How to Manage Nagios Mod-Gearman in the Nagios XI Interface

Purpose

This document describes how to use Nagios Mod-Gearman (NMG) in the Nagios XI web UI.

NMG is pre-loaded in Nagios XI 2024R2.1+.

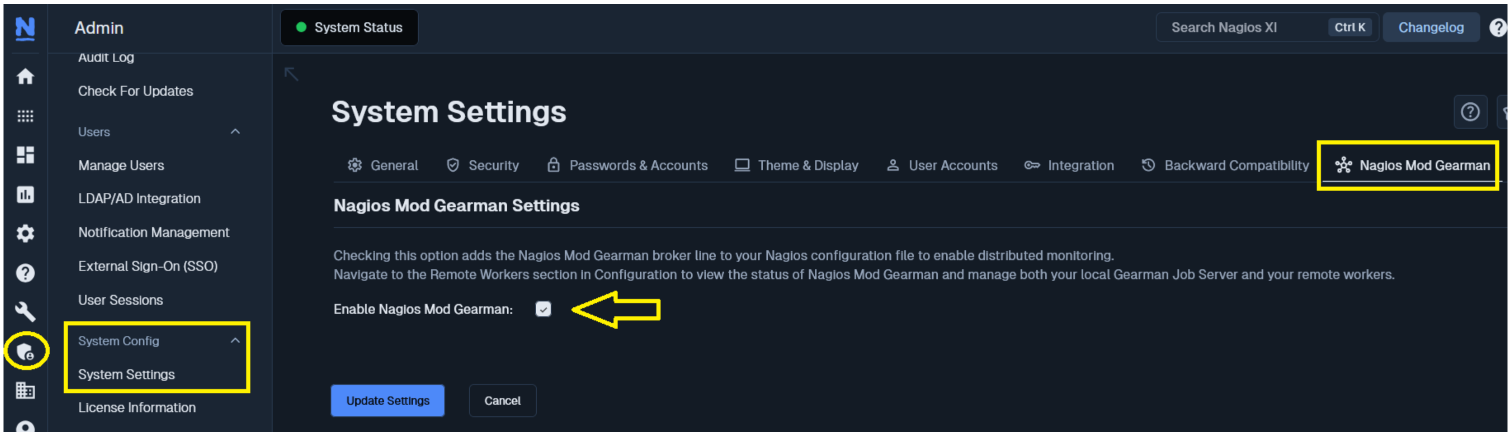
NMG is a Nagios Event Broker (NEB) module that can be used for a variety of purposes including load reduction, load balancing, and distributed monitoring.

Important Note: The in-interface Remote Worker configuration option is one of Nagios XI's Premium features, so requires active support and maintenance benefits to function. For questions about renewing your benefits if they have lapsed, please email **sales@nagios.com** so we can assist you further.

Although it is still possible to configure Nagios Mod-Gearman manually from the command line, the Remote Workers UI option significantly simplifies setup and management of your NMG server and workers.

Activating Nagios Mod-Gearman

To turn on Nagios Mod-Gearman, navigate to **Admin > System Config > System Settings**, and click the **Nagios Mod Gearman** tab, then check the **Enable Nagios Mod Gearman** checkbox. This will add a special broker line to your Nagios config file to enable distributed monitoring.



