Managing Nagios Mod-Gearman Queues and Workers

Purpose

This document describes how to manage Nagios Mod-Gearman (NMG) queues and workers.

Overview

The purpose of this article is to explain how queues work in Nagios Mod-Gearman. Having a clear understanding of this will allow you to distribute the execution of checks to specific workers.

Editing Files

In many steps of this article 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

The Basics

When you install the NMG server broker module on your XI server, by default this also installs a worker module. At this point the checks being handed over to NMG to do the executing instead of the Nagios Core engine.

A NMG worker is what executes the check from Nagios, like ping, CPU load, memory usage etc. A worker can be located on the Nagios XI server or it can be on an external server. Being on an external server means the check load is taken away from the Nagios XI server.

Queues in NMG are how checks are handed out to the workers.

With a default configuration, when nagios starts it hands off the host and service checks to NMG.

NMG creates two queues called host and service. You can think of these queues as the default "catch all" queues.

By default, any worker that connects will execute checks from these queues.

It is important to remember that an external worker needs all of the plugins installed on it so it can execute the checks that are handed to it.

Configuration File Location

NMG has two separate configuration files:

```
module.conf
```

This is the configuration file for the Nagios server Broker Module.

```
worker.conf
```

This is the configuration file for the worker. The worker exists on the Nagios server as well as your remote worker(s).

Both of these files exist in the same location:

RHEL | CentOS | Oracle Linux | Ubuntu | Debian

```
/etc/nagios-mod-gearman/
```

Remote Worker Considerations

When you add a remote worker, there are some things that need to be taken into account. The most important question you need to ask is:

"What checks should NOT be executed by the remote worker?"

Why do you need to ask such a question?

Let’s look at the standard checks that the Nagios XI server has built in for the **localhost** object:

- Current Load
- Current Users
- HTTP
- PING
- Root Partition
- SSH
- Swap Usage
- Total Processes

Let's look at the service "Root Partition". The command it executes is:

```
/usr/local/nagios/libexec/check_disk -w 20% -c 10% -p /
```

If a **remote worker** was to execute this check, the results that came back would be for the remote workers root partition, not the root partition of the Nagios XI server.

So it's pretty obvious that you don't want these checks being executed on the remote workers, how do you configure NMG to do this? The key to this is with host groups and service groups.

One other important point to be made about remote workers is that the plugins need to be installed on the workers. NMG is passing the command that needs to be executed, the worker executes the command so it needs to be able to execute the plugin.

Host groups and Service groups

The first step to stop NMG from sending local checks to remote workers is to create a host group or service group in Nagios that contains the objects you don't want executed on remote workers.

What's the difference between a host group and a service group in NMG?

- **host group**
  - If you use a host group in a NMG configuration, NMG will automatically include the services for the hosts in the host group
  - This allows for simple configurations
- **service group**
  - Using a service group in a NMG configuration allows for more granular control of what services are handled by NMG

Stopping Gearman Checks for a Group

For the purpose of simplicity, we will focus on a host group.

- Log into Nagios XI and navigate to **Configure > Core Configuration Manager**
- Under **Monitoring** click **Host Groups**
- Click the **+ Add New** button
  - Name: **nmg_objects_local**
  - Description: **Nagios-Mod-Gearman Objects - Local**
  - Click the **Manage Hosts** button
    - Add **localhost** to the *right hand* side

        - Click **Close**
    - Click **Save**
- Click the **Apply Configuration** button

Now we need to configure NMG to use this host group to exclude checks. On your Nagios XI server edit the file `module.conf`.

Find the section like this:

```
# sets a list of hostgroups which will not be executed
# by gearman. They are just passed through.
# Default is none
localhostgroups=
```

Update the line as follows:

```
localhostgroups=nmg_objects_local
```

Save the `module.conf` file.

Now you need to restart some services:

RHEL | CentOS Stream | Oracle Linux | Ubuntu | Debian

```
systemctl restart nagios.service
```

From now on, NMG will not execute any host or service checks for the hosts in the `nmg_objects_local` group, instead Nagios Core will execute them as it normally would.

What other checks should be prevented from executing on a remote worker?

Any devices that are being monitored by the "Network Switch / Router Wizard" will also need to be added to the `nmg_objects_local` group, as they target some files specific to MRTG which is run locally on the Nagios XI server. Once you add these hosts to the `nmg_objects_local` group, when you Apply Configuration Nagios will restart and NMG will then know about the updated host group.

