



Lighthouse 5 User Guide

Revision 5.2.2

2018-09-12

TABLE OF CONTENTS

1. About this User Guide	6	
2. Lighthouse overview	7	
2.1 <i>Lighthouse VM 5 host requirements</i>		7
2.2 <i>Lighthouse architecture</i>		7
2.2.1 Lighthouse to Node interactions		8
2.2.2 User to Lighthouse interactions		8
2.2.3 Node organization and filtering		9
3. Lighthouse VM installation	10	
3.1 <i>Lighthouse VM components</i>		10
3.2 <i>VMware vSphere 6.0 via the VMware vSphere 6.0 client on Windows</i>		10
3.2.1 Launch the vSphere Client and connect to a vSphere instance.		10
3.2.2 Import the Lighthouse VM Open Volume Format (.ovf) image		11
3.2.3 Launch the Opengear Lighthouse virtual machine		13
3.2.4 Access the console of a running but headless Opengear Lighthouse instance		13
3.3 <i>VMware Workstation Player on Windows as host</i>		14
3.4 <i>VMware Workstation Pro on Windows as host</i>		15
3.5 <i>VMware Workstation Player or Pro on Fedora Workstation as host</i>		15
3.6 <i>Local deployment on Hyper-V running on Windows 10/Windows Server 2016</i>		15
3.7 <i>Remote Hyper-V deployment with pre-authenticated user</i>		16
3.8 <i>Remote Hyper-V deployment with different user</i>		16
3.9 <i>VirtualBox on Windows as host</i>		17
3.10 <i>VirtualBox on macOS as host</i>		18
3.11 <i>VirtualBox on Ubuntu as host</i>		19
3.12 <i>VirtualBox on Fedora Workstation as host</i>		20
3.13 <i>Virtual Machine Manager (KVM) on Ubuntu as host</i>		20
3.14 <i>Boxes on Fedora Workstation as host</i>		21
3.15 <i>Boxes on CentOS as host</i>		21
3.16 <i>Google Compute Engine environment</i>		22
4. First boot of the Lighthouse VM	23	
5. Initial system configuration	25	
5.1 <i>Lighthouse IP addressing</i>		25
5.2 <i>Loading Lighthouse</i>		25
5.3 <i>Login to Lighthouse</i>		25
5.4 <i>Network connections</i>		26
5.5 <i>Setting the Lighthouse hostname</i>		27

5.6 Adding external IP addresses manually (optional)	28
5.7 Setting the Lighthouse internal clock	29
5.8 Examine or modify the Lighthouse SSL certificate	31
5.9 Examine or modify Lighthouse Session Settings	32
5.10 Examine or change the MTU of the Lighthouse VPN tunnel	33
5.11 Enable or modify SNMP Service	33
5.12 Lighthouse MIBs	34
6. Shut down or restart Lighthouse	40
6.1 Shut down a running Lighthouse instance	40
6.2 Restarting a running Lighthouse instance	40
7. Using Lighthouse	41
7.1 Licensing third-party nodes before enrollment	41
7.1.1 Adding a license using the Lighthouse UI	41
7.1.2 Showing installed licenses in the Lighthouse UI	42
7.1.3 Showing installed licenses via the Local Terminal	43
7.2 Enrolling nodes	44
7.2.1 Enrollment overview	44
7.2.2 Enrollment bundles	44
7.2.3 Creating an enrollment bundle	45
7.2.4 Structure of an enrollment bundle	47
7.2.5 Enrollment via Lighthouse Web UI	48
7.2.6 Enrollment via Node Web UI	51
7.2.7 Lighthouse Enrollment via OM2200 Web UI	51
7.2.8 Mass Enrollment using ZTP	51
7.2.9 Enrollment via USB drive	52
7.3 The Enrolled Nodes page	53
7.4 Filtering pages displaying nodes	54
7.4.1 Filtering using the Free Text Search field	54
7.4.2 Filtering using the Smart Group Filtering drop-down menu	54
7.4.3 Filtering using the Managed Device Filtering drop-down menu	55
7.5 Creating Smart Groups	56
7.6 Editing an existing Smart Group	56
7.7 Creating Managed Device Filters	57
7.8 Editing an existing Managed Device Filter	58
7.9 Connecting to a node's web-management interface	59
7.10 Connecting to a node's serial ports via Console Gateway	60
7.10.1 Access via HTML5 Web Terminal	60
7.10.2 Access via SSH	61
7.10.3 Example Console Gateway session	62
8. Lighthouse user management	63
8.1 Password fields in Lighthouse	63

8.2	<i>Creating new groups</i>	63
8.3	<i>Modifying existing groups</i>	64
8.4	<i>A note on default netgrp Lighthouse group</i>	65
8.5	<i>Creating new users</i>	65
8.6	<i>Modifying existing users</i>	67
8.7	<i>Deleting users</i>	68
8.8	<i>Disabling a Lighthouse root user</i>	68
8.9	<i>Configuring AAA</i>	68
8.9.1	<i>LDAP Configuration</i>	69
8.9.2	<i>RADIUS configuration</i>	69
8.9.3	<i>TACACS+ configuration</i>	70
9.	Lighthouse central configuration	72
9.1	<i>Creating new users and groups templates</i>	72
9.2	<i>Modifying existing users and groups templates</i>	74
9.3	<i>Deleting users or groups from a template</i>	76
9.4	<i>Deleting users and groups templates</i>	76
9.5	<i>Creating new authentication templates</i>	76
9.6	<i>Modifying existing authentication templates</i>	77
9.7	<i>Deleting authentication templates</i>	79
9.8	<i>Creating new script templates</i>	79
9.9	<i>Modifying existing script templates</i>	80
9.10	<i>Deleting script templates</i>	81
9.11	<i>Apply Templates</i>	82
9.12	<i>Manually Activate Secure Provisioning via Template</i>	84
10.	NetOps Automation	85
10.1	<i>Secure Provisioning for NetOps Automation</i>	85
10.2	<i>Initial Setup</i>	85
10.2.1	<i>Manually install NetOps virtual disk</i>	86
10.2.2	<i>Connect target device</i>	88
10.3	<i>Device Provisioning configuration</i>	88
10.3.1	<i>Device Resource Bundle</i>	88
10.3.2	<i>Node Inventory</i>	89
10.3.3	<i>Create device configuration</i>	89
10.3.4	<i>Using templated resources</i>	90
10.4	<i>UI-based workflow</i>	91
10.4.1	<i>Create Device Resource Bundle</i>	91
10.4.2	<i>Define Resource Distribution</i>	92
10.4.3	<i>Push Resources</i>	94
10.4.4	<i>Manage Device Provisioning on Each Interface</i>	95
10.5	<i>CLI-based workflow</i>	95

5 Lighthouse 5 User Guide

10.5.1 Create configuration YAML	95
10.5.2 Upload configuration and resources	98
10.5.3 Direct git repository access	99
10.5.4 Direct DHCP configuration	99
10.6 <i>NetOps Module management</i>	100
11. Command line tools	101
11.1 <i>node-info</i>	102
11.2 <i>node-upgrade</i>	103
11.3 <i>ogadduser</i>	104
11.4 <i>ogconfig-cli</i>	104
11.4.1 Commands to try from within the <i>ogconfig-cli</i> tool	105
11.4.2 Config searches using <i>ogconfig-cli</i>	105
11.4.3 Changing a configuration from within <i>ogconfig-cli</i>	105
11.4.4 Configuration validation from within <i>ogconfig-cli</i>	106
11.4.5 Modify LHVPN keepalive timeout for different sized deployments with <i>ogconfig-cli</i>	106
11.4.6 Support for mounting the hard disks with <i>ogconfig-cli</i>	106
11.5 <i>oglicdump</i>	107
11.6 <i>cron</i>	107
11.7 <i>sysflash</i>	108
11.8 <i>Selecting nodes using shell-based tools</i>	109
11.8.1 Select all nodes	109
11.8.2 Running commands on selected nodes	109
12. System upgrades	110
12.1 <i>Upgrading the system from within Lighthouse</i>	110
12.2 <i>Upgrading the Lighthouse system via the Local Terminal</i>	111
13. Troubleshooting	112
13.1 <i>Finding the current Lighthouse instance version</i>	112
13.1.1 Using the web UI	112
13.1.2 Via the local Lighthouse shell	112
13.1.3 Other information sources related to a Lighthouse instance's version	113
13.2 <i>Technical support reports</i>	113
13.2.1 Generate a support report via the Lighthouse interface	113
13.2.2 Generate a support report via the local terminal	114
13.3 <i>Returning a Lighthouse instance to factory settings</i>	115
14. Technical support	117
15. End-user license agreements	118
15.1 <i>Opengear end-user license agreement</i>	118
15.2 <i>GNU general public license (GPL), version 2</i>	120
16. Standard Warranty	124

1. About this User Guide

This manual covers Lighthouse 5 and is current as of 5.2.2. When using a minor release (5.2.x), there may or may not be a specific version of the user guide for that release. The current Lighthouse 5 user guide can always be found [here](#).

NOTE: OM2200 support is partial for this release. Mass node enrollment using ZTP, enrollment via USB drive, **Access Web UI** functionality, and SNMP information are not currently supported for OM2200 nodes. Script templates are supported.

Terms used in this guide to define Lighthouse elements and concepts are listed below.

Term	Definition
Enrollment	Connecting a node to Lighthouse
Enrollment Bundle	Used to assign a number of tags to a set of nodes when they are enrolled. During enrollment, the bundle is specified using its name, and a bundle-specific enrollment token.
Enrolled Node	Node that has been connected to Lighthouse and is ready for use.
Enrollment Token	A password that authorizes the node with Lighthouse. Used when performing Node-based, or ZTP enrollment.
Lighthouse	System for accessing, managing and monitoring Opengear console servers.
Lighthouse VPN	The OpenVPN based connections that the Lighthouse instance has with the nodes it is managing
Managed Device	A device that is managed via a node through a serial, USB, or network connection.
Node	A device that can be enrolled with Lighthouse, allowing it to be accessed, managed, and monitored. Currently, Opengear console servers are supported on a standard license, with support for other vendors Console Servers available as an add-on.
Pending Node	A node that has been connected to Lighthouse and has been configured with a VPN Tunnel, but which has not yet been approved for access, monitoring, or management. The approval operation can be automated by configuring Lighthouse to auto- approve nodes.
Role	A set of access rights for a particular group. Three roles are defined within Lighthouse: Lighthouse Administrator, Node Administrator, and Node User.
Smart Group	Dynamic filter used to search for particular nodes, or for defining the access rights of a group of users. Smart Groups use node properties, as well as tags defined by users.
Tag	User-defined attribute and value that is assigned to one or more nodes. Tags are used when creating Smart Groups for filtering views or access to nodes.

2. Lighthouse overview

2.1 Lighthouse VM 5 host requirements

- Lighthouse deploys as an application running in a Linux-based virtual machine (VM). The Lighthouse binary is available in open (for VM managers such as Boxes, KVM, and VirtualBox), VMware and Hyper-V specific Virtual Machine formats, and Google Compute Engine (GCE) image format.
- To run a Lighthouse VM, the host computer must be able to run a VM manager and at least one full 64-bit Linux-based virtual machine.
- To host Lighthouse, the VM needs to be configured to support:
 - 10GB SCSI disk.
 - 1 x network interface card, preferably paravirtualised (virtio, vmxnet3), Realtek rtl8139, or Intel e1000 are also supported, bridged.
 - VGA console for initial setup.

To dimension CPU and RAM resources, follow these guidelines:

CPU and RAM utilization increase with the number of enrolled nodes.

For small deployments (less than 100 nodes), allocate:

- 2 x 64-bit CPU cores.
- 4GB RAM.

For medium deployments (between 100 and 600 nodes), allocate:

- 4 x 64-bit CPU cores.
- 8GB RAM.

For large deployments (between 600 and 1200 nodes), allocate:

- 4 x 64-bit CPU cores.
- 16GB RAM.

For very large deployments (more than 1200 nodes), allocate:

- 8 x 64-bit CPU cores.
- 32GB RAM.

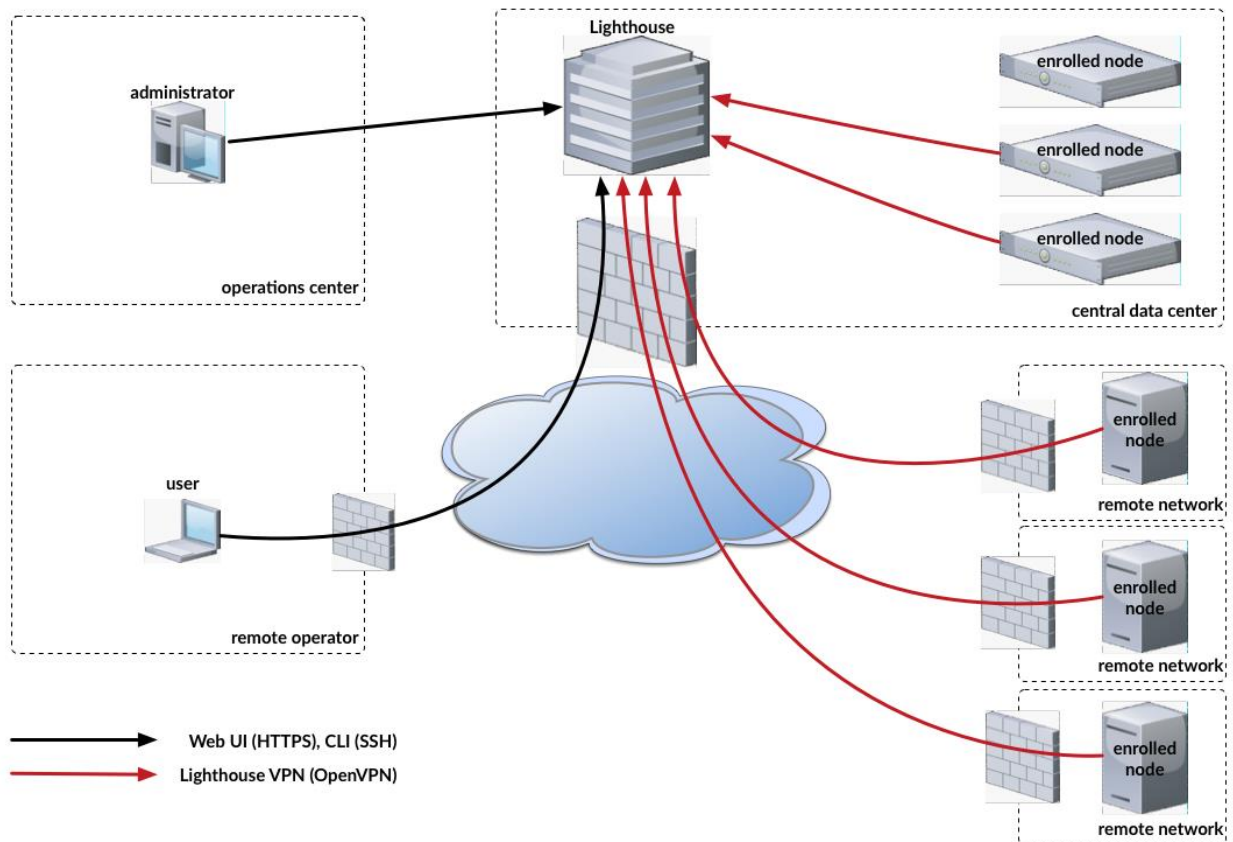
For large and very large deployments, contact us for guidance on the deployment options, including low and zero-touch enrollment. The performance and limitations are dependent on network deployment.

Also, Lighthouse VPN keepalive timeout needs to be modified according to the size of deployment. See section 10.5.5 for more information.

2.2 Lighthouse architecture

Lighthouse provides a platform for centrally accessing, managing, and monitoring OpenGear console servers.

Console servers connect to a central Lighthouse instance over an OpenVPN tunnel, and are accessed, managed, and monitored via services transported over the VPN tunnel. In Lighthouse terminology, the console server is referred to as the node.



2.2.1 Lighthouse to Node interactions

For management and monitoring operations, Lighthouse queries and pushes data to and from a REST API on the node.

When a node is enrolled in Lighthouse, Lighthouse generates an X.509 certificate. This certificate authenticates the OpenVPN tunnel and provides the node access to the Lighthouse REST API. The node also imports a Certificate Authority from Lighthouse and uses that to allow Lighthouse access to the node's REST API. Lighthouse also provides a public SSH key to the node, which allows Lighthouse to access the node's serial ports via SSH.

For serial access, a node's serial port subsystem is connected to via SSH. Users can also access the node's Web UI, which is reverse-proxied through the VPN tunnel.

2.2.2 User to Lighthouse interactions

Users interact with Lighthouse via an Ember.js JavaScript application, which communicates with Lighthouse via a REST API. This REST API can integrate Lighthouse into other systems. Documentation for this API is available for direct customer use.

While Lighthouse 5 supports REST API versions v1, v1.1, v2, and v3, some of the endpoints in v1, v1.1, and v2 have been deprecated, meaning the functionality and expected request body may be different. We advise using the v3 to ensure the latest available functionality.

2.2.3 Node organization and filtering

To help search, organize, and filter access to nodes, Lighthouse uses **Smart Groups** which allow node properties and user-supplied **tags**, consisting of a name and value, to be compiled into a search expression. These search expressions can be saved and used to filter the various lists of nodes in the Web UI, for example when selecting a serial port to connect to or to connect to the node's Web UI. They can also be used for selecting the nodes that a particular group of users can access.

To help locate managed devices, Lighthouse includes **Managed Device Filtering** which allows users to search for port labels on a node. This search can be saved and applied on the **MANAGE > Managed Devices > Console Gateway** page.

3. Lighthouse VM installation

3.1 Lighthouse VM components

Lighthouse VM is available in several formats:

- An Open Volume Format file — `lighthouse-5.2.2-ovf.zip` — inside a PKZip archive. This is for use with virtual machine managers such as KVM and Virtual Box.
- A VMware configuration file — `lighthouse-5.2.2-vmx.zip` — inside a PKZip archive. This is for use with virtual machine managers from VMware.
- A raw (.hdd) file, `lighthouse-5.2.2-raw.hdd.tar`. This file has been compressed with `tar` and is for use with hosting services such as ElasticHosts.
- An Open Virtual Appliance file — `lighthouse-5.2.2.ova`. This is for use with virtual machine managers such as VM and Virtual Box as well as for use with virtual machine managers from VMware.
- A Hyper-V configuration file with Powershell script — `lighthouse-5.2.2-hyperv.zip` — inside a PKZip archive. This is for use in Microsoft Hyper-V deployment.
- A Google Compute Engine (GCE) image — `lighthouse-5.2.2-gce.tar.gz` — inside a GNU archive. This is for use in Google Compute Engine environment.
- An upgrade file for GCE image, `lighthouse-5.2.2-gce.lh_upg`.
- An upgrade file, `lighthouse-5.2.2.lh_upg`.

3.2 VMware vSphere 6.0 via the VMware vSphere 6.0 client on Windows

This procedure assumes VMware vSphere 6.0 is installed and running on available hardware. User must have access to a Windows computer on which the VMware vSphere 6.0 client is installed and that this installed client application can connect to and manage the VMware Sphere 6.0 instance. Finally, a copy of the Lighthouse binary in Open Volume Format is required, the `.ovf` file, either copied to the Windows computer running the VMware vSphere 6.0 client or available via a URL.