System Settings

Nagios Mod Gearman Settings

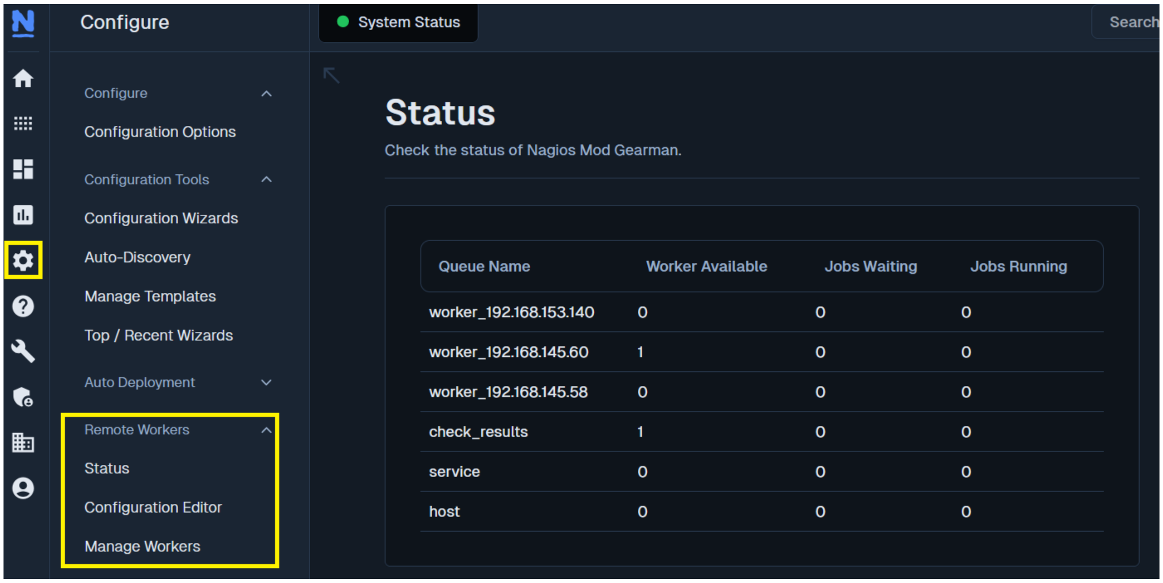
Checking this option adds the Nagios Mod Gearman broker line to your Nagios configuration file to enable distributed monitoring.
Navigate to the Remote Workers section in Configuration to view the status of Nagios Mod Gearman and manage both your local Gearman Job Server and your remote workers.

Enable Nagios Mod Gearman: ☑

Managing Nagios Mod-Gearman Server and Workers

To begin managing your NMG settings and workers, navigate to **Configure > Remote Workers**.

The **Status** section provides an overview of queues, available workers, jobs waiting, and jobs running across your NMG workers:



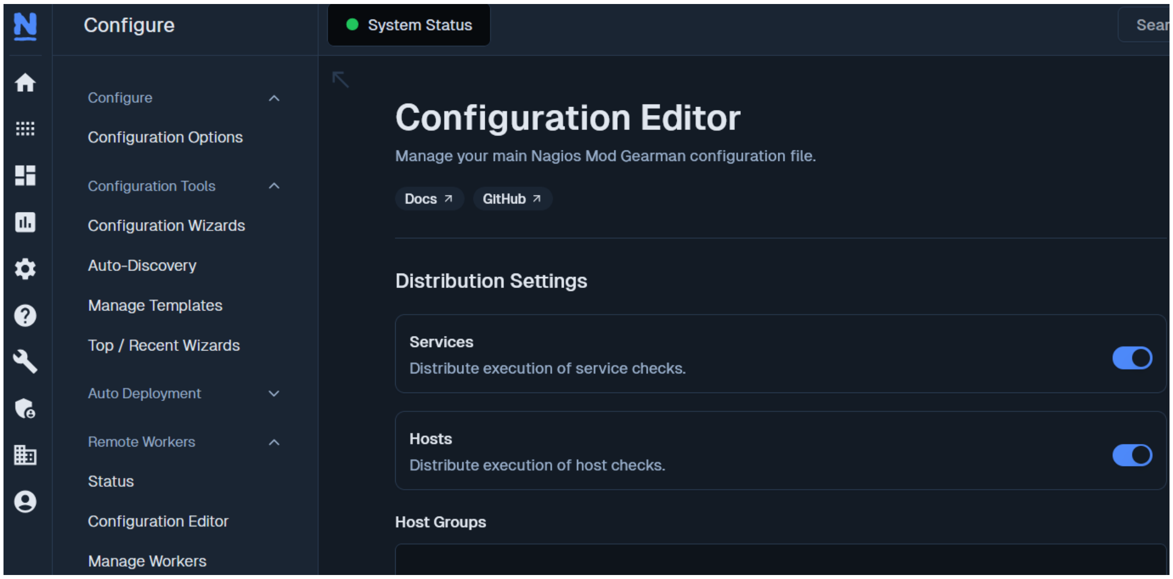
Status

Check the status of Nagios Mod Gearman.

Queue Name	Worker Available	Jobs Waiting	Jobs Running
worker_192.168.153.140	0	0	0
worker_192.168.145.60	1	0	0
worker_192.168.145.58	0	0	0
check_results	1	0	0
service	0	0	0
host	0	0	0

Configuration Editor

This section is used to manage your main NMG configuration file.



