



Purpose

The purpose of this document is to provide a guide to integrating Mod-Gearman with Nagios XI to introduce distributed monitoring in your environment. Mod-Gearman and its worker processes intercept checks made by Nagios XI and run them externally, process them, and send them back to the Nagios XI server. In this way Mod-Gearman has been shown to greatly reduce the load taken on by the Nagios XI server while still keeping its speed and efficiency. Mod-Gearman integrates with Nagios XI as a Nagios Event Broker (NEB) module. A Worker is the component that executes the actual check, it can be located on the Nagios XI server (by default) or can run on multiple external servers.

Target Audience

This guide is directed towards Nagios XI administrators interested in integrating Mod-Gearman with their Nagios XI installation. This guide has been tested on RHEL / CentOS versions 6 / 7.

Documentation Overview

This document will provide instructions to:

- Install or Upgrade Mod-Gearman Server components on your Nagios XI server
- Create firewall rules for external workers (optional)
- Install or Upgrade Mod-Gearman Worker Components on external workers (optional)

Editing Files

In many steps of this documentation you will be required to edit files. This documentation will use the vi text editor. When using the vi editor:

- To make changes press **i** on the keyboard first to enter insert mode
- Press **Esc** to exit insert mode
- When you have finished, save the changes in vi by typing **:wq** and press Enter

Nagios XI Version Notes

If you are installing Mod Gearman 3 on Nagios XI, please make sure that you have Nagios Core version \geq 4.4.2. This means that if you are running Mod Gearman 2, you will need to remove your broker module and upgrade Nagios Core. Please refer to the following KB article for detailed instructions:

[Nagios XI - Upgrading Nagios Core](#)

Mod Gearman is supported on the following distributions:

- CentOS/RHEL 6+
- Debian 8+
- Ubuntu 16.04/18.04

For each distribution, both 32-bit and 64-bit configurations are supported where available.

Installing the Nagios Repository

In order to access the packages for Mod Gearman 3, you will need to install the official Nagios repository. If you don't have this installed, please visit the [Repository Home Page](#) to view instructions for installing the repository on your distribution.

Server Installation – New

These steps are performed on your Nagios XI server. Establish a terminal session to your Nagios XI server and execute the following commands:

```
cd /tmp
wget https://assets.nagios.com/downloads/nagiosxi/scripts/ModGearmanInstall.sh
chmod +x ModGearmanInstall.sh
./ModGearmanInstall.sh --type=server
```

Server Installation – Upgrade (3 => 3)

If you are already running a previous version of Mod-Gearman 3 on your Nagios XI server, these steps will upgrade it to the latest version and maintain your existing configurations. Establish a terminal session to your Nagios XI server and execute the following commands:

```
cd /tmp
cp /etc/mod_gearman2/* /tmp/
rm -f ModGearmanInstall.sh
wget https://assets.nagios.com/downloads/nagiosxi/scripts/ModGearmanInstall.sh
chmod +x ModGearmanInstall.sh
./ModGearmanInstall.sh --type=server --upgrade
```

Server Installation – Upgrade (2 => 3)

If you are already running a previous version of Mod-Gearman 2 on your Nagios XI server, you will need to uninstall Mod Gearman 2, upgrade Nagios Core, and re-install Mod Gearman 3. It is recommended that you also upgrade to the latest version of XI if possible. Note that you will want to back up any module/worker configuration files you have. These won't be compatible with Mod Gearman 3, but will give you an indication of anything you need to change in the new configuration files. Establish a terminal session to your Nagios XI server and execute the following commands:

```
# Remove Mod Gearman 2
cp /etc/mod_gearman2/* /tmp/
yum remove gearmand gearmand-server gearmand-debuginfo gearmand-devel mod_gearman2 -y
sed -i 's/^broker\(.*\)gearman2\(.*\)#broker\1gearman2\2/' /usr/local/nagios/etc/nagios.cfg

# Upgrade Nagios Core
cd /tmp
rm -rf nagiosxi xi* ModGearmanInstall.sh
wget https://assets.nagios.com/downloads/nagiosxi/xi-latest.tar.gz
```

```
tar xzf xi-latest.tar.gz
cd nagiosxi
./init.sh
cd subcomponents/nagioscore
./upgrade

# Download and install Mod Gearman 3
cd /tmp
wget https://assets.nagios.com/downloads/nagiosxi/scripts/ModGearmanInstall.sh
chmod +x ModGearmanInstall.sh
./ModGearmanInstall.sh --type=server
```

The `ModGearmanInstall.sh` script will end by telling you the new location of your `module.conf` and `worker.conf`. If you did any custom configuration with Mod Gearman 2, you will want to transfer some of those settings to the new file.