This procedure was tested using the VMware Sphere Client 6.0 running on Windows 7 Enterprise SP 1.

3.2.1 Launch the vSphere Client and connect to a vSphere instance.

1. Launch the VMware vSphere Client. The simplest way is to use the **Start Menu** shortcut added during installation.

Start > All Programs > VMware > VMware vSphere Client

The VMware vSphere Client opens a login window.



2. Select the IP address or name of the VMware vSphere instance where Lighthouse will be installed from the **IP address/Name** drop-down list.
3. Enter the **User name** and **Password** required to gain management privileges to the selected VMware vSphere instance.
4. Click **Login** or press **Return**.

The login window displays progress text in the bottom left corner:

Connecting
Loading inventory
Loading main form
Displaying main form

The **vSphere main form** window opens.

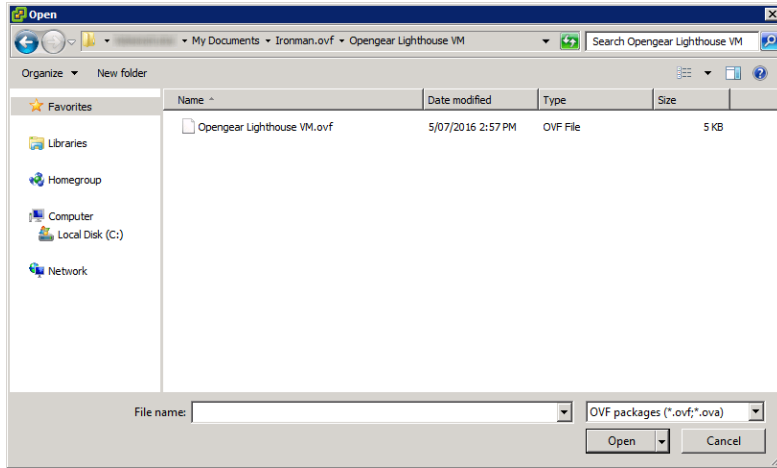
3.2.2 Import the Lighthouse VM Open Volume Format (.ovf) image

1. From the vSphere Client menu bar, choose **File > Deploy OVF Template**.
The **Deploy OVF Template** window appears, with the first stage, **Source**, pre-selected.
2. If the file `Opengear Lighthouse VM.ovf` is on a remote computer via a URL, enter this URL in the **Deploy from a file or URL** field. Otherwise, click **Browse**. An **Open** dialog appears.

Navigate to the directory containing the file `Opengear Lighthouse VM.ovf`.

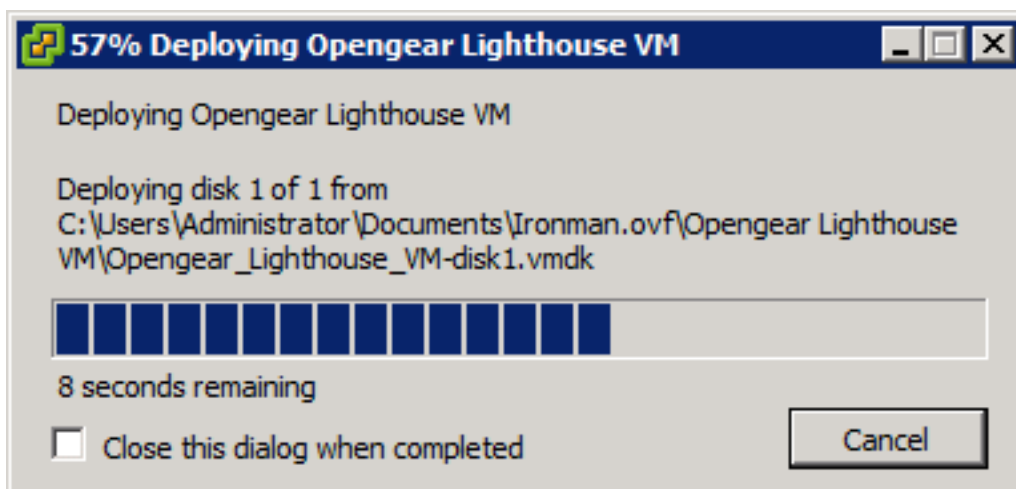
In the following screenshot, the file is located at

`C:\Users\%USERNAME%\My Documents\Ironman.ovf\Opengear Lighthouse VM\.`

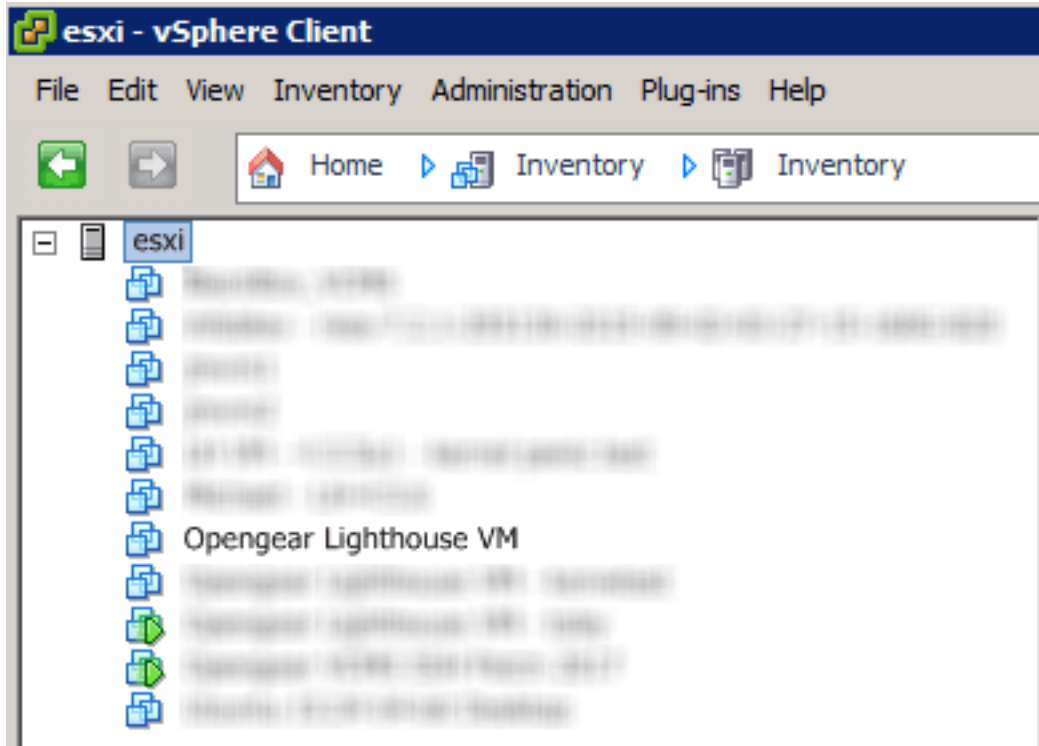


Select `Opengear Lighthouse VM.ovf` and click **Open**.

3. The **Deploy OVF Template** window opens again, with the `Opengear Lighthouse VM.ovf` file listed in the **Deploy from a file or URL** combo-box. Click **Next**.
4. The **OVF Template Details** stage appears, showing basic information about the Lighthouse VM encapsulated by the `.ovf` file. Click **Next**.
5. The **Name and Location** screen appears with the **Name** field pre-populated and pre-selected. The default name is **Opengear Lighthouse VM**. To change this, enter a new name. Click **Next**.
6. The **Disk Format** screen displays which data-store the Lighthouse VM's virtual disk uses, how much free space the virtual disk has available and which provisioning scheme is being used. Click **Next**.
7. The **Network Mapping** screen shows which destination or inventory network the Lighthouse VM's virtual network is mapped to. Click **Next**.
8. The **Ready to Complete** screen appears, listing the basic properties of the about-to-be-deployed virtual machine. To be able to power-up the new virtual machine after deployment, select the **Power on after deployment** checkbox. Click **Finish**.
9. The **Deploying Opengear Lighthouse VM** progress dialog appears.



10. Once deployment has finished the **Deployment Completed Successfully** alert appears. Click **Close**. The new virtual machine is now deployed and appears in the inventory list.



3.2.3 Launch the Opengear Lighthouse virtual machine

The vSphere Client provides several ways of launching a Virtual Machine hosted on a vSphere instance. Begin by selecting the Opengear Lighthouse VM from the vSphere Client's inventory list. The selected VM can then be launched by doing one of the following:

- Select **Inventory > Virtual Machine > Power > Power On**.
- Press **Ctrl-B**.
- Click the **Power** on the virtual machine link in the **Basic Tasks** section of the **Getting Started** tab. This option requires the **Getting Started** tab be front-most. If it is not already the front-most tab, make it active by clicking it.
- Select **Inventory > Virtual Machine > Open Console** and then:
 - Click **Power On** in the console tool bar, or
 - Choose **VM > Power > Power On** from the console menu bar, or
 - Press **Ctrl-B**.

NOTE: Only the fourth option above results in the running virtual machine being accessible from within the vSphere Client. The first three boot the Lighthouse VM and get it running headless.

3.2.4 Access the console of a running but headless Opengear Lighthouse instance

If direct interaction with a running but headless *Opengear Lighthouse VM* is required, open a console window.

Select the running Opengear Lighthouse VM in the vSphere Client's inventory list, then do one of the following:

- Select **Inventory > Virtual Machine > Open Console** or
- Right-click and select **Open Console** from the contextual menu that appears.

NOTE: A Lighthouse VM is running a bash shell with no other interactive options. As a result, when the vSphere Client opens its console window, the Lighthouse VM captures the mouse pointer, making it unavailable for use by any other window. Press **CTRL+ALT** to release the pointer.

3.3 VMware Workstation Player on Windows as host

Follow these steps when VMware Workstation Player is installed on the host Windows machine. VMware-ready virtual machine files are stored in `C:\Users\%USERNAME%\Virtual Machines\`. This is the location selected by default by VMware Workstation Player. If another location is preferred, adjust this procedure as required.

Prepare the Lighthouse VM file for import into VMware Workstation Player.

1. Move the `lighthouse-5.2.2-vmx.zip` archive to `C:\Users\%USERNAME%\Virtual Machines\`.
2. Right-click the archive and select **Extract all** from the contextual menu.
3. A **Select a Destination and Extract Files** dialog opens. By default, the location is the same folder as the archive is in: `C:\Users\%USERNAME%\Virtual Machines\`. Leave this as the destination folder.
4. Uncheck the **Show extracted files when complete** checkbox and then click **Extract**.
5. A folder called `ironman` is created inside `C:\Users\%USERNAME%\Virtual Machines\`.

Import the Opengear Lighthouse VM file into VMware Workstation Player.

1. Launch VMware Workstation Player.
2. Click **Open a Virtual Machine**.
3. Navigate to `C:\Users\%USERNAME%\Virtual Machines\ironman\`.

VMware Workstation Player points to *Libraries > Documents* and includes `C:\Users\%USERNAME%\My Documents\`.

Assuming this is the case, double-click `Virtual Machines` and then double-click `Ironman`.

4. If only one file — `Ironman` — is visible, double-click it to add the Lighthouse 5.2.2 virtual machine to the VMware Workstation 12 Player virtual machines list. If more than one file appears, double-click `Ironman.vmx`.
5. The Lighthouse virtual machine is added to the VMware Workstation 12 Player virtual machines list.
6. With **Opengear Lighthouse VM** selected in the VMware Workstation 12 Player virtual machine list, click **Play virtual machine** to boot Lighthouse.

3.4 VMware Workstation Pro on Windows as host

This procedure assumes VMware Workstation Pro is already installed on the host Windows machine and that VMware-ready virtual machine files are stored in `C:\Users\%USERNAME%\Virtual Machines\`. If another location is preferred, adjust the steps as needed.

Prepare the Opengear Lighthouse VM file for import into VMware Workstation Pro.

1. Move the `lighthouse-5.2.2-vmx.zip` archive to `C:\Users\%USERNAME%\Virtual Machines\`.
2. Right-click the `lighthouse-5.2.2-vmx.zip` archive and select **Extract all** from the contextual menu.
3. A **Select a Destination and Extract Files** dialog opens. The location is the same folder as the PKZip archive is in: `C:\Users\%USERNAME%\Virtual Machines\`. Leave this as the destination folder.
4. Uncheck the **Show extracted files when complete** checkbox and then click **Extract**.
5. A folder called **ironman** is created inside `C:\Users\%USERNAME%\Virtual Machines\`.

Import the Opengear Lighthouse VM file into VMware Workstation Pro.

1. Click **Open a Virtual Machine**.
2. Navigate to `C:\Users\%USERNAME%\Virtual Machines\ironman\`.
3. VMware Workstation Pro points to `Libraries > Documents` and this library includes `C:\Users\%USERNAME%\My Documents\`. Double-click `Virtual Machines` and then double-click `Ironman`.
4. If only one file — `Ironman` — appears, double-click it to add the Lighthouse 5.2.2 virtual machine to the VMware Workstation Pro virtual machines list. If more than one file appears, double-click `Ironman.vmx`.
5. The Lighthouse 5.2.2 virtual machine is added to the VMware Workstation Pro virtual machines list.
6. With the **Opengear Lighthouse VM** selected in the **My Computer** listing and the subsequent **Opengear Lighthouse VM** tab open, click **Power on this virtual machine** to boot Lighthouse.

3.5 VMware Workstation Player or Pro on Fedora Workstation as host

VMware Workstation Player 12 cannot be installed on Fedora 25 without substantial reconfiguration of a base Fedora Workstation setup and leaves Fedora Workstation in a state that is unsupported by any external entity.

Opengear does not support this particular combination of host operating system and virtual machine manager.

3.6 Local deployment on Hyper-V running on Windows 10/Windows Server 2016

This procedure assumes Hyper-V is already installed on a Windows 10/Windows Server 2016 host machine and the required Zip archive, `ironmam-hyperv.zip` is in `C:\Users\%USERNAME%\Downloads`.

1. Unzip `ironmam-hyperv.zip`.

2. Navigate to the extracted folder. Make sure `ironman.vhd` and `lighthouse_virtual_machine_registration.ps1` are in the folder.
3. Right-click and choose **Run with Powershell** to execute the Powershell script.
4. Leave the host name empty when prompted to deploy Lighthouse to local machine.
5. Launch Hyper-V Manager. Lighthouse should be registered as a new VM image under Virtual Machine.
6. Select **Lighthouse** from the list and click **Start** in the **Action Panel** to boot Opendgear Lighthouse.

3.7 Remote Hyper-V deployment with pre-authenticated user

In this scenario, the user who performs Lighthouse deployment does not have local access to Hyper-V installed on Windows 2016. However, user has access to a Windows 10 which can manage the Hyper-V server remotely.

This procedure assumes Hyper-V is installed on Windows Server 2016 host machine and the required Zip archive `ironman-hyperv.zip` is in `C:\Users\%USERNAME%\Downloads`. Windows 10 is already configured to manage Hyper-V on Windows Server 2016. **Windows 10 and Windows Server 2016 must have the same user (same password) created.** The user who performs the deployment must have permission to both execute the Powershell script and deploy the image on Hyper-V.

1. Login to Windows 10 with the user mentioned above.
2. Unzip `ironman-hyperv.zip`
3. Navigate to the extracted folder. Make sure `ironman.vhd` and `lighthouse_virtual_machine_registration.ps1` are in the folder.
4. Right-click and choose **Run with Powershell** to execute the Powershell script.
5. Enter the fully qualified domain name for Windows Server 2016 when prompted to deploy Lighthouse to the remotely-managed Windows Server 2016 machine.
6. Launch Hyper-V Manager. Lighthouse should be registered as a new VM image under Virtual Machine for Windows Server 2016.
7. Select **Lighthouse** from the list and click **Start** in the **Action Panel** to boot Opendgear Lighthouse.

3.8 Remote Hyper-V deployment with different user

In this scenario, the user who performs Lighthouse deployment does not have local access to Hyper-V installed on Windows Server 2016. However, user has access to Windows 10 which can manage the Hyper-V server remotely. The user who performs the deployment must have permission to both execute the Powershell script and deploy the image on Hyper-V. This procedure assumes Hyper-V is installed on Windows Server 2016 host machine and the required Zip archive, `ironmam-hyperv.zip`, is in `C:\Users\%USERNAME%\Downloads`. Windows 10 is already configured to manage Hyper-V on Windows Server 2016.

1. Login to windows 10 with a user who does not exist on Windows Server 2016.
2. Unzip `ironman-hyperv.zip`.
3. Navigate to the extracted folder. Make sure `ironman.vhd` and `lighthouse_virtual_machine_registration.ps1` are in the folder.
4. Right-click and choose **Run with Powershell** to execute the Powershell script.
5. Enter the fully qualified domain name for Windows Server 2016 when prompted to deploy Lighthouse to remotely -managed Windows Server 2016 machine.
6. Enter the user details created on Windows Server 2016 which has permission to deploy Hyper-V.
7. Launch Hyper-V Manager. Lighthouse should be registered as a new VM image under Virtual Machine for Windows Server 2016.
8. Select **Lighthouse** from the list and click **Start** in the **Action Panel** to boot Opendgear Lighthouse.

3.9 VirtualBox on Windows as host

NOTE: when a Skylake processor is available, we **do not** recommend the use of VirtualBox.

NOTE: We recommend that VirtualBox users customize their instances and change their network cards to one other than e1000. We also suggest virtio for better performance.

This procedure assumes VirtualBox is already installed on the host machine and the required PKZip archive, `lighthouse-5.2.2-ovf.zip` is in `C:\Users\%USERNAME%\Downloads`.

1. Unzip `ironman-ovf`. It may appear as `lighthouse-5.2.2-ovf.zip` depending on the Windows Explorer preference settings).
2. Right-click the `ironman-ovf` archive and select **Extract all** from the contextual menu.
3. The **Select a Destination and Extract Files** dialog opens. The destination is `C:\Users\%USERNAME%\Downloads\Ironman-ovf`.
4. Uncheck the **Show extracted files when complete** checkbox and edit the destination by removing `Ironman-ovf` from the path.
5. Click **Extract**.
6. A folder called `ironman-ovf` is created inside `C:\Users\%USERNAME%\Downloads\`.
7. Launch VirtualBox.
8. The **Oracle VM VirtualBox Manager** window appears.
9. Choose **File > Import Appliance**.
10. The **Appliance to import** dialog opens.
11. Click **Expert Mode**.
12. The **Appliance to import** dialog changes from **Guided Mode** to **Expert Mode**.
13. Click the icon of a folder with an upward pointing arrow superimposed. This icon is to the far right of the **Appliance to import** field.
14. The **Open File** dialog appears with `C:\Users\%USERNAME%\Documents` as the current folder.
15. Navigate to `C:\Users\%USERNAME%\Downloads\Ironman.ovf\Opengear Lighthouse VM\`.
16. Select the file `Opengear Lighthouse VM` and click **Open**.
17. Double-click the text `vm` in the **Name** row and **Configuration** column to make it editable.
18. Type **Opengear Lighthouse VM** and press **Enter**.
19. Click **Import**.
20. A new virtual machine, called **Opengear Lighthouse VM** is added to the list of virtual machines available to Virtual Box.
21. Select **Opengear Lighthouse VM** from the list.
22. Choose **Machine > Settings**. Or click the **Settings** icon in the **VirtualBox Manager** toolbar or press **Control+S**.
23. The **Opengear Lighthouse VM — Settings** dialog appears.
24. Click the **System** option in the list of options running down the left-hand side of the dialog.
25. The dialog shows the **System** options available as three tabs: **Motherboard**, **Processor**, and **Acceleration**. Depending on the underlying hardware platform, **Acceleration** may be greyed-out and unavailable. The **Motherboard** tab is preselected.
26. In the **Motherboard** tab, select the **Hardware Clock in UTC Time** checkbox.
27. Click **OK** or press **Return**.
28. Select **Opengear Lighthouse VM** from the list and click **Start** in the **Oracle VM VirtualBox Manager** toolbar to boot Lighthouse. Double-clicking **Opengear Lighthouse VM** in the list also boots Lighthouse.