Using A Worker for A Specific Site

In this scenario, you have a remote site where you want all the checks to be executed at the remote site by the NMG worker at the remote site.

In this example, we will focus on a host group.

- Log into Nagios XI and navigate to **Configure > Core Configuration Manager**
- Under **Monitoring** click **Host Groups**
- Click the **+ Add New** button
    - Name: **nmg_objects_site_a**
    - Description: **Nagios-Mod-Gearman Objects - Site A**
    - Click the **Manage Hosts** button
        - Add **<all the hosts at the remote site>** to the *right hand* side
        - Click **Close**
    - Click **Save**
- Click the **Apply Configuration** button

Before we configure the NMG server to use this host group as a queue, we'll configure the remote worker.

Most importantly, we ONLY want this worker to execute checks for the hosts in the host group. With that in mind, we'll configure the worker so it doesn't touch the default host and services queues.

On your remote worker edit the file `worker.conf`

Find the sections like this:

```
# defines if the worker should execute
# service checks.
services=yes

# defines if the worker should execute
# host checks.
hosts=yes
```

Change them to **no** as follows:

```
# defines if the worker should execute
# service checks.
services=no

# defines if the worker should execute
# host checks.
hosts=no
```

Find the sections like this:

```
# sets a list of hostgroups which this worker will work
# on. Either specify a comma separated list or use
# multiple lines.
#hostgroups=name1
#hostgroups=name2,name3
```

Add this line as follows:

```
hostgroups=nmg_objects_site_a
```

Save the `worker.conf` file.

You need to restart the worker service:

RHEL | CentOS Stream | Oracle Linux | Ubuntu | Debian

```
systemctl restart nagios-mod-gearman-worker.service
```

Now we need to configure the NMG server to use this host group as a queue. On your Nagios XI server edit the file `/etc/nagios-mod-gearman/module.conf`

Find the section like this:

```
# sets a list of hostgroups which will go into seperate
# queues. Either specify a comma seperated list or use
# multiple lines.
#hostgroups=name1
#hostgroups=name2,name3
```

Add this line as follows:

```
hostgroups=nmg_objects_site_a
```

Save the `module.conf` file.

Now you need to restart some services:

RHEL | CentOS Stream | Oracle Linux | Ubuntu | Debian

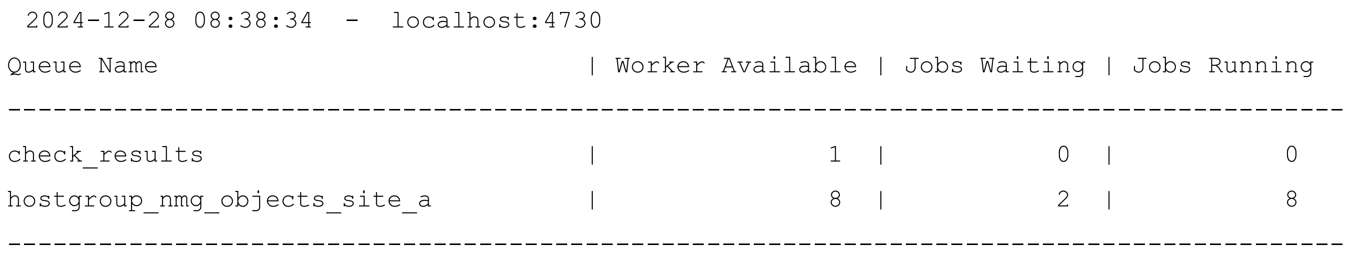
```
systemctl restart nagios.service
```

From now on, NMG will allocate any host or service checks for the hosts in the `nmg_objects_site_a` group into a queue. Any workers that are configured to target it will execute them.

You can see this new queue by using the nagios gearman top command on the XI server:

```
nagios-gearman-top
```

It will look like the example below:



```
 2024-12-28 08:38:34  -  localhost:4730
Queue Name                            | Worker Available | Jobs Waiting | Jobs Running
---------------------------------------------------------------------------------------
check_results                         |               1  |            0  |            0
hostgroup_nmg_objects_site_a          |               8  |            2  |            8
---------------------------------------------------------------------------------------
```

Ensure Worker Doesn't Touch Host And Service Queues

This was briefly mentioned in the "Using A Worker For A Specific Site" scenario, however it is worth re-iterating.