Notes About Server

You now have a Mod-Gearman server installed as an additional NEB module (`gearmand/gearman-job-server`). Nagios XI will hand the execution of checks to Mod-Gearman which will distribute it to the worker(s). Part of the installation is that a local worker was installed on the Nagios XI server and will by default execute checks. If your goal is to offload the checks from your Nagios XI server, you will need to stop the `mod-gearman-worker` service and disable it from being started at boot, please refer to the [Disable Worker](#) section for these steps.

Firewall Rules On Server

If you are going to use external workers, you will need to open the ports on the local firewall on the Nagios XI server to allow the workers to connect.

RHEL 6 | CentOS 6 | Oracle Linux 6

```
iptables -I INPUT -p tcp --dport 4730 -j ACCEPT
iptables -I INPUT -p udp --dport 4730 -j ACCEPT
service iptables save
```

```
ip6tables -I INPUT -p tcp --dport 4730 -j ACCEPT
ip6tables -I INPUT -p udp --dport 4730 -j ACCEPT
service ip6tables save
```

RHEL 7 | CentOS 7 | Oracle Linux 7

```
firewall-cmd --zone=public --add-port=4730/tcp --permanent
firewall-cmd --zone=public --add-port=4730/udp --permanent
firewall-cmd --reload
```

Ubuntu

Ubuntu does not have the firewall enabled by default, however here are the commands if it is enabled:

```
ufw allow proto tcp from any to any port 4730
ufw allow proto udp from any to any port 4730
ufw reload
```

Debian

Debian does not have the firewall enabled by default, however here are the commands if it is enabled:

```
iptables -I INPUT -p tcp --dport 4730 -j ACCEPT
iptables -I INPUT -p udp --dport 4730 -j ACCEPT
```

Worker Installation - New

These steps are completed on the remote worker system(s):

```
cd /tmp
wget https://assets.nagios.com/downloads/nagiosxi/scripts/ModGearmanInstall.sh
chmod +x ModGearmanInstall.sh
./ModGearmanInstall.sh --type=worker
```

When you are prompted with “**What is the IP address of your Nagios master server?**”, type your Nagios XI server address then press Enter, for example 10.25.5.1.

Worker Installation – Upgrade (3 => 3)

If you are already running a previous Mod-Gearman worker, these steps will upgrade it to the latest version and maintain your existing configurations:

```
cd /tmp
rm -f ModGearmanInstall.sh
wget https://assets.nagios.com/downloads/nagiosxi/scripts/ModGearmanInstall.sh
chmod +x ModGearmanInstall.sh
./ModGearmanInstall.sh --type=worker --upgrade
```

Worker Installation – Upgrade (2 => 3)

If you are already running a previous Mod-Gearman 2 worker, these steps will uninstall and upgrade to the newest version:

```
# Remove Mod Gearman 2 worker
cp /etc/mod_gearman2/* /tmp/
yum remove gearmand gearmand-debuginfo gearmand-devel mod_gearman2 -y
# Install Mod Gearman 3 worker
cd /tmp
```

```
rm -f ModGearmanInstall.sh
wget https://assets.nagios.com/downloads/nagiosxi/scripts/ModGearmanInstall.sh
chmod +x ModGearmanInstall.sh
./ModGearmanInstall.sh --type=worker
```

ModGearmanInstall.sh will give you the path to your worker.conf file. If you did any custom configuration of your workers under Mod Gearman 2, you will want to transfer some of these settings to the new file.

Notes About Worker

You now have a worker which can connect to the central Mod-Gearman installation on your Nagios XI server. You may deploy as many workers as needed though typically one worker system will give the desired results. Keep in mind you can also increase the amount of worker processes on this system by editing the worker configuration file by setting the `min-worker` and `max-worker` limits.