NOTE: Selecting the **Hardware Clock in UTC Time** checkbox is necessary because Lighthouse expects the hardware clock to be set to UTC, not local time. Unlike other Virtual Machine Managers, Virtual Box both exposes this option as a user-adjustable setting and does not set it to UTC by default.

3.10 VirtualBox on macOS as host

VirtualBox should already be installed on the host macOS machine and the required PKZip archive, `lighthouse-5.2.2-ovf.zip` is in `~/Downloads`.

1. Unzip `lighthouse-5.2.2-ovf.zip`.

This creates a folder — `Ironman-ovf` — in `~/Downloads` that contains the following files and folders:

```
Ironman-ovf
├── Opengear Lighthouse VM
│   ├── Opengear Lighthouse VM.ovf
│   └── Opengear_Lighthouse_VM-disk1.vmdk
```

2. Launch Virtual Box.
The **Oracle VM VirtualBox Manager** window appears.
3. Choose **File > Import Appliance** or press `Command+I`.
4. The **Appliance to import** dialog sheet slides down from the **Oracle VM VirtualBox Manager** toolbar.
5. Click **Expert Mode**.
The **Appliance to import** dialog sheet changes from **Guided Mode** to **Expert Mode**.
6. Click the icon of a folder with an upward pointing arrow superimposed. This icon is to the far-right of the **Appliance to import** field.
7. The **Open File** dialog sheet slides down from the **Oracle VM VirtualBox Manager** toolbar. This sheet opens with `~/Documents` as the current folder.
8. Navigate to `~/Downloads/Ironman.ovf/Opengear Lighthouse VM/`.
9. Select `Opengear Lighthouse VM` and click **Open**. (Depending on the Finder Preferences settings, the file may present as `Opengear Lighthouse VM.ovf`.)
10. Double-click the text `vm` in the **Name** row and **Configuration** column to make it editable.
11. Type **Opengear Lighthouse VM** and hit Return.
12. Click **Import**.
A new virtual machine, called **Opengear Lighthouse VM** is added to the list of virtual machines.
13. Select **Opengear Lighthouse VM** from the list.
14. Choose **Machine > Settings**. Or click the **Settings** icon in the VirtualBox Manager toolbar. The **Opengear Lighthouse VM — Settings** dialog appears.
15. Click the **System** option in the dialog's toolbar.
16. The dialog shows the **System** options available as three tabs: **Motherboard**, **Processor**, and **Acceleration**. (Depending on the underlying hardware platform, **Acceleration** may be greyed-out and unavailable). The **Motherboard** tab is preselected.
17. In the **Motherboard** tab, select the **Hardware Clock in UTC Time** checkbox.
18. Click **OK** or press Return.
19. Select **Opengear Lighthouse VM** from the list and click **Start** in the **Oracle VM VirtualBox Manager** toolbar to boot Lighthouse. Double-clicking **Opengear Lighthouse VM** in the list also boots Lighthouse.

NOTE: Selecting the **Hardware Clock in UTC Time** checkbox is necessary because Lighthouse expects the hardware clock to be set to UTC, not local time. Unlike other Virtual Machine Managers, Virtual Box both exposes this option as a user-adjustable setting and does not set it to UTC by default.

NOTE: By default, VirtualBox stores virtual machines in `~/VirtualBox VMs`. If this is the first virtual machine setup by VirtualBox, it creates the `VirtualBox VMs` folder in the current user's home-directory and a folder — `Opengear Lighthouse VM` — inside the `VirtualBox VMs` folder. The `Opengear Lighthouse VM` folder contains the files and folders which make up Lighthouse when run under Virtual Box.

3.11 VirtualBox on Ubuntu as host

Before beginning, make certain that VirtualBox and all required support files are installed on the host machine and the PKZip archive, `lighthouse-5.2.2-ovf.zip` is in `~/Downloads`.

1. Unzip `lighthouse-5.2.2-ovf.zip`.

This creates a folder — `Ironman-ovf` — in `~/Downloads` that contains the following files and folders:

```
Ironman-ovf
├── Opengear Lighthouse VM
│   ├── Opengear Lighthouse VM.ovf
│   └── Opengear_Lighthouse_VM-disk1.vmdk
```

2. Launch Virtual Box.
3. The **Oracle VM VirtualBox Manager** window appears.
4. Choose **File > Import Appliance**.
5. The **Appliance to import** dialog opens.
6. Click **Expert Mode**.
7. The **Appliance to import** dialog changes from **Guided Mode** to **Expert Mode**.
8. Click the icon of a folder with an upward pointing arrow superimposed. This icon is to the far right of the **Appliance to import** field.
9. A file-navigation dialog, **Choose a virtual appliance to import**, opens with `~/Documents` as the current folder.
10. Navigate to `~/Downloads/Ironman.ovf/Opengear Lighthouse VM/`.
11. Select `Opengear Lighthouse VM.ovf` and click **Open**.
12. Double-click the text `vm` in the **Name** row and **Configuration** column to make it editable.
13. Type **Opengear Lighthouse VM** and hit Return.
14. Click **Import**.
15. A new virtual machine, called **Opengear Lighthouse VM** is added to the list of virtual machines available to Virtual Box.
16. Select **Opengear Lighthouse VM** from the list and click **Start** in the **Oracle VM VirtualBox Manager** toolbar to boot Lighthouse. Double-clicking **Opengear Lighthouse VM** in the list also boots Lighthouse.

NOTE: VirtualBox stores virtual machines in `~/VirtualBox VMs`. If this is the first virtual machine setup by VirtualBox it creates the `VirtualBox VMs` folder in the current user's home-directory and a

folder — `Opengear Lighthouse VM` — inside the `VirtualBox VMs` folder. Inside `Opengear Lighthouse VM` are the files and folders which make up Lighthouse when run under Virtual Box.

3.12 VirtualBox on Fedora Workstation as host

Before beginning, make certain that VirtualBox and all required support files are already installed on the host machine and the PKZip archive, `lighthouse-5.2.2-ovf.zip` is in `~/Downloads`.

1. Unzip `lighthouse-5.2.2-ovf.zip`. This creates a folder — `Ironman.ovf` — in `~/Downloads` that contains the following files and folders:

```
Ironman.ovf
├── Opengear Lighthouse VM
│   ├── Opengear Lighthouse VM.ovf
│   └── Opengear_Lighthouse_VM-disk1.vmdk
```

2. Launch Virtual Box.
The **Oracle VM VirtualBox Manager** window appears.
3. Choose **File > Import Appliance** or press Control-I.
The **Appliance to import** dialog opens.
4. Click **Expert Mode**.
The **Appliance to import** dialog changes from *Guided Mode* to *Expert Mode*.
5. Click the icon of a folder with an upward pointing arrow superimposed. This icon is to the far right of the **Appliance to import** field.
The **Open File** dialog opens with `~/Documents` as the current folder.
6. Navigate to `~/Downloads/Ironman.ovf/Opengear Lighthouse VM/`.
7. Select `Opengear Lighthouse VM` and click **Open**.
8. Double-click the text `vm` in the **Name** row and **Configuration** column to make it editable.
9. Type **Opengear Lighthouse VM** and hit Return.
10. Click **Import**.
A new virtual machine, called **Opengear Lighthouse VM** is added to the list of virtual machines available to Virtual Box.
11. Select **Opengear Lighthouse VM** from the list and click **Start** in the **Oracle VM VirtualBox Manager** toolbar to boot Lighthouse. Double-clicking **Opengear Lighthouse VM** in the list also boots Lighthouse.

NOTE: VirtualBox stores virtual machines in `~/VirtualBox VMs`. If this is the first virtual machine setup by VirtualBox, it creates the `VirtualBox VMs` folder in the current user's home-directory and a folder — `Opengear Lighthouse VM` — inside the `VirtualBox VMs` folder. Inside `Opengear Lighthouse VM` are the files and folders which make up Lighthouse when run under Virtual Box.

3.13 Virtual Machine Manager (KVM) on Ubuntu as host

Virtual Machine Manager and all required support files should be installed on the host machine and the `.tar` archive, `lighthouse-5.2.2-raw.hdd.tar` is in `~/Downloads`.

1. Expand `lighthouse-5.2.2-raw.hdd.tar`. This extracts `lighthouse-5.2.2-raw.hdd` in `~/Downloads`.
2. Launch **Virtual Machine Manager**.

3. Click **New** at the top left of the **Virtual Machine Manager** window (or choose **File > New Virtual Machine**). The **Source Selection** window opens.
4. Click **Select a file**. A **Select a device or ISO file** dialog slides into view.
5. Navigate to `~/Downloads/`.
6. Select the file `lighthouse-5.2.2-raw.hdd` and click **Open** in the top right-hand corner of the dialog. A **Review** window opens providing basic information about the virtual machine or box, as Boxes calls them, to be created.
7. Click **Create** in the top right corner of the **Review** window.
8. A new virtual machine instance, **Opengear_Lighthouse_VM-disk1**, is created and presented in the **Boxes** window.
9. To rename the virtual machine instance, right-click on the machine instance and choose **Properties** from the contextual menu that appears. Click anywhere in the **Name** field to select and edit the name. Click the close box to save the changes.

3.14 Boxes on Fedora Workstation as host

Boxes and all required support files should be installed on the host machine and `lighthouse-5.2.2-ovf.zip` is in `~/Downloads`.

1. Unzip `lighthouse-5.2.2-ovf.zip`. This creates a folder — `Ironman.ovf` — in `~/Downloads` that contains the following files and folders:

```
Ironman.ovf
├── Opengear Lighthouse VM
│   ├── Opengear_Lighthouse_VM.ovf
│   └── Opengear_Lighthouse_VM-disk1.vmdk
```

2. Launch **Boxes**.
3. Click **New** in the **Boxes** window title bar. The **Source Selection** window opens.
4. Click **Select a file**. A **Select a device or ISO file** dialog opens.
5. Navigate to `~/Downloads/Ironman.ovf/Opengear Lighthouse VM/`.
6. Select the file `Opengear_Lighthouse_VM-disk1.vmdk` and click **Open** in the top right-hand corner of the dialog. A **Review** window opens providing basic information about the virtual machine (or 'box', as Boxes calls them) to be created.
7. Click **Create** in the top right corner of the **Review** window.
8. A new virtual machine instance, **Opengear_Lighthouse_VM-disk1** is created and presented in the **Boxes** window.
9. To rename the virtual machine instance, right-click on the machine instance and choose **Properties** from the contextual menu that appears. Click anywhere in the **Name** field to select and edit the name. Click **Close** to save the changes.

3.15 Boxes on CentOS as host

CentOS should be installed, complete with the Gnome desktop environment as the host operating system. CentOS includes the full complement of KVM-centric virtualization tools including the GUI-based virtualization management tools **Boxes** and **virt-manager** and the shell-based virtualization management tool **virsh**.

This procedure assumes **Boxes** is used to setup and manage the Lighthouse VM and that the required PKZip archive, `lighthouse-5.2.2-ovf.zip` is in `~/Downloads`.

1. Unzip `lighthouse-5.2.2-ovf.zip`.

This creates a folder — `Ironman.ovf` — in `~/Downloads` that contains the following files and folders:

```
Ironman.ovf
├── Opengear Lighthouse VM
│   ├── Opengear Lighthouse VM.ovf
│   └── Opengear_Lighthouse_VM-disk1.vmdk
```

2. Launch Boxes
3. Click **New** in the Boxes title bar.
4. Navigate to `~/Downloads/Ironman.ovf/Opengear Lighthouse VM/`
5. Select **Opengear Lighthouse VM** and click **Open**. A new virtual machine, called **Opengear LighthouseVM** is added to the list of virtual machines available to Boxes.

3.16 Google Compute Engine environment

These steps are for setting up Google Compute Engine (GCE) images for automating the initial boot process using the instance metadata. The root password, hostname, and external IP address are pulled from GCE and configured on the image. Any SSH keys provided by GCE will be added to the root user's `authorized_keys` file.

To set the root password automatically, it is required to define the custom metadata **og-initial-root-pass** for the instance. If this field is not defined, then the user will be prompted to enter a password over the GCE serial console.

The hostname and external IP metadata are automatically generated but if for any reason the image cannot obtain this metadata, defaults will be used (**lighthouse** for hostname, and external IP unset).

1. Import an image on Google Cloud following the instructions at https://cloud.google.com/compute/docs/images/import-existing-image#import_image

Configure an instance of the image including:

- Add a metadata field with the key **og-initial-root-pass** and a root password.
- Add an SSH public key to test trusted key authentication.
- Enable HTTPS checkbox to access Web UI.

2. Start up the image.

4. First boot of the Lighthouse VM

During boot, two screens open.

1. The first notes the VM is **Booting to latest installed image**.

The selected image is *Lighthouse Root 1*. Two other images are available: *Lighthouse Root 1* and *Root 2*. Do not change the boot image the VM boots from.

2. The second screen prompts to **Select Lighthouse boot mode** and displays four options:

- Graphics console boot
- Graphics console recovery mode
- Serial console boot
- Serial console recovery mode

3. **Graphics console boot** is pre-selected and should not be changed. After the first boot has completed a message appears:

```
Welcome to Lighthouse. This is software version:  
5.2.2
```

4. The final procedure in the initial setup appears:

```
To complete initial setup, please set a new root password.  
Press ENTER to continue.
```

5. After pressing **Enter**, a prompt appears:

```
Enter new root password:
```

6. Enter a password and press **Enter**. Keep in mind that non-US-English keyboards are not supported in the graphics console.

NOTE: We recommend you set a temporary password at this point and change it to a very strong high-entropy password as soon as possible using the WebUI.

7. The confirm prompt appears:

```
Confirm given password
```

8. Re-enter the password and press **Enter**. Multiple configuration notices appear ending with a login prompt:

```
lighthouse login:
```

9. Enter `root` and press **Enter**. A password prompt appears:

Password:

1. Enter the newly-set password and press **Enter**. A standard **bash** shell prompt appears with the list of static and DHCP addresses.

```
net1          192.168.0.1/24
```

```
net1:dhcp    [DHCP-supplied address]
```

```
root@lighthouse:~#
```


5. Initial system configuration

5.1 Lighthouse IP addressing

When the Lighthouse VM is booted and running, it can be reached at:

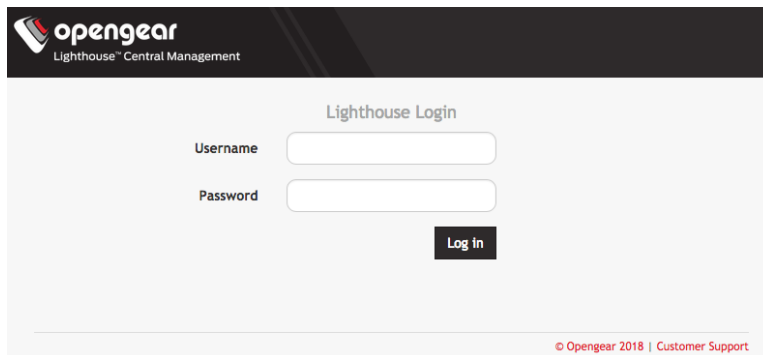
- The static address, 192.168.0.1, or
- The address it is assigned by any DHCP server it finds. Type **ifconfig** command to see which IP address the VM has been allocated by DHCP.
- Static IP address on another subnet, requiring IP address, mask, gateway set using `ogconfig-cli` commands.

Only the first two options are available out-of-the-box. The static IP on another subnet has to be configured first.

5.2 Loading Lighthouse

Open a new browser window or tab and enter:

1. `https://192.168.0.1/` or `https://[DHCP-supplied address]/` in the address bar
2. Press **Return**. The Lighthouse Login page loads.



The screenshot shows the Lighthouse Login interface. At the top left is the OpenGear logo with the text 'Lighthouse Central Management'. The main heading is 'Lighthouse Login'. Below this are two input fields: 'Username' and 'Password'. A 'Log in' button is positioned below the password field. At the bottom right of the page, there is a small red copyright notice: '© OpenGear 2018 | Customer Support'.

5.3 Login to Lighthouse

To login to Lighthouse:

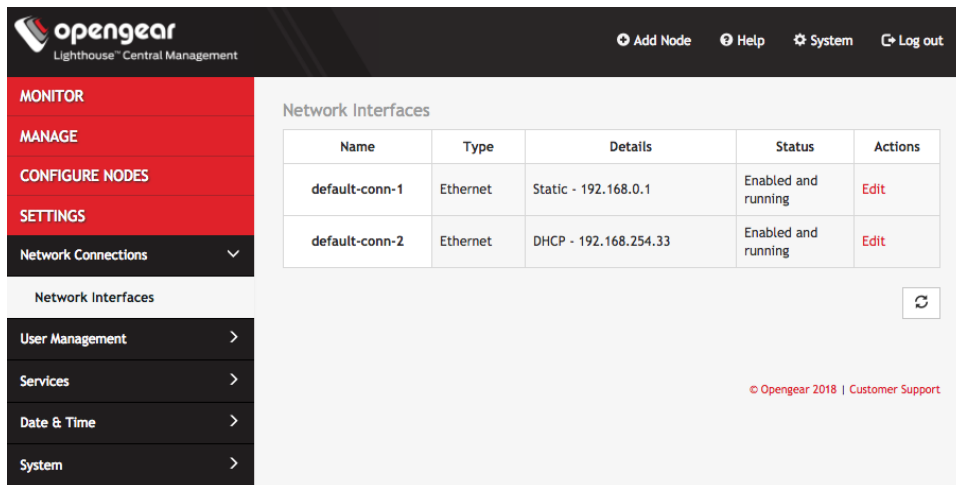
1. Enter a **username** in the **Username** field.
2. Enter the password in the **Password** field.
3. Click **Log In** or press **Enter**. The **Dashboard** loads.
4. Click **System** right top icon to see Current user.

The elements that appear on the **Dashboard** page depend on the privileges granted to the currently logged in user.

NOTE: The appearance of the Dashboard, the Sidebar, and other Lighthouse pages depends on the privileges assigned to the logged-in user. In this guide, screenshots represent what the **root** user sees. Users with different privileges will see filtered views of available nodes, managed devices, users, groups, tags, and Smart Groups have different privileges regards creating and changing settings within Lighthouse.