Configuration Editor

Manage your main Nagios Mod Gearman configuration file.

Distribution Settings

Services
Distribute execution of service checks.

Hosts
Distribute execution of host checks.

Host Groups

There are a lot of settings here, so we'll provide details on the function of each:

Distribution Settings

- **Services/Hosts on/off**: determines whether the module distributes execution of host and service checks to be run on Nagios Mod-Gearman Workers.
- **Hostgroups/Servicegroups**: lists of hostgroups and servicegroups which will go into separate queues to be run on Nagios Mod-Gearman Workers. This can be a comma separated list if you wish to add multiple entries.
- **Do Hostchecks**: turn this off if you want NMG to only take care of service checks. No host checks will be processed by NMG. Use this option to disable host checks and still have the possibility to use hostgroups for easy configuration of your services. If turned on (as it is by default), you still have to define which host checks should be processed by either using the **Hosts** or the **Host Groups** options.

Server Settings

- **Server:** the address of your Gearman Job Server, aka the Nagios XI server.
- **Duplicate Server:** the address of a second Gearman job server. NMG uses the first server available.

Encryption Settings

- **Encryption:** enables or disables encryption. It is strongly advised to not disable encryption. Anybody will be able to inject packages to your worker. Encryption is enabled by default and you have to explicitly disable it.
- **Key:** the shared password for communication encryption. **Note** that if you change the Key here, you must update the Key setting on each worker, then **Sync** the changes to the worker so they can connect to the Nagios server so the NMG workers can continue to function.
- **Key File:** the encryption keyfile location, if this is used.

Logging & Debugging

- **Debug on/off:** turning this on will increase the verbosity of the module logging, switching it from debug level 0 to debug level 1.
- **Log File:** the location where module logs will be stored
- **Log Stats Interval:** the interval in seconds to log NMG submission statistics

Local Settings

- **Local Host Groups:** a list of hostgroups that will **not** be executed by NMG. They will be passed through by the module. This can be a comma separated list if you wish to add multiple entries.
- **Local Service Groups:** a list of servicegroups that will **not** be executed by NMG. They will be passed through by the module. This can be a comma separated list if you wish to add multiple entries.

Orphan Settings

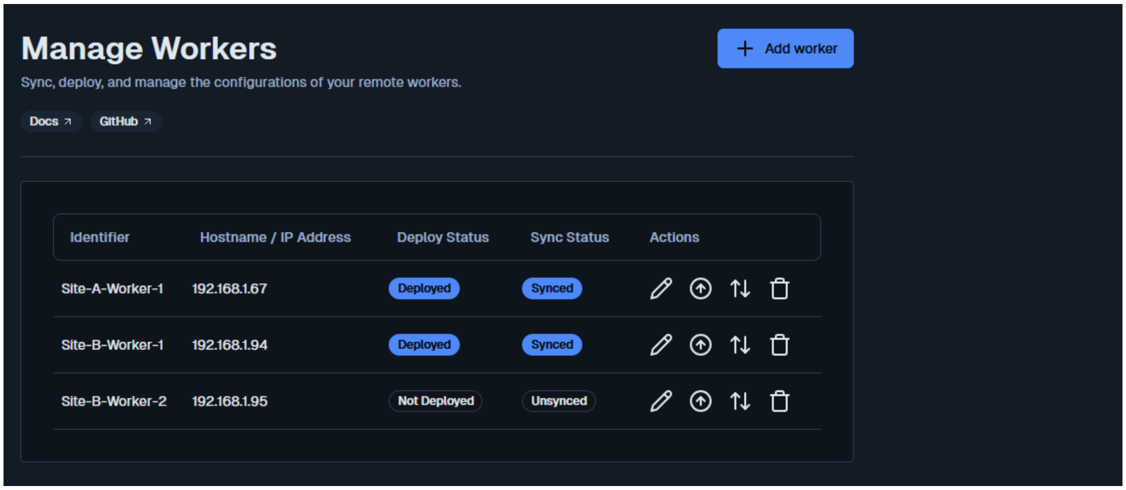
- **Orphan Host Checks:** turn on to submit a fake result for orphaned host checks. The Nagios-Mod-Gearman NEB module will submit a fake result for orphaned host checks with a message saying there is no worker running for this queue.
- **Orphan Service Checks:** turn on to submit a fake result for orphaned service checks. The Nagios-Mod-Gearman NEB module will submit a fake result for orphaned service checks with a message saying there is no worker running for this queue.
- **Orphan Return:** the desired return code for orphaned checks.
- Refer to the Troubleshooting Orphaned Checks section for more information on resolving this issue.