Another thing you will need to keep in mind is that if you use NRPE checks on certain hosts, their `nrpe.cfg` must have all of your worker server's IP addresses listed as allowed connections. This means if you are using the xinetd version you need to edit the “`only_from`” section, or if you are using the standard version you must edit the “`allowed_hosts`” section. Similarly if you are using NSClient++ you must add it in the “`allowed_hosts`” section as well to reflect this.

In addition to this you must have all of the plugins which your hosts use to run checks installed on your worker servers.

This guide explains plugins in Nagios XI:

https://assets.nagios.com/downloads/nagiosxi/docs/Managing_Plugins_in_Nagios_XI.pdf

This guide explains how you can install Nagios Plugins on your worker(s):

[Installing Nagios Plugins From Source](#)

Notes About Queues

Understanding how queues work in Mod-Gearman is important to getting the most out of Mod-Gearman. The following KB article explains this in detail and is well worth reading:

[Documentation - Mod-Gearman Queues and Workers](#)

Security Note

Shared key or keyfile needs to be the same on both the workers and gearmand server. A generic key is used in the installation however it's recommended that you change it. On your Nagios XI server edit the both the NEB and Worker configuration files to change the key:

CentOS/RHEL:

```
vi /etc/mod_gearman/module.conf
vi /etc/mod_gearman/worker.conf
```

Debian/Ubuntu:

```
vi /etc/mod-gearman/module.conf
vi /etc/mod-gearman/worker.conf
```

Note that `/etc/mod_gearman/` should be replace with `/etc/mod-gearman/` on Debian-like systems. On remote worker(s) edit the Worker Configuration file to change the key:

CentOS/RHEL:

```
vi /etc/mod_gearman/worker.conf
```

Debian/Ubuntu:

```
vi /etc/mod-gearman/worker.conf
```

The entry in both files is as follows:

```
key=should_be_changed
```


The key shown above is what allows Mod Gearman to send and receive instructions, disabling this is not recommended as any set of instructions can then be injected into the daemon or workers. The key defaults to the value shown above but it is recommended that you change this to something more secure. The key must be at least 7 characters long but no longer than 32 characters.

An alternative to the key is the keyfile. If you do not want your key openly visible in a configuration you may supply a path to the keyfile from which the pass key will be read.

```
keyfile=/path/to/secret.file
```

Be sure to restart Nagios after making changes to the configuration files:

```
service nagios restart
```

You may also start and stop the job server's service and workers with the following commands on the Nagios XI server:

CentOS/RHEL:

```
service gearmand "start/stop/status/restart/reload/condrestart/help"  
service mod-gearman-worker "start/stop/status/restart"
```

Debian/Ubuntu:

```
service gearman-job-server "start/stop/status/restart/reload/condrestart/help"  
service mod-gearman-worker "start/stop/status/restart"
```

Service Start/Stop Order

When stopping or starting Nagios or Mod Gearman services the following order must be obeyed for proper functionality:

Service start order:

CentOS/RHEL:

```
service gearmand start
service mod-gearman-worker start
service nagios start
```

Debian/Ubuntu:

```
service gearman-job-server start
service mod-gearman-worker start
service nagios start
```

Service stop order:

CentOS/RHEL:

```
service nagios stop
service mod-gearman-worker stop
service gearmand stop
```

Debian/Ubuntu:

```
service nagios stop
service mod-gearman-worker stop
service gearman-job-server stop
```

Disable Worker

If you plan on using a remote worker server you may want to stop the local `mod_gearman_worker` service on your Nagios XI server, that way, your remote workers will take over processing. Execute the following commands in a terminal session on your Nagios XI Server:

RHEL/CentOS 5.x/6.x:

```
service mod-gearman-worker stop
chkconfig mod-gearman-worker off
```

RHEL/CentOS 7.x/Debian/Ubuntu:

```
systemctl stop mod-gearman-worker.service
systemctl disable mod-gearman-worker.service
```

Tips

You can view the jobs, processes, and workers by running the following on the Nagios XI server:

```
gearman_top
```

Plugins for monitoring the gearman process can be found at:

https://exchange.nagios.org/directory/Plugins/Clustering-and-High-2DAvailability/check_gearman/details

Additional Resources

If you would like more information on Mod-Gearman please visit the following link:

http://labs.consol.de/nagios/mod-gearman/#_what_is_mod_gearman

Finishing Up

This completes the documentation on how to integrate Mod-Gearman in 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>