The goal is that you ONLY want the worker to execute checks in the specific queues that it has been configured for, using the hostgroups and/or servicegroups directives. You DO NOT want it executing checks in the default host and service queues.

Why would you want this type of configuration?

Let's say you have a remote worker that is executing checks for the devices located at that physically remote location. The local worker on the Nagios XI server is executing the host and service queues. If the remote worker was executing checks for the host and service queues, the following would happen:

- Remote worker gets plugin command to execute and executes it
- The host address is actually for a device back at the Nagios XI server location
- Network traffic is generated to go back across to the Nagios XI server location to perform the plugin check against the host
- Network traffic is generated to return the plugin result to the remote worker
- Network traffic is generated to from the remote worker to return the plugin result back to the NMG server on the Nagios XI server

You can see that there is a lot of unnecessary traffic being generated across network links. Hence it is important when have remote workers at physically different locations, they need to be correctly configured so they don't touch the host and service queues.

On your worker edit the file `/etc/nagios-mod-gearman/worker.conf`

Find the sections like this:

```
# defines if the worker should execute
# service checks.
services=yes

# defines if the worker should execute
# host checks.
hosts=yes
```

Change them to **no** as follows:

```
# defines if the worker should execute
# service checks.
services=no

# defines if the worker should execute
# host checks.
hosts=no
```

Save the `worker.conf` file.

You need to restart the worker service:

RHEL | CentOS Stream | Oracle Linux | Ubuntu | Debian

```
systemctl restart nagios-mod-gearman-worker.service
```

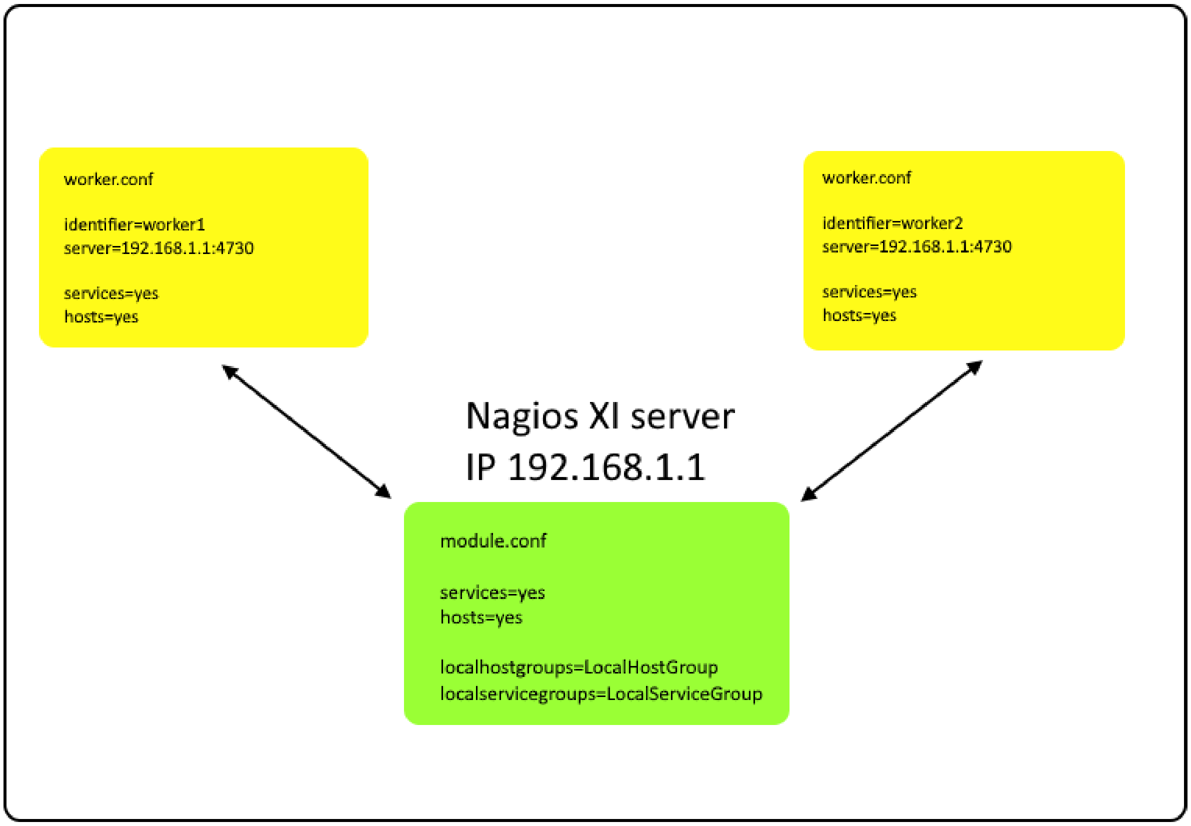
In the following sections we'll discuss load balancing and distributed monitoring with NMG.

