart.sh
```

You can now test the script works by executing the following command:

```
/usr/local/nagios/libexec/service_restart.sh CRITICAL 10.25.13.34 crond
```

When the script is run, it receives three arguments which are referenced as **\$1**, **\$2**, **\$3** in the script.

**\$1** = The state of the service.

**\$2** = The host address of the Linux server.

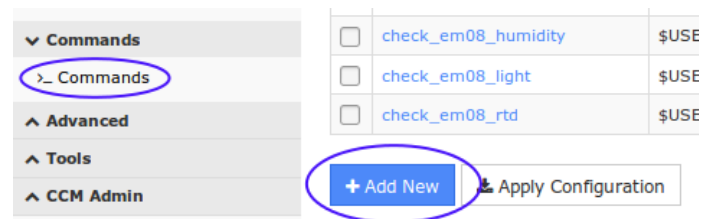
**\$3** = The name of the service being restarted.

You can see from the script above that it's only when the service is in a **CRITICAL** state that the `service_restart` command will be executed.

## Create Event Handler

Now an event handler on the Nagios XI server will be created which will be used by your services.

Navigate to **Configure > Core Config Manager**.



Select **Commands** from the list on the left, click the **>\_ Commands** link and then click the **Add New** button.

You will need to populate the fields with the values on the following page:

Command

```
Service Restart - Linux
```

Command line

```
$USER1$/service_restart.sh $SERVICESTATE$ $HOSTADDRESS$ $_SERVICE$
```

Command type

```
misc command
```

Check the **Active** check box.

Click the **Save** button and then **Apply Configuration**.

## Command Management

Command Name \*

Example: check\_example

Command Line \*

Example: \$USER1\$/check\_example -H \$HOSTADDRESS\$ -P \$ARG1\$ \$ARG2\$

Command Type:

Active ?

Available Plugins

 ?



## Adding a Service Check

Now we will need to create a Service using the Linux Server Monitoring Wizard. This guide will not go into the entire steps required, please refer to the steps in the following documentation:

[Monitoring Hosts Using NRPE](#)

On **Step 2** of the wizard you need to select the **crond** service from the list of **Services**.

Finish the wizard to create the new service.

### Services

Specify any services normally started by the init process that should be monitored to ensure they're in a running state.

init.d Service	Display Name
<input type="checkbox"/> sshd	SSH Server
<input checked="" type="checkbox"/> crond	Cron Scheduling Daemon
<input type="checkbox"/> syslog	System Logging Daemon

## Update Service With Event Handler

Now that the Nagios service is created we need to do two things:

- Select Event Handler
- Add the name of the service we want to restart as a custom variable to the service object. This is how the event handler knows what the name of the service is to restart.

Navigate to **Configure > Core Config Manager > Monitoring > Services**.

Click the service **Cron Scheduling Daemon** to edit the service.

Click the **Check Settings** tab.

From the **Event handler** drop down list select the option **Service Restart - Linux**.

For **Event handler enabled** click **On**.

### Service Management

Common Settings	✓ Check Settings	Alert Settings	Misc Settings
<b>Initial state</b> <input type="radio"/> Warning <input type="radio"/> Critical <input type="radio"/> Ok <input type="radio"/> Unreachable		<b>Obsess over service</b> <input type="radio"/> On <input type="radio"/> Off <input type="radio"/> Skip <input type="radio"/> Null	
<b>Check interval</b> <input type="text" value="5"/> min		<b>Event handler</b> <input type="text" value="Service Restart - Linux"/>	
<b>Retry interval</b> <input type="text" value="1"/> min		<b>Event handler enabled</b> <input checked="" type="radio"/> On <input type="radio"/> Off <input type="radio"/> Skip <input type="radio"/> Null	

Click the **Misc Settings** tab and then click the **Manage Free Variables** button.

### Service Management

Common Settings	✓ Check Settings	Alert Settings	Misc Settings
<b>Notes</b> <input type="text"/>		<b>2D coords</b> <input type="text" value="x,y"/>	
<b>VRML image</b> <input type="text"/>		<b>3D coords</b> <input type="text" value="x,y,z"/>	
<b>Notes URL</b> <input type="text"/>		<input type="button" value="Manage Free Variables"/>	

We will be adding a custom variable so that the event handler knows the name of the service to restart.

Name:

`_SERVICE`

Value:

`crond`

Click **Insert** and the variable will be added to the list on the right.

Name	Value	
_xiwizard	linux-server	X
_SERVICE	crond	X

Click the **Close** button and then click the **Save** button.

Click **Apply Configuration** for the changes to take affect.

In the event handler command you created, you can see the macro `$_SERVICE$SERVICE$` was used. This is how a service macro is referenced by the Nagios Core engine. More information on custom variables can be found here:

<https://assets.nagios.com/downloads/nagioscore/docs/nagioscore/4/en/customobjectvars.html>

## Test

To test simply force the service to stop on the Linux machine. Execute the following command on your Linux machine:

```
service crond stop
```

Wait for the Nagios service to go to a critical state or force the next check. Once the Nagios XI **Cron Scheduling Daemon** service is in a critical state the event handler will be executed and the Linux `crond` service will be restarted. The next time Nagios XI checks the **Cron Scheduling Daemon** service it will return to an **OK** state as the Linux `crond` service will now be running.

## Troubleshooting

If the event handler does not appear to be working as expected, check the `/usr/local/nagios/var/nagios.log` file for any errors, for example:

```
[1481763272] SERVICE ALERT: 10.25.13.34;Cron Scheduling Daemon;CRITICAL;SOFT;1;crond is stopped
[1481763272] wproc: SERVICE EVENTHANDLER job 7 from worker Core Worker 12627 is a non-check
helper but exited with return code 13
[1481763272] wproc:   early_timeout=0; exited_ok=1; wait_status=3328; error_code=0;
[1481763272] wproc:   stderr line 01: execvp(/usr/local/nagios/libexec/service_restart.sh, ...)
failed. Errno is 13: Permission denied
```

In the log entries above you can see that the worker reported that it did not have permission to execute the `service_restart.sh` command.

If the command returns “NRPE: Unable to read output” when run manually, make sure that NRPE command definition redirects `stderr` to `stdout`; i.e., change

```
command[service_restart]=sudo service $ARG1$ restart
```

to

```
command[service_restart]=sudo service $ARG1$ restart 2>&1
```



## Finishing Up

This completes the documentation on how to restart Linux services with NRPE and Nagios XI.

If you have additional questions or other support related questions, please visit us at our Nagios Support Forums:

<https://support.nagios.com/forum>

The Nagios Support Knowledgebase is also a great support resource:

<https://support.nagios.com/kb>