Additional Settings

- **Queue Custom Variable:** define the target queue by a custom variable in addition to hostgroups and servicegroups
- **Result Workers:** defines the number of check_results workers on your XI server. This enables the broker to receive more check statuses and output from workers, which the Nagios Core engine can use to process more checks at once. Set to `0` to disable Nagios from processing any checks from the NMG workers.
- **Use Unique Jobs toggle:** prevent the queue from filling up when there is no worker.
- **Accept Clear Results:** accept unencrypted results from workers.
- **Gearman Connection timeout:** connection timeout (in milliseconds) while submitting jobs to gearmand server (-1 for no timeout).

Managing Workers

In the **Manage Workers** menu, you can add, deploy, edit, sync, and delete your NMG workers.



Manage Workers

Sync, deploy, and manage the configurations of your remote workers.

Identifier	Hostname / IP Address	Deploy Status	Sync Status	Actions
Site-A-Worker-1	192.168.1.67	Deployed	Synced	✎ ⊕ ⇅ 🗑
Site-B-Worker-1	192.168.1.94	Deployed	Synced	✎ ⊕ ⇅ 🗑
Site-B-Worker-2	192.168.1.95	Not Deployed	Unsynced	✎ ⊕ ⇅ 🗑

Important Note: Network Access to Workers

For the **Deploy** and **Sync** functions to work, your Nagios XI server will need an inbound network path to the worker server on Port 22 (SSH). If this is not possible or desirable, you can choose to install and manage workers manually from the CLI. Worker communication for checking queues and sending results is inbound to Nagios XI from the worker, so an inbound path to the worker from XI is not necessary for NMG itself to work.

Keep in mind that the same encryption **Key** you define in the **Configuration Editor** in Nagios XI for your NMG Server will need to be set on the workers.

You can find instructions for installing and configuring NMG workers manually here on page 5:

Integrating Nagios Mod-Gearman with Nagios XI

Important Note: Plugins on Workers

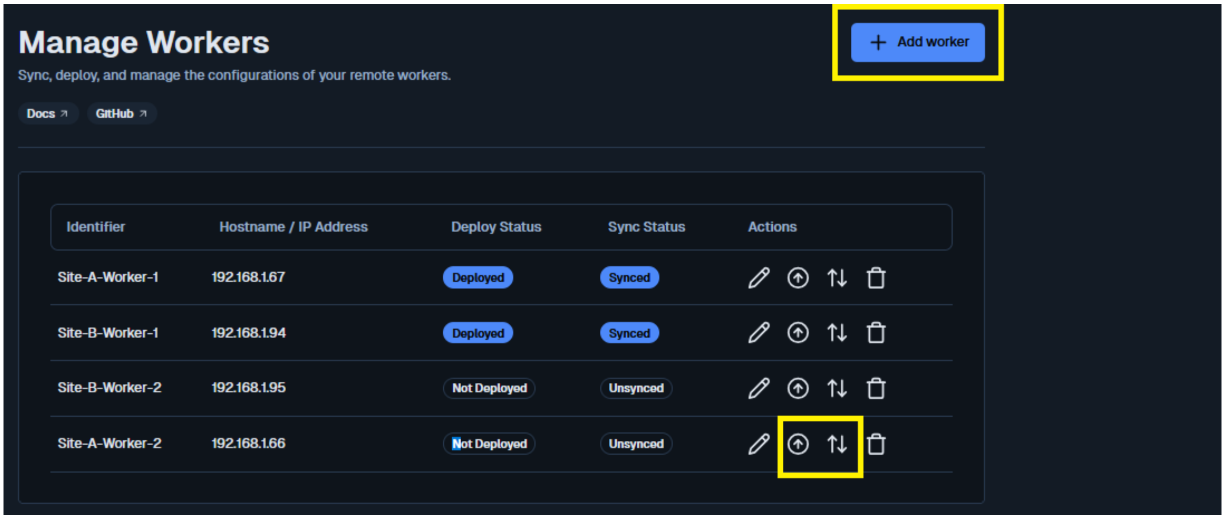
You must also have all of the plugins used to run your checks installed on your worker servers.
The default Nagios XI plugins are automatically installed on workers when they are Deployed, though some may require extra steps depending on the Linux OS chosen for the worker.
However, you will need to add any custom plugins and the requirements for the custom plugins you use on any workers that will handle those checks.