5.4 Network connections

To see the network connections available to Lighthouse, select **SETTINGS > Network Connections > Network Interfaces**



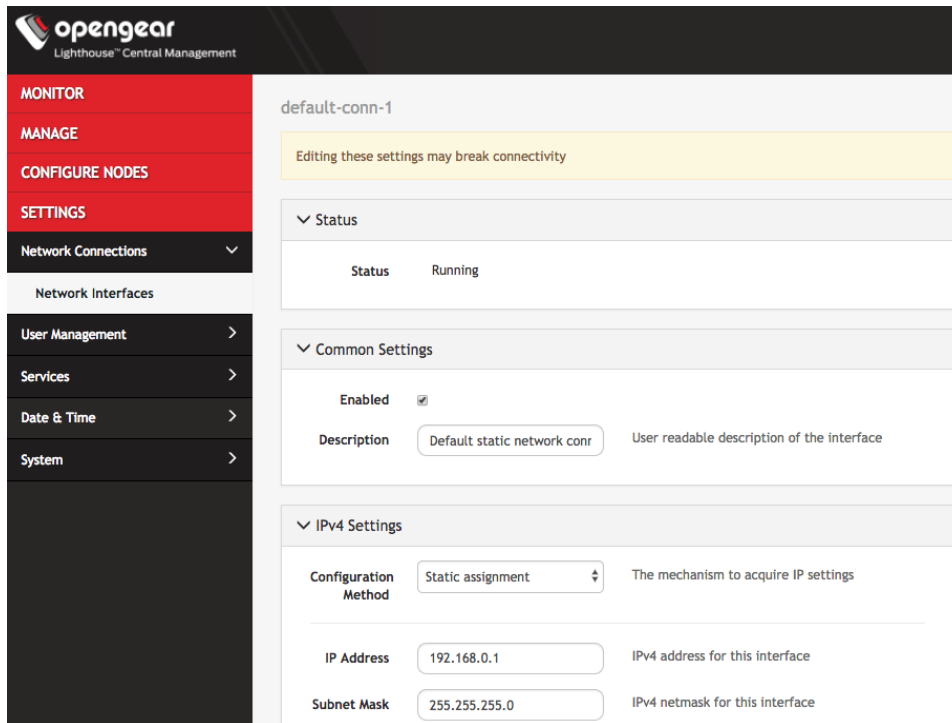
This displays two connections: static and DHCP interfaces.

Log in to the Lighthouse VM and run **ifconfig**. The two connections listed correspond to the following returned interfaces:

- *default-static* is `net1`
- *default-DHCP* is `net1:dhcp`

To edit a given network interface:

1. Select **SETTINGS > Network Connections > Network Interfaces**
2. Click **Edit** in the **Actions** section of the network interface to be modified.
3. Make the desired changes.
4. Click **Apply**.

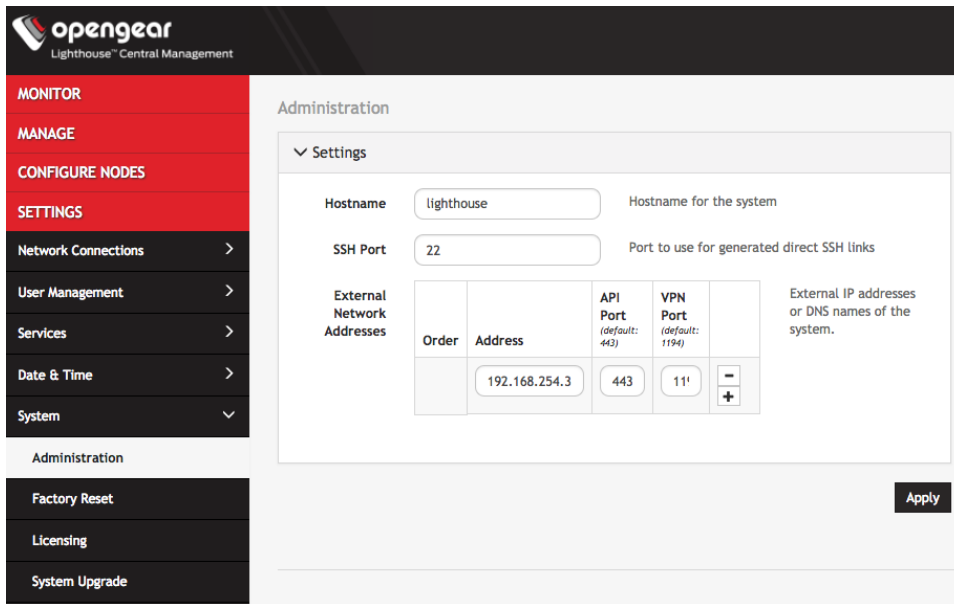


NOTE: Don't change the configuration method. Instead, disable the interface which is not planned to be used by unchecking the **Enabled** checkbox. If **default-static** and **default-DHCP** are changed to the same configuration method (i.e. both are set to **Static assignment** or both are set to **DHCP**) neither interface works.

5.5 Setting the Lighthouse hostname

To set the hostname for a running Lighthouse instance:

1. Select **SETTINGS > System > Administration**.
2. Edit the **Hostname** field as required.



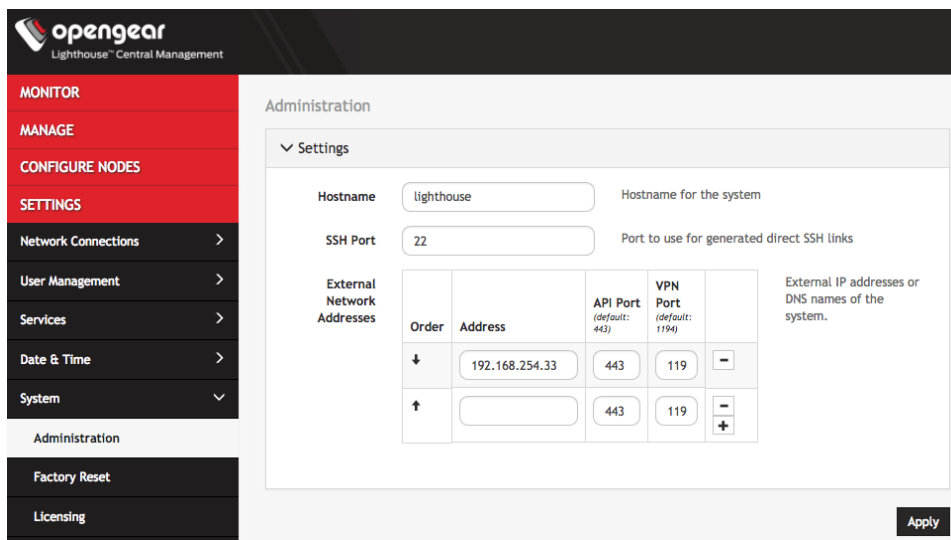
3. Click **Apply**.

5.6 Adding external IP addresses manually (optional)

Adding a Lighthouse instance's external IP address or addresses to a Lighthouse instance's configuration is an optional step.

To add a single external address:

1. Select **SETTINGS > System > Administration**.



2. In the **Address** field of the **External Network Addresses** section, enter an IP address.

3. (Optional step) Change the API Port, VPN Port or both, if the ports used on the entered IP address are different from the default (443 and 1194, respectively).
4. Click **Apply**.

To add further external addresses to a Lighthouse instance's configuration:

1. Click the **+** button. A second row appears in the **External Network Addresses** section.
2. In the **Address** field, enter an IP address.
3. (Optional step) Change the API Port, VPN Port or both, if the ports used on the entered IP address are different from the default (443 and 1194, respectively).
4. Add further IP addresses as required by repeating the steps above.
5. Click **Apply**.

To change the order in which manually-added IP addresses are sent to remote nodes:

1. Click the up and down arrows in the **Order** column to change the order in which the IP addresses are listed.
The presented order reflects the order in which these addresses are sent out.
2. Click **Apply**.

If external IP addresses are manually added to a Lighthouse configuration, these addresses are sent to a remote node during enrollment. If no external IP address is manually added, default external IP addresses are used.

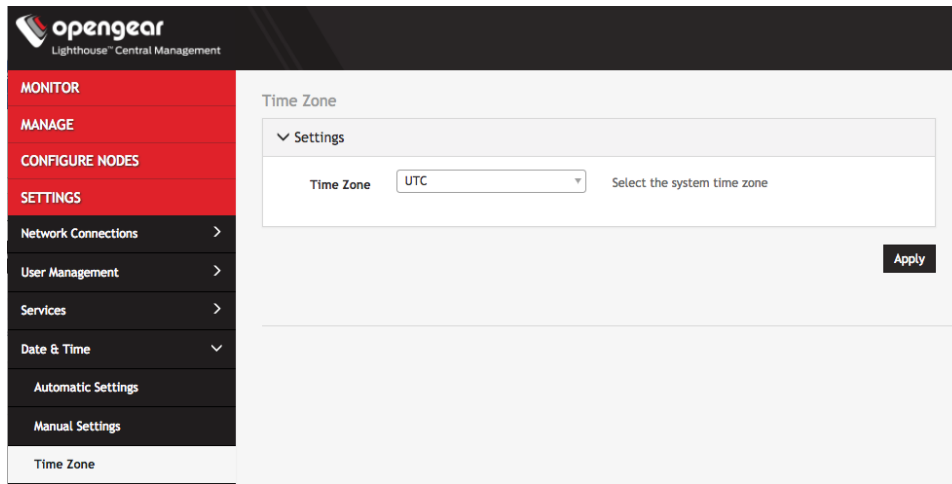
The external IP addresses are sent to a remote node during enrollment in the following order:

1. `net1:dhcp`
2. `net1`
3. The IP address connected to the default gateway.

5.7 Setting the Lighthouse internal clock

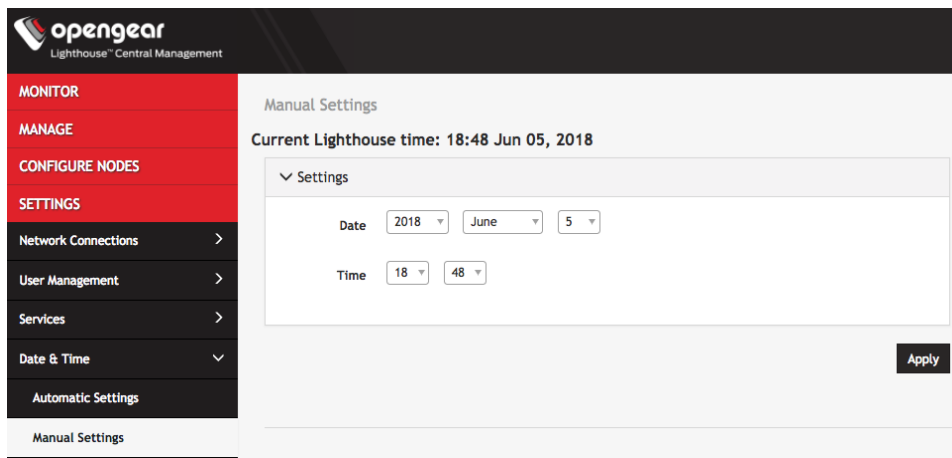
To set the time zone:

1. Select **SETTINGS > Date & Time > Time Zone**.
2. Select the Lighthouse instance's time-zone from the **Time Zone** drop-down list.
3. Click **Apply**.



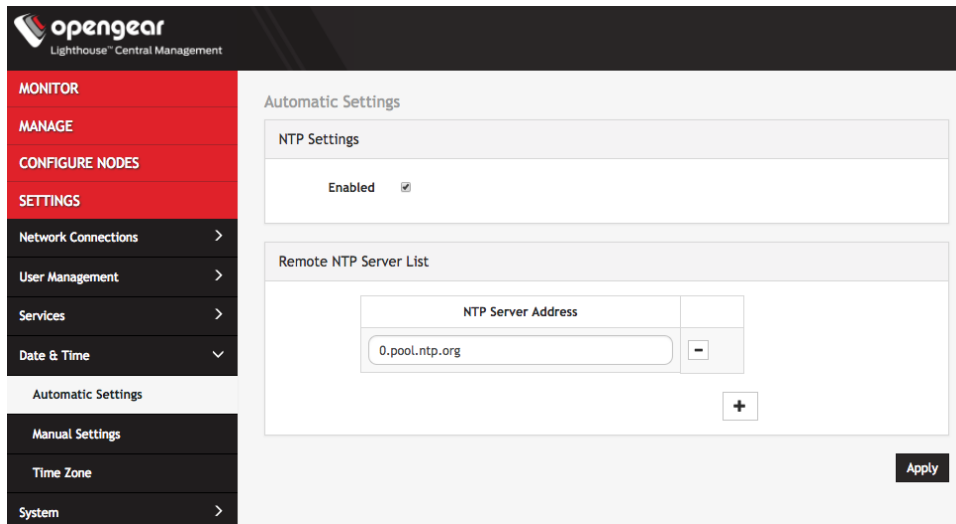
To set the correct time and date, either

1. Select **SETTINGS > Date & Time > Manual Settings**.
2. Enter the current **Date** and **Time**.
3. Click **Apply**.



or

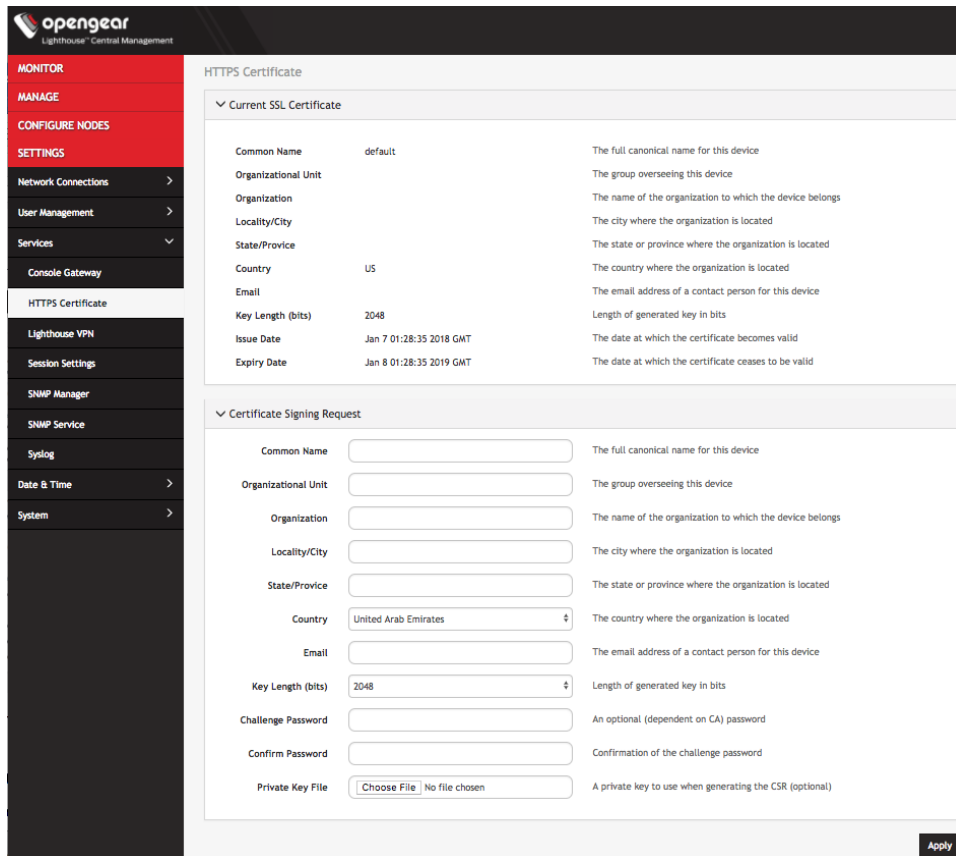
1. Select **SETTINGS > Date & Time > Automatic Settings**.
2. Click the **Enabled** checkbox.
3. Enter a working NTP Server address in the **NTP Server Address** field.
4. Click **Apply**.



5.8 Examine or modify the Lighthouse SSL certificate

Lighthouse ships with a private SSL Certificate that encrypts communications between it and the browser.

To examine this certificate or generate a new Certificate Signing Request, select **SETTINGS > Services > HTTPS Certificate**. The details of the **Current SSL Certificate** appear.

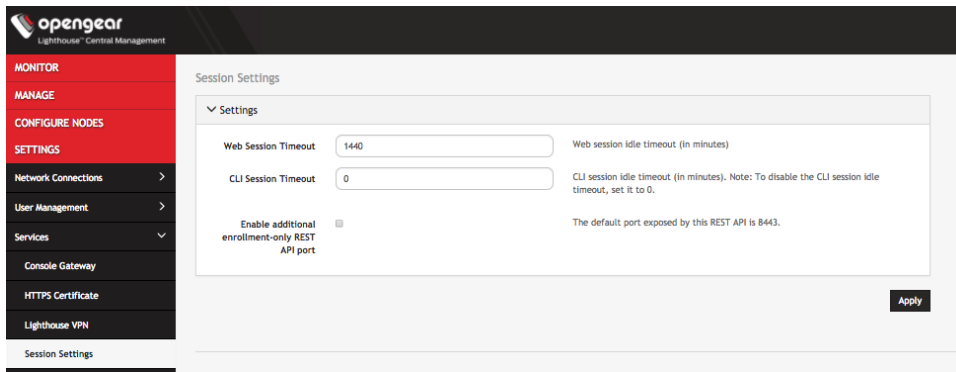


Below this listing is a **Certificate Signing Request** form, which can be used to generate a new SSL certificate.

5.9 Examine or modify Lighthouse Session Settings

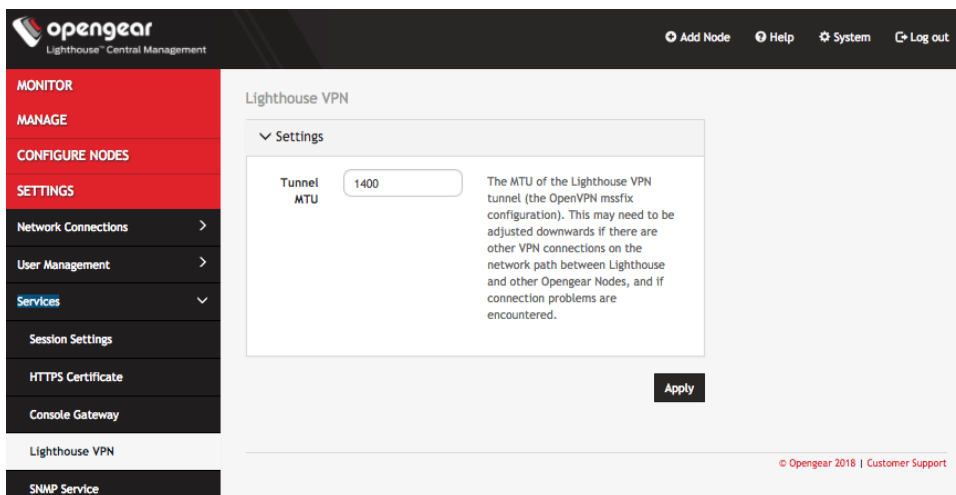
To modify Web and CLI session settings select **SETTINGS > Services > Session Settings**.

- **Web Session Timeout:** This value can be set from 1 to 1440 minutes.
- **CLI Session Timeout:** This value can be set from 1 to 1440 minutes or set it to 0 to disable the timeout. Changes take effect the next time a user logs in via the CLI.
- **Enable additional enrollment-only REST API port:** This port defaults to 8443. When this option is enabled, only /nodes endpoint is accessible via port 8443(GET/POST/PUT) and all other endpoints return a *404 Not Found* error. Enabling this API allows users who are using NAT for the Lighthouse to expose an external port publicly only for nodes that are attempting to enroll to the Lighthouse, and not for the other functionality available from the REST API. After this option is disabled, all endpoints should be accessible as per normal usage.