Load Balancing

The easiest setup to do is a simple load balancing. For example, if your single Nagios server just cannot handle the load, you could just add a worker or 2 in the same network to reduce your load on the Nagios server.

Therefore we just enable hosts and services on the server and the workers.

Load Balancing



```
worker.conf

identifier=worker1
server=192.168.1.1:4730

services=yes
hosts=yes
```

```
worker.conf

identifier=worker2
server=192.168.1.1:4730

services=yes
hosts=yes
```

```
module.conf

services=yes
hosts=yes

localhostgroups=LocalHostGroup
localservicegroups=LocalServiceGroup
```

In the above example, there are 2 Nagios Mod-Gearman workers called **worker1** and **worker2** that contact the contact the Nagios XI server at IP address 192.168.1.1.

Both the `services=` and `hosts=` options in the `/etc/nagios-mod-gearman/worker.conf` file are set to `yes` which means they will run any host or service check that is scheduled on the XI server.

The `identifier=` option is the name of the worker you want to use for identification and the `server=` option has to be changed to use the Nagios server's IP address.

On the Nagios server, the same settings in the `/etc/nagios-mod-gearman/module.conf` file for the `services=` and `hosts=` option has to be set to `yes` which will enable those checks to be available to be ran by a Nagios Mod Gearman worker.

In the picture, it shows these two options set:

`localhostgroups=` and `localservicegroups=`

These two settings optional but you may need to set them in your environment.

Certain checks run by Nagios cannot be run on remote workers because they need to access files or directories on the XI server which do not exist on the workers.

One example is the localhost checks in XI. They all should be run on the XI server as they are configured to check the XI server.

So what you would do is create a Host Group in XI called `LocalHostGroup` and add the localhost host to it. That way, the checks will not be run on a worker.

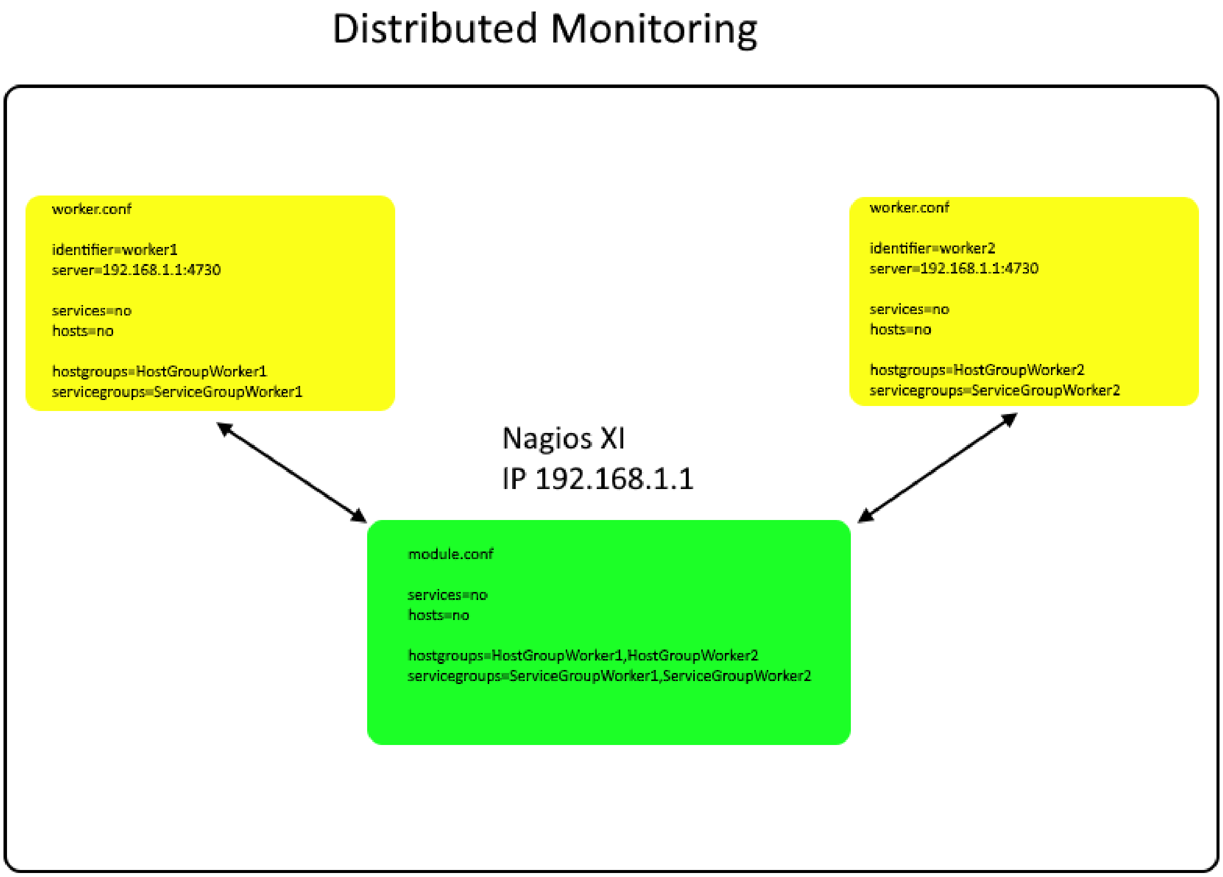
Another example is Bandwidth checks, they do need to be run on the XI server so you would add those hosts to the `LocalHostGroup` to ensure that they are run on the XI server.