Important Note: Supported Linux Operating Systems for Workers

Debian-based Linux operating systems are currently **not** supported by the UI worker management function, but may be in the future.

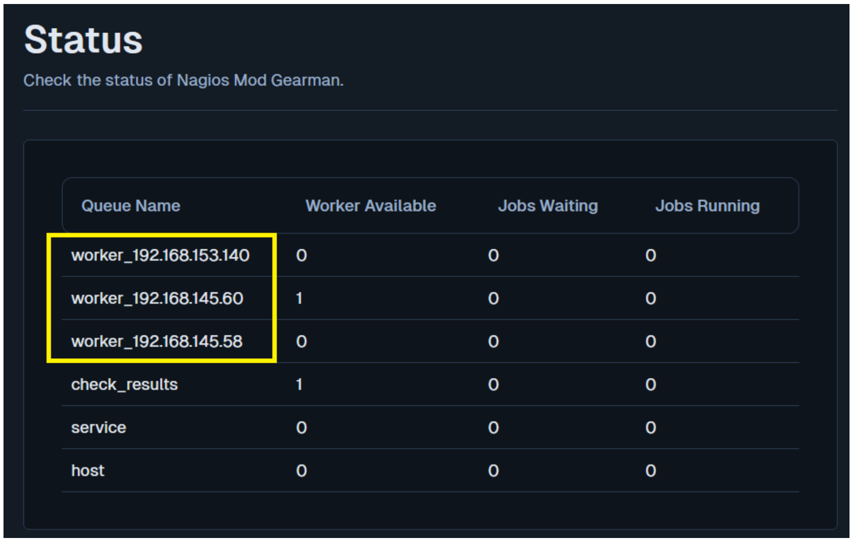
Adding a Worker

1. To begin, navigate to the **Manage Workers** section, and click **+ Add worker** on the upper right.
2. Configure the worker settings.
3. Click the **Deploy Worker** icon to push and deploy the new worker to your target worker host.
4. Once deployed, click the **Sync** (two arrows) icon to sync your custom settings to the worker.
5. Note that until you click **Deploy Worker** the worker will not be installed on the target host, and until you click **Sync** your worker settings will not be applied to it.


Manage Workers
Sync, deploy, and manage the configurations of your remote workers.
Docs ↗
GitHub ↗
+ Add worker

Identifier	Hostname / IP Address	Deploy Status	Sync Status	Actions
Site-A-Worker-1	192.168.1.67	Deployed	Synced	✎ ⊕ ⇅ 🗑
Site-B-Worker-1	192.168.1.94	Deployed	Synced	✎ ⊕ ⇅ 🗑
Site-B-Worker-2	192.168.1.95	Not Deployed	Unsynced	✎ ⊕ ⇅ 🗑
Site-A-Worker-2	192.168.1.66	Not Deployed	Unsynced	✎ ⊕ ⇅ 🗑