5.10 Examine or change the MTU of the Lighthouse VPN tunnel

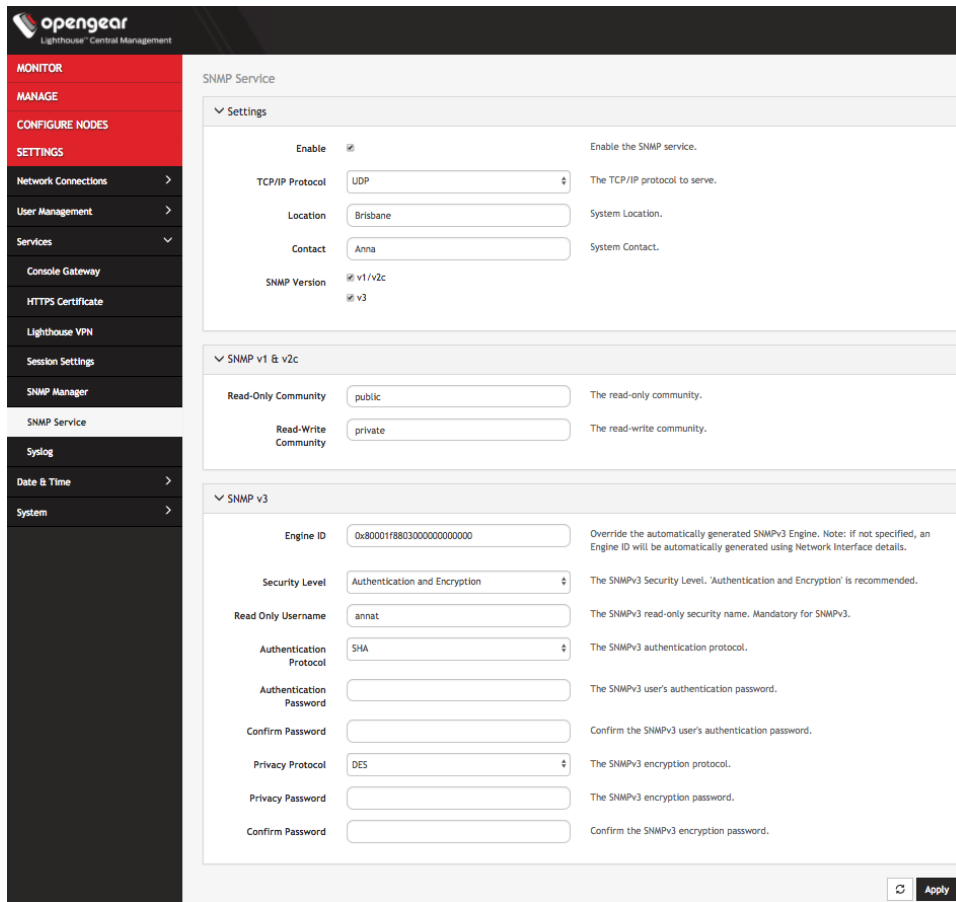
The MTU setting can be configured for traffic that is travelling through the Lighthouse VPN in an attempt to solve MTU path discovery problems. To examine the MTU of the Lighthouse VPN tunnel, or to modify it, select **SETTINGS > Services > Lighthouse VPN**. Allowed values are between 1280 and 1500.



5.11 Enable or modify SNMP Service

Administrative users can configure SNMP settings under **SETTINGS > Services > SNMP Service**.

Lighthouse supports both v1/v2 and v3 SNMP versions, which can be running at the same time. The SNMP service is not enabled by default and starts once it has been configured correctly. If the user does not provide an engineID, the auto-generated ID coming out of snmpd are displayed. Only standard enterprise MIBs can be used currently, Lighthouse Health statistics (load/uptime/memory usage, etc.) can be retrieved.



To enable SNMP Service,

1. Select the **Enable** checkbox.
2. Choose from the **v1/v2c** and **v3** checkboxes.
3. Fill in the appropriate information for the SNMP versions.
4. Click **Apply**.

5.12 Lighthouse MIBs

Lighthouse MIBs can be found in `/usr/share/snmp/mibs/`.

Lighthouse can be configured to expose managed node information such as node name, node model number, node port label, license status, etc. via SNMP.

Some generic information about Lighthouse version and nodes count can be found at:
ogLhStatus:

- ogLhVersion
- ogLhNodes
 - ogLhNodesTotal
 - ogLhNodesPending
 - ogLhNodesConnected

ogLhNodesDisconnected
ogLhNodesTable with detailed information about nodes.

For enrolled Opengear node, the following information is available.

ogLhNodesTable:

- ogLhNodeIndex
- ogLhNodeName
- ogLhNodeModel
- ogLhNodeProductType
- ogLhNodeVpnAddress
- ogLhNodeSerialNumber
- ogLhNodeUptime
- ogLhNodeConnStatus

ogLhNodePortsTable:

- ogLhPortIndex
- ogLhPortLabel
- ogLhPortID

ogLhNodeInterfacesTable:

- ogLhNodeInterfaceIndex
- ogLhNodeInterfaceName
- ogLhNodeInterfaceAddress

For enrolled third-party node, the following information is available:

ogLhThirdPartyNodesTable:

- ogLhThirdPartyNodeIndex
- ogLhThirdPartyNodeSSHPort
- ogLhThirdPartyNodeName
- ogLhThirdPartyNodeModel
- ogLhThirdPartyNodeProductType
- ogLhThirdPartyNodeAddress
- ogLhThirdPartyNodeSerialNumber
- ogLhThirdPartyNodeUptime
- ogLhThirdPartyNodeConnStatus

ogLhThirdPartyNodePortsTable:

- ohLhThirdPartyPortIndex
- ogLhThirdPartyPortLabel
- ogLhThirdPartyPortConnectionMethod
- ogLhThirdPartyPortMode
- ogLhThirdPartyRemotePort
- ogLhThirdPartyPortLineID

You can also query for licensing information.

ogLhLicenseStatus:

- ogLhLicInstalled
- ogLhLicSupported
- ogLhLicExpiry
- ogLhLicStatus
- ogLhLicFeatureName

SNMP commands such as `snmpwalk` or `snmpget` retrieve Lighthouse specific information.

Setup: SNMP is configured with version 1 and public is community string
Lighthouse public IP address is 192.168.1.1
All MIBs, including Lighthouse MIB are available in /usr/share/snmp/mibs

Below are some examples of Lighthouse MIB queries using SNMP:

Walk through the entire ogLighthouseMib using name:
snmpwalk -m ALL -v1 -c public 192.168.1.1 ogLighthouseMib
snmpwalk -m ALL -M /usr/share/snmp/mibs -v1 -c public 192.168.1.1
ogLighthouseMib

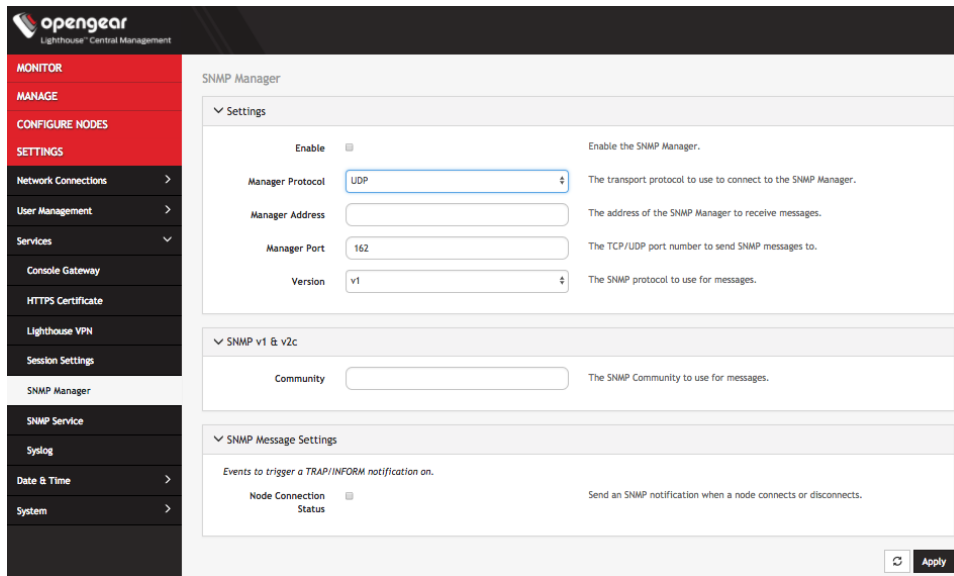
Walk through the entire ogLighthouseMib using the OID directly:
snmpwalk -m ALL -M /usr/share/snmp/mibs -v1 -c public 192.168.1.1
1.3.6.1.4.1.25049.18.1

Get the total nodes enrolled in Lighthouse:
snmpget -m ALL -v1 -c public 192.168.1.1 ogLhNodesTotal.0
snmpwalk -m ALL -v1 -c public 192.168.1.1 ogLhNodesTotal

Get serial number with enrolled node having VPN address 192.168.128.2:
snmpwalk -m ALL -v1 -c public 192.168.1.1 ogLhNodeSerialNumber.192.168.128.2
snmpget -m ALL -v1 -c public 192.168.1.1 ogLhNodeSerialNumber.192.168.128.2

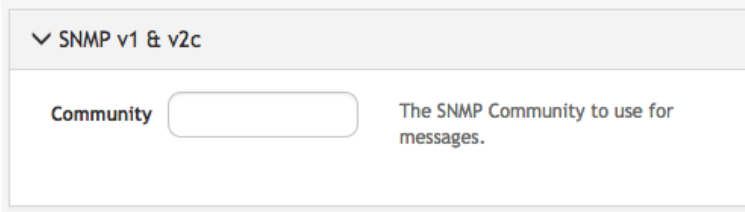
5.13 SNMP Manager Settings

Administrative users can configure the SNMP Manager settings. Select **SETTINGS > Services > SNMP Manager**. The SNMP Manager allows SNMP TRAP/INFORM messages to be sent from Lighthouse to a configured server any time a node connection status is changed.



To enable the SNMP Manager,

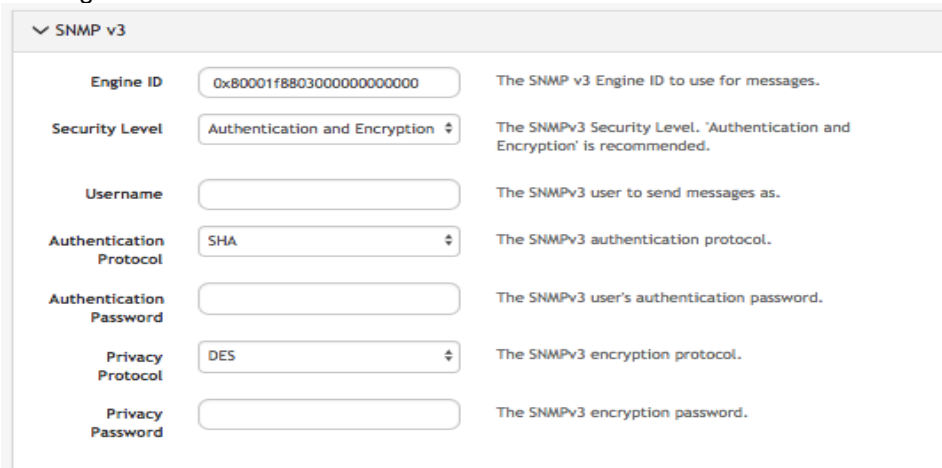
1. Under the **Settings** section, select the **Enable** checkbox.
2. Choose **UDP** or **TCP** as the **Manager Protocol** drop-down.
3. Enter the **Manager Port** to receive SNMP messages.
4. Check the SNMP protocol **Version** from the **v1**, **v2c**, **v3** drop-down.
5. Choose the **SNMP Message Type** to be sent, either **TRAP** or **INFORM**.
6. Depending on the selected SNMP version, complete the following steps.



▼ SNMP v1 & v2c

Community The SNMP Community to use for messages.

If SNMP version **v1/v2c** is selected, expand that section and enter the SNMP **Community** to use for messages.



▼ SNMP v3

Engine ID The SNMP v3 Engine ID to use for messages.

Security Level The SNMPv3 Security Level. 'Authentication and Encryption' is recommended.

Username The SNMPv3 user to send messages as.

Authentication Protocol The SNMPv3 authentication protocol.

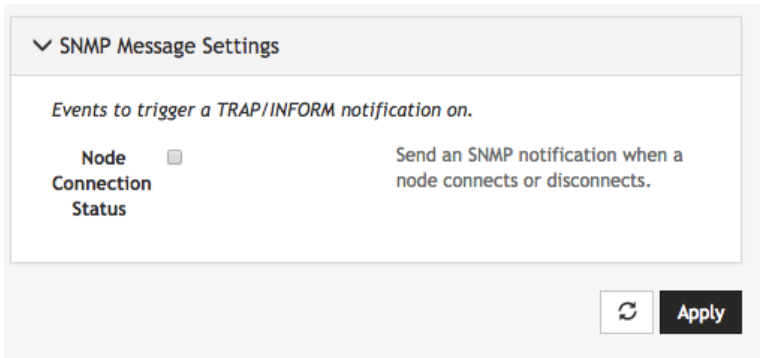
Authentication Password The SNMPv3 user's authentication password.

Privacy Protocol The SNMPv3 encryption protocol.

Privacy Password The SNMPv3 encryption password.

If SNMP version **v3** is selected, expand that section and fill in the following:

1. Specify an optional **Engine ID** for sending an SNMP TRAP message. If left blank, the auto-generated Engine ID from the SNMP Service will be used. An EngineID is not needed for an SNMP INFORM message.
2. Select the desired **Security Level**.
3. Enter the SNMPv3 **Username** to send messages as.
4. Select the desired **Authentication Protocol**, either **MD5** or **SHA**.
5. Enter the **Authentication Password** for the user.
6. Choose the **Privacy Protocol**, either **DES** or **AES**.
7. Enter the **Privacy Password**.



8. Finally, to activate notifications, expand the **SNMP Message Settings**. To trigger a TRAP/INFORM notification whenever a node connection status is changed, check the **Node Connection Status** checkbox.
9. Click **Apply**.

NOTE: Lighthouse can deliver SNMP notifications to a configured SNMP manager upon connection status change of nodes when configured to do so.

When a node connection status changes, a *nodeStatusNotif* notification is sent, populated with data about the node's connection status, address and name.

Structure of notifications for Opengear nodes:

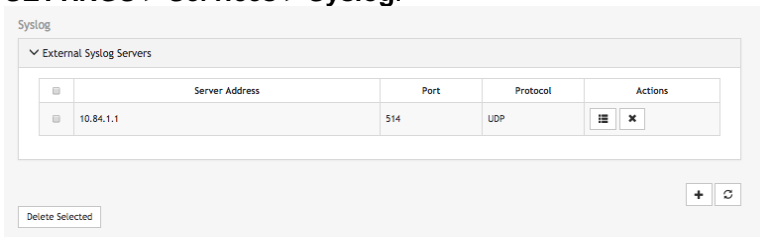
```
nodeStatusNotif
  ogLhNodeName
  ogLhNodeIndex
  ogLhNodeConnStatus
```

Structure of notifications for third-party nodes:

```
thirdPartyNodeStatusNotif
  ogLhThirdPartyNodeIndex
  ogLhThirdPartyNodeName
  ogLhThirdPartyNodeAddress
  ogLhThirdPartyNodeConnStatus
```

5.14 Syslog export

Administrative users can specify multiple external servers to export the syslog to via TCP or UDP. Select **SETTINGS > Services > Syslog**.



This page lists any previously added external syslog servers. To add a new one,

1. Click the plus sign (+) at the end of the list. The **Add External Syslog Server** dialog opens.

Add External Syslog Server

Server Address	<input type="text"/>	The address of the external syslog server
Protocol	<input type="text" value="UDP"/>	The protocol used to send syslog messages
Port	<input type="text"/>	The port to use to communicate with the syslog server For UDP, the default is 514 For TCP, the default is 601

2. Enter the **Server Address**.
3. Enter the **Protocol**, either **UDP** or **TCP**.
4. Enter the correct **Port**. If no port is entered, UDP defaults to port 514 and TCP defaults to 601.
5. Click **Apply**.

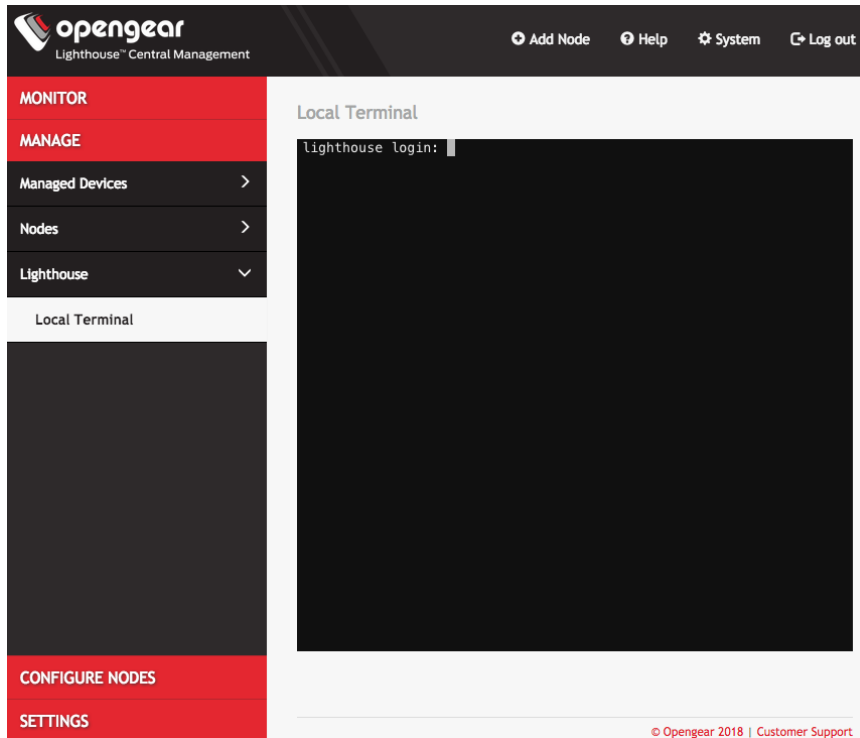
To edit an existing syslog server, click the **Edit** button  under **Actions**. Delete a server by clicking the X button .

6. Shut down or restart Lighthouse

6.1 Shut down a running Lighthouse instance

To shut down a running Lighthouse instance:

1. Select **MANAGE > Lighthouse > Local Terminal**



2. At the **Local Terminal** login prompt enter a username with administrative privileges (e.g. **root**).
3. At the **Password:** prompt, enter that account's password. A **Last login** date and time for that account are returned to `STD OUT` and a shell prompt for the logged in user appears.
4. Enter the command `shutdown now` and press **Return**. The virtual machine shuts down.

6.2 Restarting a running Lighthouse instance

To restart a running Lighthouse instance, follow the first three steps of the *Shutting down a running Lighthouse instance* procedure above. At the shell prompt, enter one of these commands and press **Return**:

- `reboot`
- `shutdown -r now`

The Lighthouse virtual machine shuts down and reboots.

7. Using Lighthouse

After Lighthouse has been installed and configured, a small set of nodes should be enrolled, and a set of tags and smart groups should be created that allow nodes access to be filtered to the correct subset of users.

Once these nodes are installed, access to the Node's Web UI and serial ports should be tested.