The `localservicegroups=` option is similar to the above except it is used for Services. If you only need specific services to run but the other checks can be ran anywhere, you would create a ServiceGroup in XI called `LocalServiceGroup` and add the specific services to it so they are run by the XI server and not the workers.

Distributed Monitoring

If your checks have to be run from different network segments, then you can use the hostgroups (or servicegroups) to define a hostgroup for specific worker. The general hosts and services queue is disabled for this worker and just the hosts and services from the given hostgroup or servicegroup will be processed.

This will also reduce load on the Nagios server and give the ability to access remote networks using a Nagios Mod-Gearman Worker that the XI server cannot directly access itself.



```
worker.conf

identifier=worker1
server=192.168.1.1:4730

services=no
hosts=no

hostgroups=HostGroupWorker1
servicegroups=ServiceGroupWorker1
```

```
worker.conf

identifier=worker2
server=192.168.1.1:4730

services=no
hosts=no

hostgroups=HostGroupWorker2
servicegroups=ServiceGroupWorker2
```

```
module.conf

services=no
hosts=no

hostgroups=HostGroupWorker1,HostGroupWorker2
servicegroups=ServiceGroupWorker1,ServiceGroupWorker2
```

In the above example, there are 2 Nagios Mod Gearman workers called **worker1** and **worker2** that contact the Nagios XI server at IP address 192.168.1.1.

Both the `services=` and `hosts=` options in the `/etc/nagios-mod-gearman/worker.conf` file are set to no which means they will not run any Host or Service check that is scheduled on the XI server on the worker.

The `identifier=` option is the name of the worker you want to use for identification and the `server=` option has to be changed to use the Nagios server's IP address

On the Nagios server, the same settings in the `/etc/nagios-mod-gearman/module.conf` file for the `services=` and `hosts=` option has to be set to no which will disable those checks to be available to be ran by a Nagios Mod Gearman worker. You will also have to set the `hostgroups=` and the `servicegroups=` options for the checks you want to run on the remote workers.

In the picture, it shows these 2 options set on both the workers and the XI server.

`hostgroups=`

This sets a list of hostgroups which this worker will run.

`servicegroups=`

This sets a list of servicegroups which this worker will run.

If you look at **worker1**, it is setup to only run checks for the Hosts that are set up in the `HostGroupWorker1` Host Group that is setup on the XI server. And it will run the Service Checks for the hosts in that same group.

The `servicegroups=ServiceGroupWorker1` setting will only run the services in that group and not the Host checks for the services.

The same for **worker2** except it is setup to run checks from the groups that end in 2. `HostGroupWorker2` and `ServiceGroupWorker2`.

Finishing Up

This completes the documentation on Managing Nagios Mod-Gearman Queues and Workers. If you have additional questions or other support-related questions, please visit us at our Nagios Support Forum, Nagios Knowledge Base, or Nagios Library:

Visit Nagios Support Forum Visit Nagios Knowledge Base Visit Nagios Library