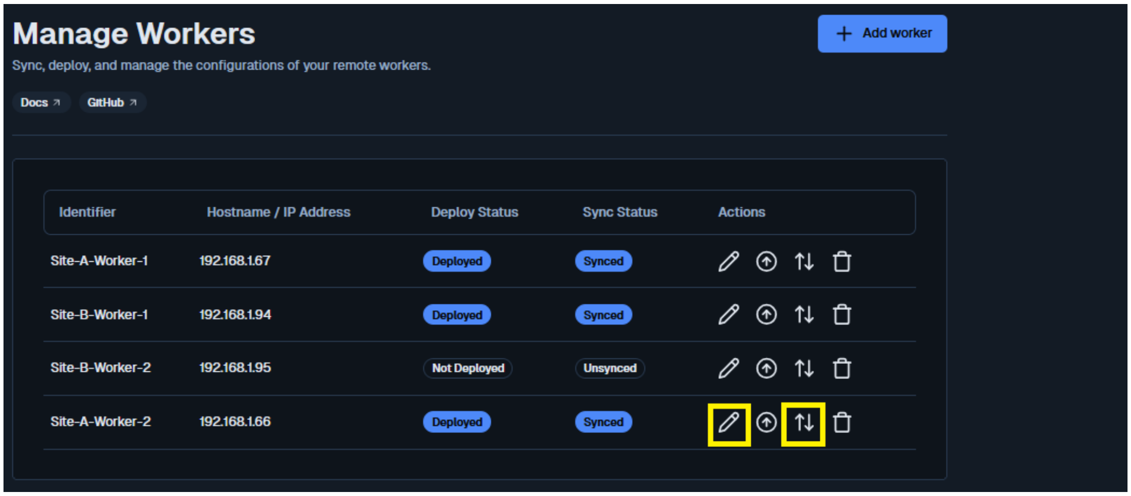
6. You can visit the **Status** page to verify the worker queue is working. If it is, you will see `worker_<new_worker_identifier>` in the table



Queue Name	Worker Available	Jobs Waiting	Jobs Running
worker_192.168.153.140	0	0	0
worker_192.168.145.60	1	0	0
worker_192.168.145.58	0	0	0
check_results	1	0	0
service	0	0	0
host	0	0	0

Editing a Worker

1. Click the **Edit** (pencil) icon to the right of the worker.
2. Modify the necessary worker settings.
3. Click the **Sync** (two arrows) icon to push your changes to the worker.



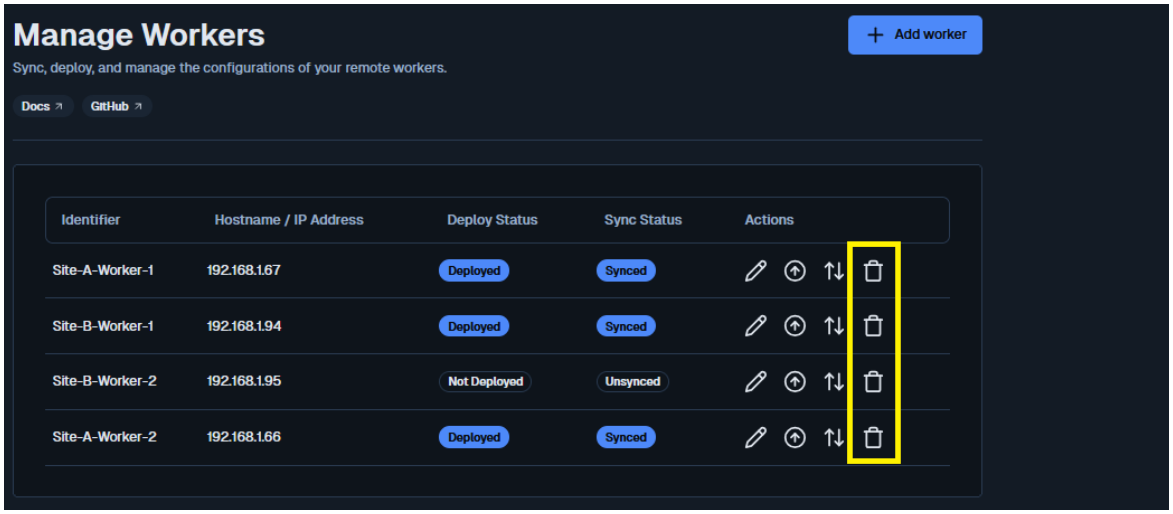
Manage Workers

Sync, deploy, and manage the configurations of your remote workers.

Identifier	Hostname / IP Address	Deploy Status	Sync Status	Actions
Site-A-Worker-1	192.168.1.67	Deployed	Synced	✎ ⊕ ⇅ 🗑
Site-B-Worker-1	192.168.1.94	Deployed	Synced	✎ ⊕ ⇅ 🗑
Site-B-Worker-2	192.168.1.95	Not Deployed	Unsynced	✎ ⊕ ⇅ 🗑
Site-A-Worker-2	192.168.1.66	Deployed	Synced	✎ ⊕ ⇅ 🗑

Deleting a Worker

To delete a worker, simply click the trashcan icon on the far right. This will delete the local setings, and remove NMG from the worker.



Manage Workers

Sync, deploy, and manage the configurations of your remote workers.