This section covers:

1. Licensing third-party nodes before enrollment
2. Enrolling nodes
3. The Enrolled Nodes page
4. Filtering pages displaying nodes
5. Creating Smart Groups
6. Editing an existing Smart Group
7. Creating Managed Device Filters
8. Editing an existing Managed Device Filter
9. Connecting to a node's web-management interface
10. Connecting to a node's serial ports via Console Gateway

7.1 Licensing third-party nodes before enrollment

Lighthouse includes support for managing third-party remote nodes. Support for third-party remote nodes is not built-in to a new Lighthouse instance, however: it is added via a license.

A license is an encrypted, RFC 7519-compliant, JSON web token that contains key-value pairs describing the features and entitlements of a given third-party remote node. Licenses are distributed by Opengear and are available as encrypted ASCII strings sent by e-mail via a fulfillment procedure.

Before enrolling a third-party remote node, its corresponding license must be added to Lighthouse as follows:

7.1.1 Adding a license using the Lighthouse UI

1. Select **SETTINGS > System > Licensing**
2. Click the **+** button.

New License

License body

Cancel Apply

3. Paste the encrypted license text string into the **License body** text box.
4. Click **Apply**.

7.1.2 Showing installed licenses in the Lighthouse UI

To see all installed licenses, select **SETTINGS > System > Licensing**.

opengear
Lighthouse™ Central Management

MONITOR
MANAGE
CONFIGURE NODES
SETTINGS

Network Connections >
User Management >
Services >
Date & Time >
System >
Administration
Factory Reset
Licensing
System Upgrade

Licensing

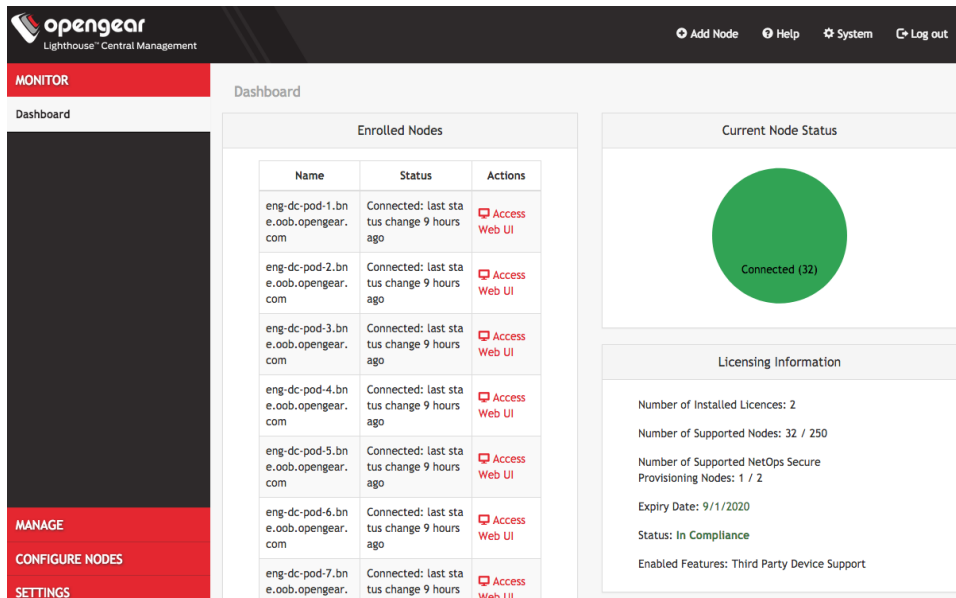
License Information

OGLH

SKU	OGLH	SKU Code for the license information
Customer Name	N/A	The customer name for this license
Email	nocontact@opengear.com	The email address of a customer for this license
License Expiry	9/1/2020	The date when license will expire (aggregated date of expiry across all applied licenses for this SKU)
Number of Supported Devices	250	The number of supported devices for this license (aggregated across all applied licenses for this SKU)
License Features	Third Party Device Support	Additional features supported

+ ↻

Installed licenses are also shown on the Lighthouse dashboard at **MONITOR > Dashboard**.



The dashboard also displays messages when:

- The number of nodes supported by a license has been reached or exceeded.
- The maintenance period of a license has expired.

7.1.3 Showing installed licenses via the Local Terminal

`oglicdump` is a shell-based tool that writes the current licensing status of a Lighthouse instance to `STDOUT` (or, using the `-o` switch, a file).

For example:

```
# oglicdump
{
  "OGLH": {
    "contact": {
      "email": "nocontact@opengear.com",
      "name": "N/A",
      "phone": "N/A"
    },
    "features": {
      "additional": {
        "thirdpartynodes": "1"
      },
      "maintenance": 1599004800,
      "nodes": 250
    }
  }
}
```

If no licenses are installed, **oglicdump** returns the following:

```
# oglicdump
```

No data found

7.2 Enrolling nodes

7.2.1 Enrollment overview

Enrolling nodes is the process of connecting nodes to Lighthouse to make them available for access, monitoring, and management. Enrollment can be performed via:

- The Lighthouse Web UI
- The Node Web UI
- ZTP
- USB key

Credentials must be provided to authenticate either the Lighthouse during enrollment via the Lighthouse WebUI, or the node during the other enrollment scenarios.

The Lighthouse VPN uses certificate-authenticated OpenVPN tunnels between Lighthouse and remote nodes. These tunnels rely on the time being synchronized between the Lighthouse instance and the console server or other remote node. During enrollment, if a remote node is not relying on an NTP server to set its time, it inspects the **HTTP Date** header sent by Lighthouse and sets its local time to match that of the Lighthouse instance.

If a remote node is relying on an NTP server to set its own time, it still checks the **HTTP Date** header sent by Lighthouse to affect the time synchronization but does not set its local time to that of the Lighthouse instance.

When enrolling via Lighthouse, an administration username and password for the node must be provided. When enrolling via the node, an enrollment **token** must be provided. A default enrollment token can be set on the **CONFIGURE NODES > Node Enrollment > Enrollment Settings** page, and individual tokens set per enrollment bundle.

Enrollment is a two-step process:

1. Once enrollment starts, nodes receive their enrollment package, and establish a VPN connection to Lighthouse.
2. The node is now in the **Pending** state and needs to be **Approved** before the node is available for access, management, or monitoring.

NOTE: This second step can be skipped by selecting the **Auto-approve node** checkbox when configuring an enrollment bundle.

7.2.2 Enrollment bundles

An enrollment bundle is a downloadable file that stores provisioning information, allowing for bulk enrollment and manipulation of remote nodes.

Applying an enrollment bundle during enrollment allows tags to be associated with nodes when they're first enrolled, rather than manually assigning tags after the nodes are enrolled.

This is useful for larger roll-outs where there are many nodes deployed with a similar configuration and responsibilities. If relevant Smart Groups and tags have been set up, newly enrolled nodes are immediately visible for the relevant users to configure and use.

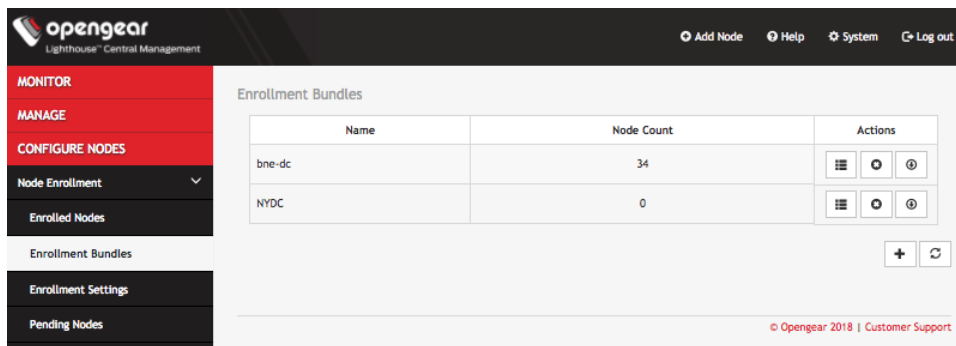
Associating templates with an enrollment bundle allows to run a set of templates on a node, after it has been enrolled. Any template defined on the Lighthouse can be added to an enrollment bundle, and each bundle supports any number of templates.

NOTE: With this release, NetOps modules (see Chapter 10) can also be associated with enrollment bundles.

7.2.3 Creating an enrollment bundle

An enrollment bundle can be created in a Lighthouse instance as follows:

1. Select **CONFIGURE NODES > Node Enrollment > Enrollment Bundles**



2. Click the + button. The **Enrollment Bundle Details** page appears.

The screenshot shows the 'Enrollment Bundles' configuration page in the OpenGear Lighthouse Central Management interface. The sidebar on the left contains navigation options: MONITOR, MANAGE, CONFIGURE NODES, Node Enrollment (expanded), and Enrolled Nodes. The main content area is titled 'Enrollment Bundles' and contains the following sections:

- Enrollment Bundle Details:** Includes input fields for 'Name' (Descriptive bundle name) and 'Token' (Authentication token used to request enrollment with this bundle). There is also an 'Auto-approve node' checkbox with the label 'Automatically approve enrollment after identification'.
- Enrollment Bundle Node Tags:** Includes a text box for 'Tag values specified here will automatically be applied to any nodes enrolled against this bundle.' and two dropdown menus with a '+' and '-' button.
- Templates:** Includes a text box for 'Templates selected here will automatically be applied to any nodes enrolled against this bundle in the specified order. Template push operations will stop from continuing if one fails.' Below this is a table with columns 'Order', 'Template Type', 'Template Name', and 'Actions'. The table currently shows 'No Templates have been selected' and a '+' button.
- NetOps Modules:** Includes a text box for 'NetOps Modules selected here will automatically be activated on any supported nodes enrolled against this bundle.' Below this is a table with columns 'Order', 'Module Name', and 'Actions'. The table currently shows 'No modules have been selected' and a '+' button.

At the bottom right of the main content area, there are 'Cancel' and 'Apply' buttons.

3. Enter a **Name** and **Authentication Token** for the bundle in the respective fields.
4. Select the number of **Tags** and **Values** to apply to any nodes that enroll using this enrollment bundle.
5. (Optional) Select the **Auto-approve node** checkbox.

When this is checked, a device configured using this enrollment bundle is not placed in pending mode during the enrollment process. Instead, it is automatically approved for enrollment after it has been identified.

- You can also use this bundle to automatically activate NetOps modules for any supported nodes. Click the **+** button under **the NetOps Modules** section. The **Module Details** page appears.

- Select the desired **Module Name** from the drop-down list. Click **Apply**.

With the enrollment bundle named, use the **Enrollment Bundle Node Tags** to populate it with the desired name-value pairs:

- Select a field name from the left-most drop-down menu.
- Select or enter a value from the right-most drop-down menu.
- Click the **+** button to add a new pair of drop-down menus.
- Select another field name and select or enter another value.
- Repeat until all desired name-value pairs are displayed.
- Click **Apply**.

With the enrollment bundle named, use the **Templates** to populate it with the desired list of templates to be applied post-enrollment:

- Click the **+** button to add a new pair of drop-down menus.
- Select a value from the **Template Type** menu. The selected template type filters the available names to those templates of that type.
- Select a value from the **Template Name** menu.
- Repeat until all desired type-name pairs are displayed.
- Click **Apply**.
- The templates in the table can be reordered using the arrow buttons in the far-left column of the table and are executed in the order they appear. The order buttons appear if there is more than one template in the table.

Template push operations stop if one template fails.

7.2.4 Structure of an enrollment bundle

An enrollment bundle file, `manifest.og`, contains a series of field-value pairs that an unconfigured device can use to configure itself.

Options that can be set in `manifest.og` include new firmware, custom configuration scripts, OPG config files, and Lighthouse enrollment details.

By default, `manifest.og` includes the following field-value pairs (with example values):

```
address=192.168.88.20
```

```
api_port=4443
bundle=bne-dc
password=secret
```

Custom field-value pairs can be added manually. The field names are potential field names for a real-world, customized file, but the values following each field name are examples:

```
script=configure_ports.sh
image=acm7000-3.16.6.image
external_endpoints=192.168.1.2:4444,192.168.1.3:4445
```

7.2.5 Enrollment via Lighthouse Web UI

Enrollment via Lighthouse Web UI only works if the Node is reachable from Lighthouse.

1. Select the **Add Node** shortcut in the top menu bar to bring up the new enrollment dialog.
2. Select the **Product** type from the **Product** drop-down menu.
3. Available options in the **Product** drop-down menu are:
 - An Opendgear device
 - A generic third-party device
 - An Avocent ACS6000
 - An Avocent ACS8000
 - An Avocent ACS Classic
 - A Cisco 2900 Series

Cancel Apply

NOTE: Enrolling an Avocent ACS6000, an Avocent ACS8000, an Avocent ACS Classic, or a Cisco 2900 Series requires the device's license to have been added as per the *Licensing third-party nodes before enrollment* procedure above. If an appropriate license has not been added to Lighthouse, the procedure returns a **No licenses have been applied** error and the node is not added to Lighthouse.

4. Enter the **Name**, **Network Address**, **Username**, and **Password** of the node being enrolled. The **Username** and **Password** fields are for the login credentials required by the remote node being enrolled, *not* the login credentials used to login to the Lighthouse instance.

New Enrollment	
Product	<input type="text" value="An Opengear device"/> The type of device to enroll
Network Address	<input type="text"/> Current network address for this node
Username	<input type="text"/> Admin username to use when connecting to the node
Password	<input type="text"/> Admin user password for the node
Auto-approve node	<input checked="" type="checkbox"/> Automatically approve enrollment after identification

NOTE: Lighthouse populates the node name field with the hostname of the enrolled node rather than a user provided value. It is no longer possible for users to specify a custom name, except when enrolling third party nodes. Console servers with firmware 4.1.1 and higher provide their hostname in the node information, with pre-4.1 nodes instead just having their node id used as the name. Nodes enrolled prior to upgrading to 5 have their names switched to the new standard if the node is running 4.1.1 firmware but retain their old name if older firmware is still installed.

5. To enroll a generic third-party device, there are three more required fields: **Connection Method**; **Base Protocol Port**; and **Port Count**.

NOTE: The following procedure assumes the third-party device's license has been added as per the *Licensing third-party nodes before enrollment* procedure above. If an appropriate license has not been added to Lighthouse, the procedure returns a **No licenses have been applied** error and the node is not be added to Lighthouse.

New Enrollment

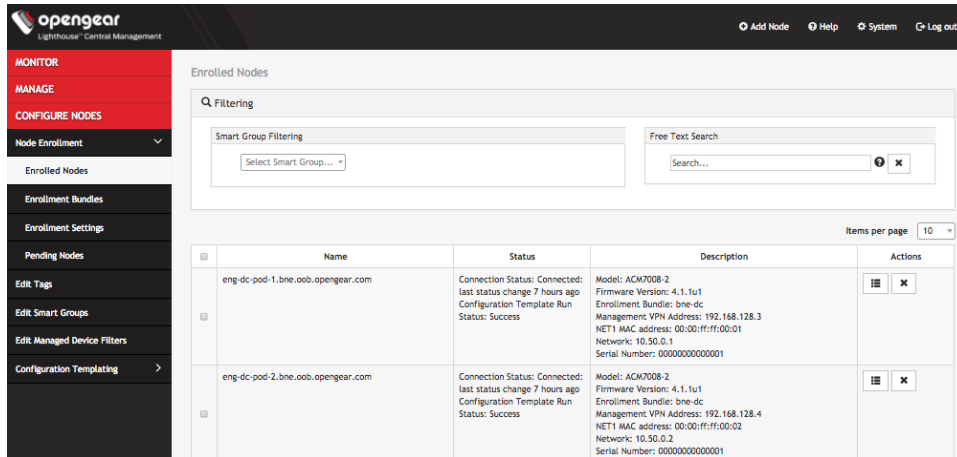
Product	<input type="text" value="A generic third party device"/>	The type of device to enroll
Name	<input type="text"/>	Brief name for this node
Network Address	<input type="text"/>	Current network address for this node
Connection Method	<input type="text" value="SSH"/>	The protocol used to connect to serial ports
Username	<input type="text"/>	Username for a node user with access to all serial ports
Password	<input type="text"/>	Password for a node user with access to all serial ports
Auto-approve node	<input checked="" type="checkbox"/>	Automatically approve enrollment after identification
Base Protocol Port	<input type="text" value="3000"/>	The base number from which network ports for individual serial ports will be derived
Port Count	<input type="text" value="4"/>	The number of serial ports on the target device (maximum 400)

Serial Port Labels

Port Label 1	<input type="text" value="Port 1"/>
Port Label 2	<input type="text" value="Port 2"/>
Port Label 3	<input type="text" value="Port 3"/>
Port Label 4	<input type="text" value="Port 4"/>

6. Choose **SSH** or **Telnet** from the **Connection Method** drop-down menu, as appropriate for the connection method supported by the third-party device.
7. Enter a base number in the **Base Protocol Port**. By default, this is set to 3000. The Base Protocol Port number is the starting port number from which the third-party device's individual serial port network port numbers will be derived.
8. Enter the number of serial ports the third-party device has in the **Port Count** field. Below the **Port Count** field is a **Serial Port Labels** section. Whatever number is entered in the **Port Count** field, the **Port Label x** fields in this section update to match.
9. Optionally, edit the labels used to identify each serial port in the **Serial Port Labels** section.
10. Click **Apply**.

Once enrolled, the console server's details are removed from the **Pending Nodes** page and added to the **CONFIGURE NODES > Node Enrollment > Enrolled Nodes** page.



7.2.6 Enrollment via Node Web UI

If the node is situated behind a firewall, Lighthouse is not able to initiate an enrollment. It needs to be triggered from the Node Web UI.

1. Log into the Node Web UI.
2. Select **Serial & Network > Lighthouse**.
3. Enter the **Server Address**.
4. Optionally, enter the **Server Port**.
5. Enter the **Enrollment Bundle** (if a specific bundle is being used), and the **Enrollment Token** (either the global token or the bundle-specific token).
6. Select **Apply Settings**. The enrollment process begins.

7.2.7 Lighthouse Enrollment via OM2200 Web UI

OM2200 nodes can be enrolled into a Lighthouse instance on OM2200 Web UI using the **CONFIGURE > Lighthouse Enrollment** menu item and the `lhvpn-callhome` command. See the OM2200 User Guide for more details.

7.2.8 Mass Enrollment using ZTP

For mass node enrollments using ZTP, three new custom DHCP fields are handled by ZTP scripts.

These fields contain the **URL**, **Bundle Name** and **Enrollment Password** used in an enrollment which is kicked off after all other ZTP handling is completed. If a reboot is required because of a config file being provided the enrollment starts after the reboot. Otherwise it happens immediately.

Here is a sample configuration file for the ISC DHCP Server:

```
option space opengear code width 1 length width 1;  
option opengear.config-url code 1 = text;  
option opengear.firmware-url code 2 = text;  
option opengear.enroll-url code 3 = text;  
option opengear.enroll-bundle code 4 = text;  
option opengear.enroll-password code 5 = text;
```

```
class "opengear-config-over-dhcp-test" {
  match if option vendor-class-identifier ~~ "^Opengear/";
  vendor-option-space opengear;
  option opengear.config-url "http://192.168.88.1/config.xml";
  option opengear.enroll-url "192.168.88.20";
  option opengear.enroll-bundle "";
  option opengear.enroll-password "default";
}
```

NOTE: The maximum amount of data allowable as DHCP options is 1200 bytes, including all overhead inherent in the structuring of this data. Individual options are limited to 255 characters.