Identifier	Hostname / IP Address	Deploy Status	Sync Status	Actions
Site-A-Worker-1	192.168.1.67	Deployed	Synced	✎ ⊕ ⇅ 🗑
Site-B-Worker-1	192.168.1.94	Deployed	Synced	✎ ⊕ ⇅ 🗑
Site-B-Worker-2	192.168.1.95	Not Deployed	Unsynced	✎ ⊕ ⇅ 🗑
Site-A-Worker-2	192.168.1.66	Deployed	Synced	✎ ⊕ ⇅ 🗑

Worker Settings

Workers have a lot of possible settings too, so we'll detail each. Note that outside of special use cases, you should only need to define the **Basic** and **Distribution Settings**. You can keep the defaults for everything else.

Basic Settings

- **Identifier**: a friendly name for the worker to help you easily identify it.
- **Hostname or IP Address:** the IP address or FQDN of the target worker host.
- **SSH Username:** the system username that will be used to connect via SSH. Note that this is the username that will be used to install the worker, so must be root or root equivalent.
- **SSH Password:** the password for the system username.

Distribution Settings

- **Services/Hosts toggles**: define whether the module should distribute service and host checks to the worker.
- **Hostgroups/Servicegroups:** define the hostgroups and servicegroups that should be distributed to the worker. These can be comma separated lists if you wish to add multiple entries.
- **Note** that the gearman worker service (`nagios-mod-gearman-worker`) will fail on the worker if no queues (services, hosts, hostgroups, or servicegroups) are enabled for the worker. You must assign at least one queue for the service to start on the worker so it will show up in the **Status** page.

Encryption Settings

- **Encryption toggle:** enables or disables encryption. It is strongly advised to not disable encryption. Anybody will be able to inject packages to your worker. Encryption is enabled by default and you have to explicitly disable it.
- **Key:** the shared password for communication encryption. This needs to be the same **Key** you set in the **Configuration Editor** settings.
- **Key File:** the encryption keyfile location, if this is used.

Server Settings

- **Server:** the address of your Gearman Job Server, aka the Nagios XI server.
- **Duplicate Server**: the address of a second Gearman Job Server. NMG uses the first server available.

Logging & Debugging

- **Debug toggle**: turning this on will increase the verbosity of the module logging, switching it from debug level 0 to debug level 1.
- **Log File**: the location where module logs will be stored.

Advanced Settings

- **PID File:** custom path to the PID file
- **Job Timeout:** default job timeout in seconds. By default the worker will use the values from the core for host and service checks.
- **Min Workers:** the minimum number of worker processes which should run at any time.
- **Max Workers:** the maximum number of worker processes which should run at any time. You may set this equal to min-worker setting to disable dynamic starting of workers. When setting this to 1, all services from this worker will be executed one after another.
- **Idle Timeout:** time in seconds after which an idling worker exits. This parameter controls how fast your waiting workers will exit if there are no jobs waiting. Set to 0 to disable the idle timeout.
- **Max Jobs:** controls the amount of jobs a worker will do before he exits. Use this to control how fast the amount of workers will go down after high load times. Disabled when set to 0.
- **Spawn Rate:** defines the rate of spawned worker per second as long as there are jobs waiting.
- **Fork on Exec toggle:** use this option to disable an extra fork for each plugin execution. Disabling this option will reduce the load on the worker host, but may cause trouble with unclean plugins.
- **Load Limit 1, 5, and 15:** set a limit based on the 1min, 5 min, and 15 min load average. When exceeding the load limit, no new worker will be started until the current load is below the limit. No limit will be used when set to 0.
- **Show Error Output toggle:** use this option to show stderr output of plugins too. When set to no, only stdout will be displayed.

- **Timeout Return:** defines the return code for timed out checks. Accepted return codes are 0 (Ok), 1 (Warning), 2 (Critical) and 3 (Unknown).
- **Dup Results Are Passive toggle:** use this option to set if the duplicate result sent to the 'dupserver' will be passive or active.
- **Gearman Connection Timeout:** timeout in milliseconds when connecting to gearmand daemon.
- **Restrict Path:** allows you to restrict this worker to only execute plugins from a particular folder. *Note that only a single path can be defined.* By default this is left blank.
- **Restrict Command Characters:** when Restrict Path is active, no shell will be spawned and no shell characters (`$&();<>`"'|`) are allowed in the command line itself. By default this is left blank.
- **Workaround RC 25 on/off:** duplicate jobs from gearmand result sometimes in exit code 25 of plugins because they are executed twice and get killed because of using the same resource. Sending results (when exit code is 25) will be skipped with this enabled. Only needed if you experience problems with plugins exiting with exit code 25 randomly.

Troubleshooting Orphaned Checks

Several factors can cause one of the following orphaned errors in service or host details:

```
service check orphaned, is the nagios-mod-gearman worker on queue 'service' running?
```

```
host check orphaned, is the nagios-mod-gearman worker on queue 'service' running?
```

1. The worker machine may be down. Verify that the worker machine is running and there are no errors in the OS, hardware, etc. You will want to work with your IT team to resolve hardware, Linux and component issues that may be interfering with the functionality of the worker machine.

2. The worker is not running on the worker machine. Verify the Nagios Mod Gearman worker is running the worker machine.

Test by executing this command as root on the worker machine:

```
systemctl status nagios-mod-gearman-worker
```

If the worker is running, you'll see results like the following:

```
nagios-mod-gearman-worker.service - Nagios-Mod-Gearman Worker
Loaded: loaded (/usr/lib/systemd/system/nagios-mod-gearman-worker.service;
enabled; preset: enabled)
Active: active (running) since DDD YYYY-MM-DD HH:MM:SS UTC; Hh MMmin ago
```

If the worker is not running, enable the Nagios Mod-Gearman Worker to start at boot and start the worker, by running the following as root:

```
systemctl enable nagios-mod-gearman-worker
systemctl start nagios-mod-gearman-worker
```

Also note that the gearman worker service will fail on the worker if no queues (services, hosts, hostgroups, or servicegroups) are enabled for the worker. You must assign at least one queue for the service to start on the worker so it will show up in the **Status** page.

Refer to the Distribution Settings section of Worker settings for more information.

3. The network connection from the worker machine to the XI instance is not working. Verify that network connectivity from the host machine to the XI instance is working.

Test by executing this command on the worker machine (where `xxx.xxx.xxx.xxx` is the IP address of the XI instance). You may need to install NMAP (https://nmap.org/) .

```
nmap xxx.xxx.xxx.xxx -p 4730
```

If the connection is working, you should see output like the following:

```
PORT     STATE SERVICE
4730/tcp open  gearman
```

If a different output/error is generated above, you will need to work with your IT/Networking team to ensure the worker machine can connect back to the XI instance on port 4730.

4. Configuration issues with the worker.

Verify that the server field in `/etc/nagios-mod-gearman/worker.conf` contains the XI instance IP (where `xxx.xxx.xxx.xxx` is the IP address of the XI instance):

```
server=xxx.xxx.xxx.xxx:4730
```

5. Check the worker log file for additional errors and address, if possible:

```
/var/log/nagios-mod-gearman/nagios-mod-gearman-worker.log
```

Extra Plugin Steps

check_vsphere_xi.py on RHEL/OEL 8 Workers

You may see the following error in XI for the VSphere plugin when it runs on a Gearman Worker:

```
File "/usr/local/nagios/libexec/check_vsphere_xi.py", line 8, in <module> from pyVim import connect

File "/usr/local/lib/python3.6/site-packages/pyVim/connect.py", line 23, in <module> from pyVmomi.five import PY3, HTTPConnection, HTTPSConnection

File "/usr/local/lib/python3.6/site-packages/pyVmomi/__init__.py", line 14, in <module> from . import _init_utils # noqa: E402

File "/usr/local/lib/python3.6/site-packages/pyVmomi/_init_utils.py", line 8, in <module>

raise Exception(msg)

Exception: Python 3.9 or newer is required (found 3.6)]
```

To resolve the error, run the following as root on the NMG Worker to remove the newer Python modules that require a newer version of Python to be installed by default:

RHEL 8

```
pip3 uninstall pyVmomi
pip3 uninstall pyvim
dnf install python3-pyvim
```

OEL 8

```
pip3 uninstall pyVmomi
pip3 uninstall pyvim
dnf install python3-pyvim
dnf install php-xml
```

Finishing Up

This completes the documentation on How to Manage Nagios Mod-Gearman in the Nagios XI Interface. 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