7.2.9 Enrollment via USB drive

USB Enrollment enables the configuration of a device using a manifest file copied to a USB drive and inserted into the unconfigured device before it first boots.

Once created (see *Creating an enrollment bundle* above), `manifest.og` files can be downloaded from a Lighthouse instance as follows:

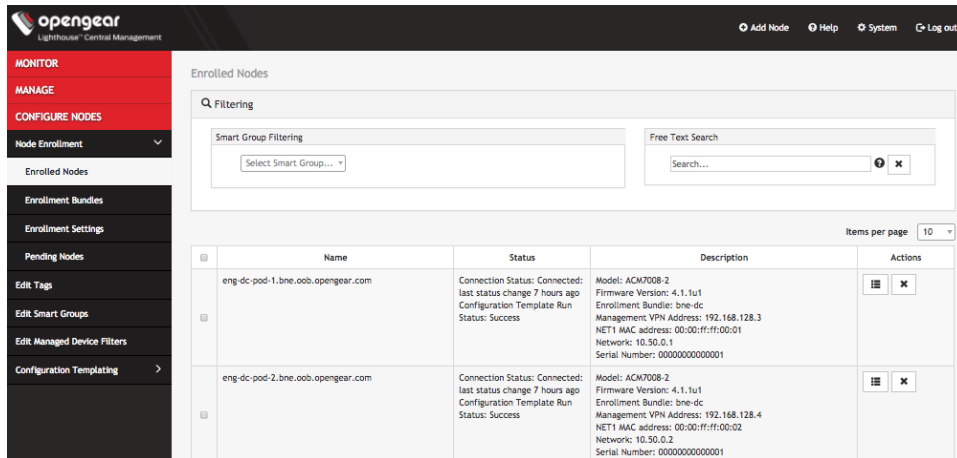
1. Select **CONFIGURE NODES > Node Enrollment > Enrollment Bundles**. A list of existing **Enrollment Bundles** appears.
2. In the **Actions** column of the particular bundle, click the **download** button, a downward arrow in a circle.
3. Depending on the browser's configuration, a `manifest.og` file is either downloaded to the local system or the browser opens a dialog asking to specify where download should be copied.

To enroll via USB drive:

4. Copy `manifest.og` to the root directory on a USB drive.
5. Plug the USB drive into an unconfigured and powered-down console server.
6. Power the console server up.

On first boot, the device looks for a file — `manifest.og` — on any USB drives attached to the device and configures the device based on their contents.

7.3 The Enrolled Nodes page



CONFIGURE NODES > Node Enrollment > Enrolled Nodes lists all enrolled nodes in the order they are enrolled to Lighthouse.

The **Items per page** drop-down allows user to select the number of nodes per page. Choose a default value of 10, 20, 50, 80, or 100 nodes per page, or enter a Text custom value between 1 and 100. This setting applies to the current user session only and will be lost when current user logs out. This drop-down is also presented on Pending Nodes, Console Gateway, and Node Web UI pages.

It also displays details about each node (such as model, firmware version, serial number) and status.

Connection Status is the current status of the node and displays either of two things:

- **Connected: Last status change x [time unit] ago:** The time since Lighthouse connected to the console server.
- **Disconnected: last status change x [time unit] ago:** The time since Lighthouse disconnected from the console server.

Configuration Retrieval Status displays if any configuration retrieval sections failed when performing a configuration sync with this node, such as Groups, Users, Node Description, Authorization, or Serial Ports.

Configuration Template Run Status displays the result of the most recent configuration template push on this node, listing which templates finished applying, or failed to apply to the node. This information is displayed until the next template push has completed on this node.

The **Configuration Retrieval Status** and **Configuration Template Run Status** are not displayed if there is no relevant data to display and are only displayed for users with **Lighthouse Administrator** or **Node Administrator** permissions.

Results of the **Configuration Retrieval Status** and **Configuration Template Run Status** will indicate:

- **Success:** all templates were successfully executed on the node.
- **Partial Failure:** some templates failed to execute on the node, or some config sections failed to synchronize.

- **Failure:** all templates failed to execute on the node, or all config sections failed to synchronize.

The detailed information is shown in a popover that appears when the summary of each status is clicked on, navigated to, or hovered over. The format of the detailed information for each status shown on relevant popovers is as follows:

- Retrieval failed for: section_name, section_name, section_name.
- Template(s) failed to apply: template_name, template_name, template_name.
- Template(s) successfully applied: template_name, template_name, template_name.

7.4 Filtering pages displaying nodes

There are three ways to filter search results: Free Text Search, Smart Group Filtering, and Managed Device Filtering. They can be used independently from each other or in combination. **MANAGE > Managed Devices > Console Gateway** uses all of them because it is the only page which lists all nodes with managed devices.

7.4.1 Filtering using the Free Text Search field

The Free Text Search text-entry field allows the near real-time filtering. It searches over node name, firmware version, management VPN address, MAC address, and serial number. Type a string (e.g. *4.1.1* or *192.168.128.1* or *CM7148*) and press **Return**. Only nodes which include that string in their **Name** or **Description** are displayed.

The Free Text Search field treats multiple search terms (i.e. terms delimited by the space character) as Boolean AND searches.

For example, a search on the string:

4.1.1 CM7148

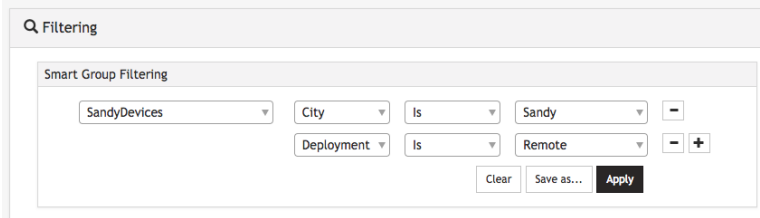
returns any nodes that have both *CM7148* AND *4.1.1* in searchable fields (e.g. *CM7148* in the name field and *4.1.1* in the firmware version field).

To make a search string that contains spaces into a single searched entity, enclose the string in double quotes.

7.4.2 Filtering using the Smart Group Filtering drop-down menu

Selecting from the **Select Smart Group** drop-down menu sets the page to display the subset of nodes that belong to the selected group. See *Creating Smart Groups* below for how to create such groups.

Once a particular Smart Group has been selected, further filtering options become available. For example:



The image shows a 'Filtering' section with a search icon and the text 'Filtering'. Below it is a 'Smart Group Filtering' panel. The panel contains a dropdown menu with 'SandyDevices' selected. To its right are two rows of filters. The first row has 'City' as a dropdown, 'Is' as an operator, and 'Sandy' as a value. The second row has 'Deployment' as a dropdown, 'Is' as an operator, and 'Remote' as a value. There are minus and plus signs to the right of each row. At the bottom of the panel are three buttons: 'Clear', 'Save as...', and 'Apply'.

In the example above, the **CONFIGURE NODES > Node Enrollment > Enrolled Nodes** page is being filtered on the **SandyDevices** Smart Group.

It is then being further filtered to only display nodes with a **City** of *Sandy*, and a **Deployment** of *Remote*.

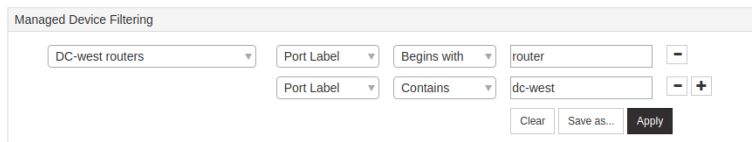
To add more filtering options:

1. Click the **+** button. An extra row of drop-down menus appears.
2. Select the desired tag from the left-most drop-down menu.
3. Select the filtering operator from middle drop-down menu.
4. Select or enter the value to be filtered against from the right-most drop-down menu.
5. Click **Apply**.

7.4.3 Filtering using the Managed Device Filtering drop-down menu

Selecting from the **Select Managed Device Filter** drop-down menu sets the page to display the subset of nodes with filtered managed devices. See *Creating Managed Device Filter* below for how to create managed device filters.

Once a particular Managed Device Filter has been selected, further filtering options become available. For example:



The image shows a 'Managed Device Filtering' panel. It contains a dropdown menu with 'DC-west routers' selected. To its right are two rows of filters. The first row has 'Port Label' as a dropdown, 'Begins with' as an operator, and 'router' as a value. The second row has 'Port Label' as a dropdown, 'Contains' as an operator, and 'dc-west' as a value. There are minus and plus signs to the right of each row. At the bottom of the panel are three buttons: 'Clear', 'Save as...', and 'Apply'.

In the example above, the **MANAGE > Managed Devices > Console Gateway** page is being filtered on the **DC-west routers** Managed Device Filter. It is then being further filtered to only display nodes with a **Port Label** Begins with *router*, and a **Port Label** Contains *dc-west*.

To add more filtering options:

1. Click the **+** button. An extra row of drop-down menus appears.
2. Select the Port Label from the left-most drop-down menu.
3. Select the filtering operator from middle drop-down menu.
4. Enter the value to be filtered against from the right-most drop-down menu.
5. Click **Apply**.

7.5 Creating Smart Groups

Smart Groups are saved search parameters used within Lighthouse for grouping related remote nodes.

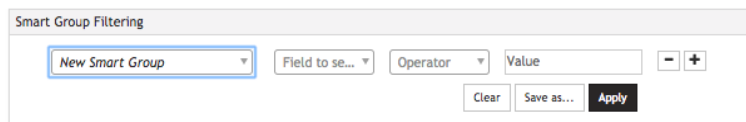
A given User Group can be linked to a particular Smart Group. When a Group is linked in this fashion, members of the Group inherit rights over all nodes in the group based on the Group's Role. See *Modifying existing groups* for how to set a Group's Role and Linked Smart Group.

Smart Groups can also be used to filter visible nodes on pages that display enrolled nodes (such as **CONFIGURE NODES > Node Enrollment > Enrolled Nodes**) to make it easier to drill down to a particular console.

Smart groups are dynamic, so as more nodes are added to the system, the filters update.

To create a Smart Group:

1. Navigate to any page which displays the Smart Group search interface, for example **CONFIGURE NODES > Node Enrollment > Enrolled Nodes** or **MANAGE > Nodes > Node Web UI**.
2. Click on the **Select Smart Group** drop-down and select **New Smart Group**. This populates a number of new drop-downs and text boxes.



The screenshot shows a 'Smart Group Filtering' interface. It features a dropdown menu on the left with 'New Smart Group' selected. To the right of this dropdown are three input fields: 'Field to se...', 'Operator', and 'Value'. Below these fields are three buttons: 'Clear', 'Save as...', and 'Apply'.

3. Click the **Field to search** drop-down to select a node attribute to filter on.

These attributes include details about the device (**Model, Firmware Version, Serial Number, NET1 MAC Address**), and include any **tags** that have been configured in the system. For filtering access to devices, tags are the most useful attributes to filter on. When a tag is selected, the **Value** text box becomes a drop-down with the values for that tag.

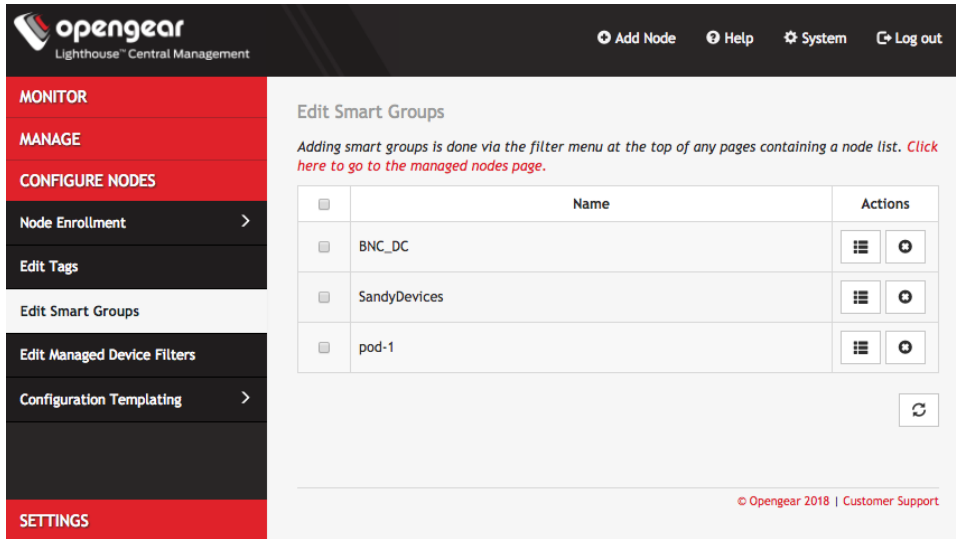
4. Click the **Operator** drop-down to select the operator to apply to the **Value**. In general, the **Is** operator is the most useful.
5. Select the **Value** to be matched against.
6. Click **Apply** to see the results of the filter.
7. Click **Save As** and type in a name for the search.

This Smart Group can now be used for filtering nodes for display, and for access.

7.6 Editing an existing Smart Group

To edit an existing Smart Group:

Select **CONFIGURE NODES > Edit Smart Groups**.



- Click the **X** icon to delete an existing Smart Group.
- Click the **Edit Group** icon to change a Smart Group's name.

To change the search parameters used by a Smart Group:

1. Navigate to a page that displays Smart Groups for filtering (e.g. **CONFIGURE NODES > Node Enrollment > Enrolled Nodes**).
2. Select the required Smart Group to be changed from the **Select Smart Group** drop-down menu.
3. Change the **Tag** and **Operator** values as required.
4. Click **Save as**.

Save Smart Group

Name Save as a new group, or overwrite the existing one

5. Leave the Smart Group name unedited and click **Apply**. The changed **Smart Group** overwrites the existing Smart Group.

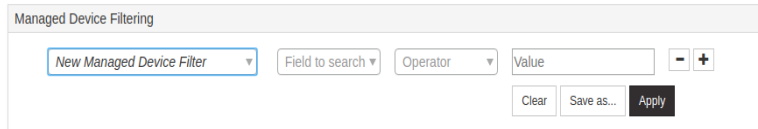
7.7 Creating Managed Device Filters

Managed Device Filters are saved search parameters for grouping related managed devices on remote nodes. Managed Device Filters can be used to filter visible nodes with managed devices on the **MANAGE > Managed Devices > Console Gateway** page to make it easier to find a particular console.

Managed Device Filters are dynamic, so as more nodes with managed devices which match saved filters are added to the system, the filters update.

To create a Managed Device Filter:

1. Navigate to the **MANAGE > Managed Devices > Console Gateway** page.
2. Click on the **Select Managed Device Filter** drop-down and select **New Managed Device Filter**. This populates a number of new drop-downs and text boxes.

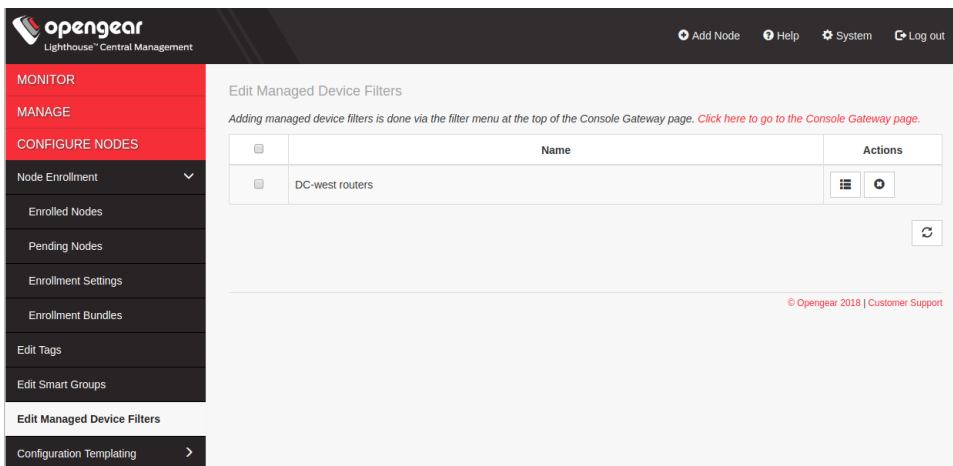


3. Click the **Field to search** drop-down to select a node attribute to filter on.
4. Select **Port Label** configuration.
5. Click the **Operator** drop-down to select the operator to apply to the **Value**. In general, the **Contains** operator is the most useful.
6. Populate the **Value** to be matched against.
7. Click **Apply** to see the results of the filter.
8. Click **Save As** and type in a name for the filter.

This Managed Device Filter can now be used for filtering nodes with managed devices.

7.8 Editing an existing Managed Device Filter

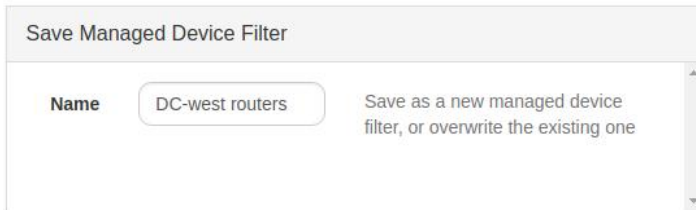
To edit an existing Managed Device Filter, select **CONFIGURE NODES > Edit Managed Device Filters** page.



- Click the **X** icon to delete an existing Managed Device Filter.
- Click the **edit** icon to change a Managed Device Filter's name.

To change the search parameters used by a Managed Device Filter:

1. Navigate to a page that displays Managed Device Filter, such as **MANAGE > Managed Devices > Console Gateway**.
2. Select the Managed Device Filter to change from the **Select Managed Device Filter** drop-down menu.
3. Change the parameters (e.g. **Operator** values) as required.
4. Click **Save as**.
5. Leave the Managed Device Filter name unedited and click **Apply**. The modified **Managed Device Filter** overwrites the existing Managed Device Filter.

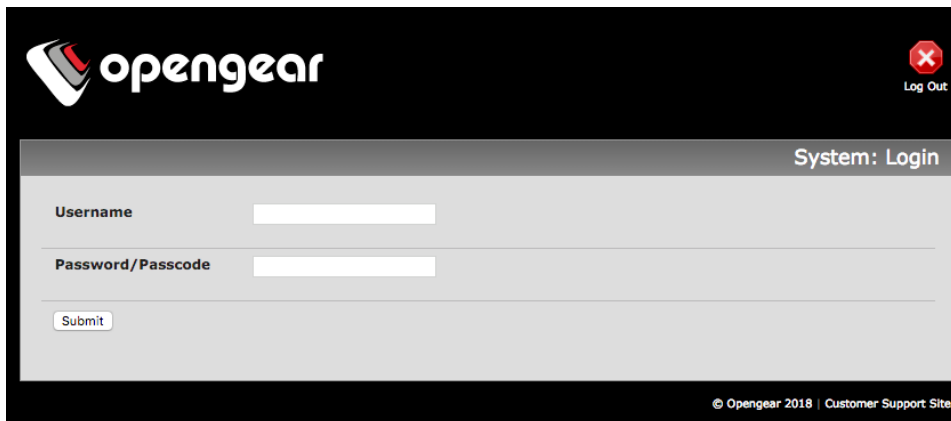


Cancel Apply

7.9 Connecting to a node's web-management interface

Once a node has been enrolled, its own web-management interface can be accessed from within the Lighthouse UI. To connect to an enrolled node's web-management interface:

1. Select **MANAGE > Nodes > Node Web UI**.
2. In the **Actions** column, click the **Access Web UI** link for the particular node. The web-based login for that node loads.
3. Authenticate using the username and password required by that node.



This system is being accessed via Lighthouse - [click here to return to Lighthouse](#)

At the bottom of the browser window is a visual indication that the console server session is being mediated through Lighthouse and a link allowing for a quick return to Lighthouse.

7.10 Connecting to a node's serial ports via Console Gateway

Searching for serial ports on Lighthouse can be accomplished by selecting **MANAGE > Managed Devices > Console Gateway** and **MANAGE > Managed Devices > Quick Search**.

The **Items per page** drop-down on Quick Search page allows user to select the number of ports per page. Choose a default value of 10, 20, 50, 80, or 100 ports per page, or enter a custom value between 1 and 100. This setting applies to the current user session only and will be lost when user logs out.

NOTE: Port-centric search allows filtering via the Managed Device Filters and displays a list of ports within enrolled nodes that match the search terms, while node-centric search allows filtering via Smart Groups and node properties. Quick Search can be used to filter on the managed device label.

Node-centric searching

1. Select **MANAGE > Managed Devices > Console Gateway**.
2. Find the particular port using the **Smart Group Filtering** options to restrict the listed nodes.
3. Click the **+** icon in the **Access Console Ports** row adjacent the particular node.

Port-centric searching

1. Select **MANAGE > Managed Devices > Console Gateway**.
2. Find the particular port by using the **Managed Device Filtering** options to restrict the listed managed devices within enrolled nodes.

Once the serial port is located, serial port access via **Console Gateway** can be accomplished in two ways:

- HTML5 Web Terminal
- SSH

Quick Search

1. Select **MANAGE > Managed Devices > Quick Search**.
2. Enter the managed device label, aka name, in the **Quick Managed Device Search** field. This search live-updates as user type.
3. Use **Web Terminal** and/or **SSH** links inside **Actions** on a particular port to access it.

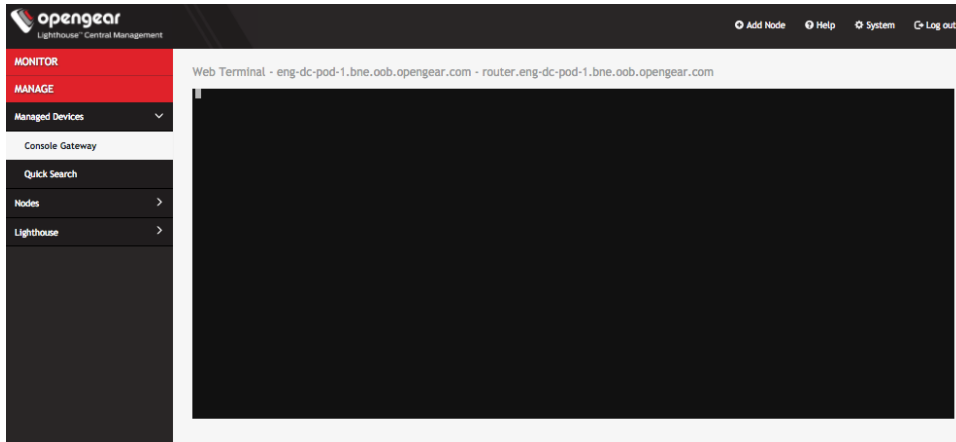
7.10.1 Access via HTML5 Web Terminal

To provide easy console port access, Lighthouse includes a HTML5 Web Terminal. The HTML5 Web Terminal includes native cut, copy and paste support. The terminals available on nodes do not.

To access a console port via the **Web Terminal**:

1. Locate the particular port by using one of the search techniques discussed above.
2. Click the **Web Terminal** link for the particular port. A new tab opens containing the **Web Terminal**.

To close a terminal session, close the tab, or type **~.** in the **Web Terminal** window.



7.10.2 Access via SSH

To access ports via SSH, the user can either use a console chooser menu to select the node and the console port or use a direct SSH link from the Web UI to connect to the port.

To access a console port via a Direct SSH link:

1. Locate the particular port by using one of the search techniques discussed above.
2. Click the **SSH** link to connect to the URL.

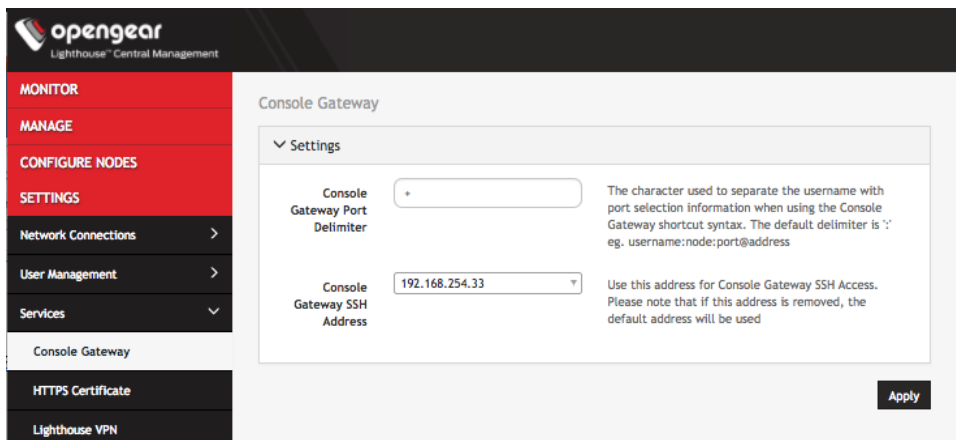
These auto-generated links use the colon (:) as the field-delimiter. The auto-generated SSH link has the following form:

```
ssh://user-name:console-server-name:port-number@lighthouse-ip-address
```

Some web browsers associate the colon character with delimiting the protocol at the beginning of a URI so they don't pass these auto-generated URIs correctly.

To work around this, the default delimiter character can be changed. To change this character:

Select **SETTINGS > Services > Console Gateway**.



- Enter a delimited character in the **Console Gateway Port Delimiter** text-entry field. The carat, ^, is the most common alternative.
- Use the **Console Gateway SSH Address** drop-down menu to choose an address from which to SSH. The list of available addresses contains the current network interfaces and external network addresses. The value defaults to *net1:dhcp* if it exists and *net1:static* otherwise. The additional external addresses can be added to this list using the **SETTINGS > System > Administration** page.

To use the console chooser menu, SSH to the Lighthouse appliance with the username format *username:serial*. This connects to the Lighthouse and presents a list of nodes that the user can access. Once the user selects a node, they are presented with a list of console ports they have access to. When one is selected, the user is connected to that port.

For faster access, there are username format shortcuts that give more specific lists of serial ports, or direct access without a menu.

- **username:node_name**
When a valid node name is specified, a list of console ports that the user can access on that node is shown. If they do not have access to this node, the connection fails.
- **username:node_name:port_name**
When a valid node name and port name are specified, and the user has access to that node and port, the user is connected to this port. If they do not have access to that port, the connection fails.
- **username:port_name**
When a valid port name is specified, the user is connected to first port with that port name found. If the user does not have access to this port, the connection fails.

NOTE: Node names and port names are not case sensitive.

7.10.3 Example Console Gateway session

```
$ ssh adminuser:serial@lighthouse-name-or-ip-here

1: cm71xx

Connect to remote > 1

1: Cisco Console                2: Port 2

Connect to port > 1
router#
```

8. Lighthouse user management

Lighthouse 5 supports locally defined users, and remote users that are authenticated and authorized by AAA.

Users must be members of one or more groups. Each group has a role assigned to it which controls the level of access that group members have to the system. These roles are:

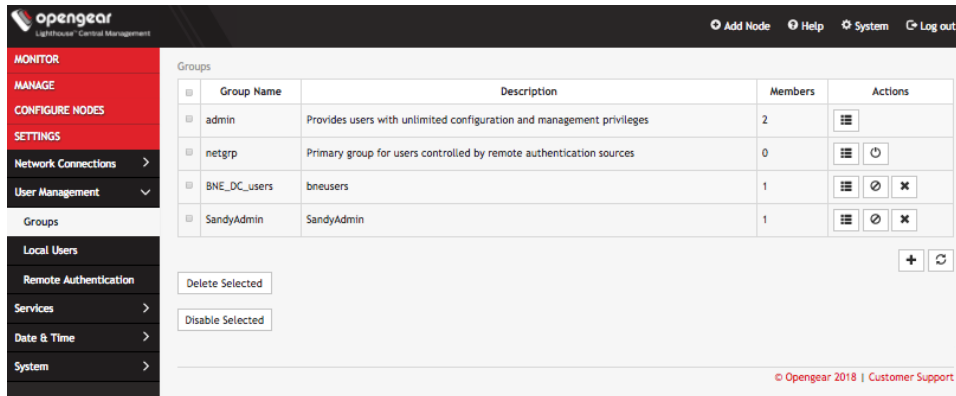
Role	Description
Lighthouse Administrator	The Lighthouse Administrator role is assigned to groups whose members need to manage and maintain the Lighthouse appliance. Members have access to all data on the Lighthouse system
Node Administrator	The Node Administrator role is assigned to groups that need to manage and maintain a set of Nodes. Each group with the Node Administrator role must have an associated Smart Group which is evaluated to define the set of nodes that the group members have access to.
Node User	The Node User role is assigned to groups that need to access a set of nodes. Each group with the Node User role must have an associated Smart Group which is evaluated to define the set of nodes that the group members have access to. Optionally, access to the managed devices can be limited by associating the saved Managed Device Filter with the Node User role.

Group membership can either be defined locally for local users or defined on the AAA server. Groups that are assigned by the AAA servers must still exist locally.

8.1 Password fields in Lighthouse

All password fields in Lighthouse are **write-only**. They accept data from the clipboard or pasteboard but do not pass data out.

8.2 Creating new groups



To create a new group:

1. Select **SETTINGS > User Management > Groups**.
2. Click **+**. The **New Group** dialog opens.
3. Enter a **Group Name**, **Description**, and select a **Role** for the group.

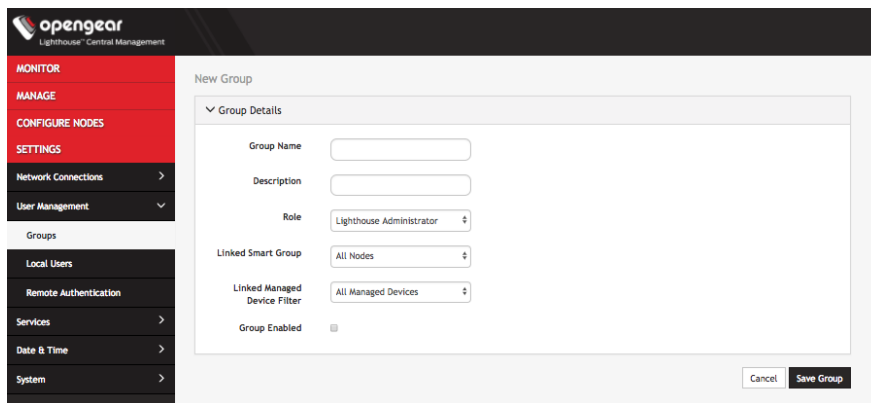
Group Name is case sensitive. It can contain numbers and some alphanumeric characters. When using remote authentication, characters from a user's remote groups that are not allowed on Lighthouse are converted to underscores during authentication. Local groups can be created that take that into account, allowing the authentication to continue.

If the **Role** selected is **Lighthouse Administrator**, members of the group have access to all nodes and managed devices.

If the **Role** selected is **Node Administrator**, select a **Linked Smart Group** to define the nodes that the group has access to. Members of the group have access to all managed devices.

If the **Role** selected is **Node User**, select a **Linked Smart Group** to define the nodes that the group has access to. Choose **All Managed Devices** or a saved managed device filter from **Linked Managed Device Filter** drop-down to define the managed devices that the group has access to.

1. Select **Group Enabled** checkbox to enable group.
2. Click **Save Group**.

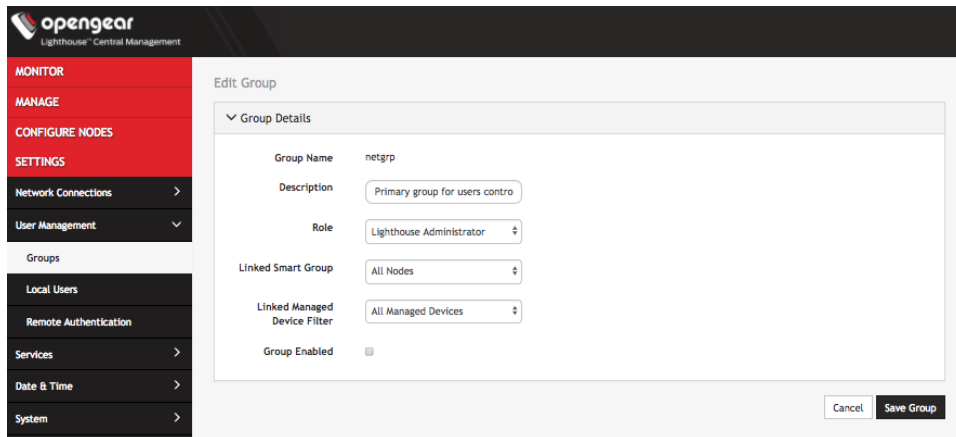


NOTE: When a new group is given the **Lighthouse Administrator** role, members of the group have access to the `sudo` command. Groups or users with the **Lighthouse Administrator** role are added to the **admin** group, which is in the list of allowed sudoers. On first boot of a new Lighthouse instance, the **root** user is the only member of the **admin** group and the only user with `sudo` access.

8.3 Modifying existing groups

To modify an existing group:

1. Select **SETTINGS > User Management > Groups**.
2. Click **Edit** in the **Actions** section of the group to be modified and make desired changes.
3. Click **Save Group**.



The **Modify Group** dialog allows the group's **Description**, **Role**, **Linked Smart Group**, and **Linked Managed Device Filter** to be set and changed.

If a Group's **Role** is **Lighthouse Administrator**, the group's **Linked Smart Group** is **All Nodes** and **Linked Managed Device Filter** is **All Managed Devices**. This cannot be changed. If a Group has a **Linked Smart Group** other than **All Nodes** or a **Linked Managed Device Filter** other than **All Managed Devices**, the group's **Role** cannot be set to **Lighthouse Administrator**.

See *Creating Smart Groups* above for details regarding creating and using Smart Groups and *Creating Managed Device Filters* for details regarding creating and using Managed Device Filters.

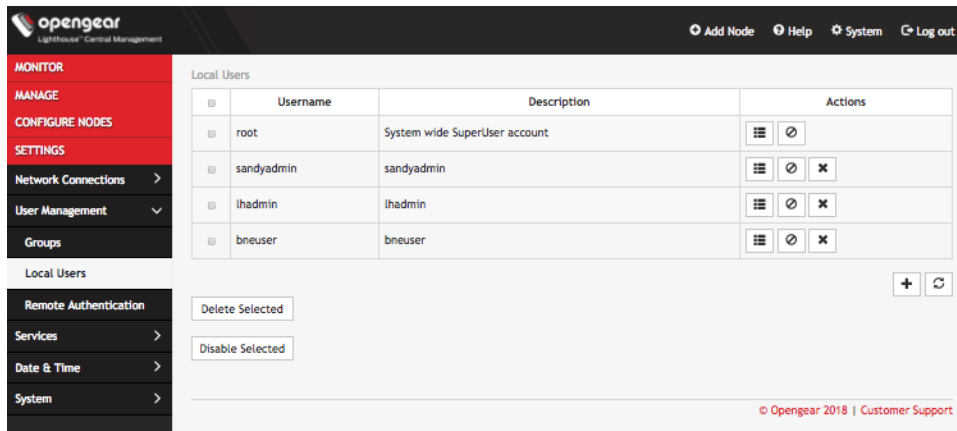
The **Groups** page also allows user to delete a group. All users who were members of the deleted group lose any access and administrative rights inherited from the group.

8.4 A note on default netgrp Lighthouse group

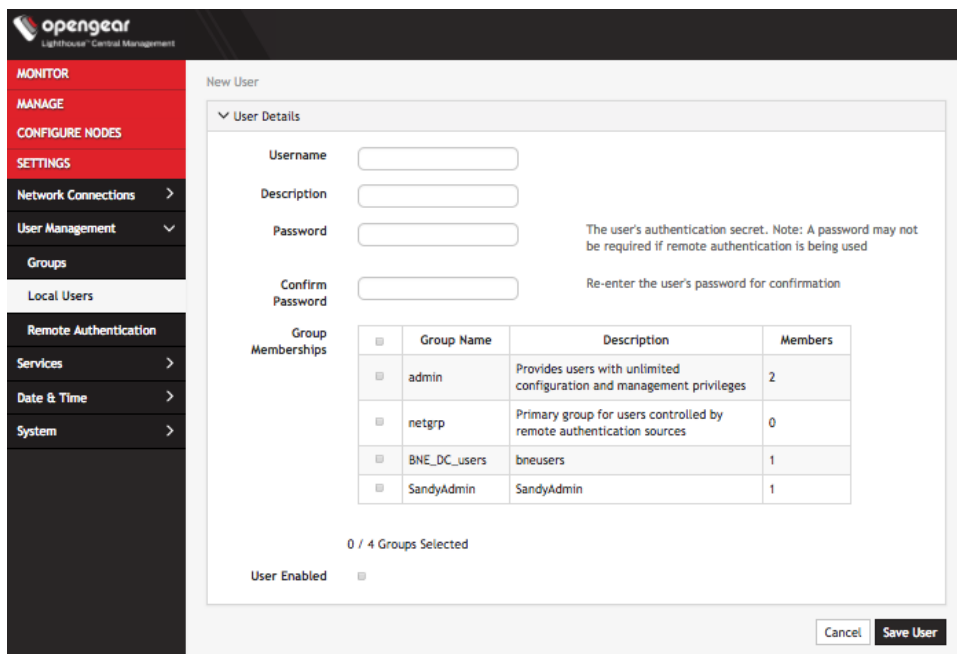
The **netgrp** group is inherited as the primary group for all remote AAA users who are not defined locally on Lighthouse. By default, **netgrp** has the **Lighthouse Administrator** role and is disabled - it must be enabled to take effect for remote AAA users.

8.5 Creating new users

To create a new user:



1. Select **SETTINGS > User management > Local Users**.
By default, the root user is the only user listed.
2. Click the **+** button. The **New User** dialog appears.



3. Enter a **Username**, **Description**, and **Password**.
4. Re-enter the **Password** in the **Confirm Password** field.
5. Select the **Enabled** checkbox.
6. Click **Apply**.

To create a new user without password which causes them to fail back to remote authentication:

1. Select **SETTINGS > User Management > Remote Authentication**
2. Apply Remote Authentication Settings.
3. Select **SETTINGS > User management > Local Users**
4. Click the **+** button. The **New User** dialog loads.

5. Enter a **Username**, **Description**.
6. Select the **Remote Password Only** checkbox.
7. Select the **Enabled** checkbox.
8. Click **Apply**.

NOTE: When a new user is created, an entry is added to the syslog, indicating the new user's name, the user that performed the operation, and the time that it occurred:

```
2018-04-03T12:42:48.587744+00:00 lighthouse configurator_users[28915]: User <newuser>
added to passwords file
2018-04-03T12:42:48.710530+00:00 lighthouse og-rest-api: User <newuser> created by
<root>
```

If the created user is set to disabled, the `configurator_users` message does not appear as they have not been added to the passwords file.

The syslog can be accessed from Lighthouse by clicking **Help > Technical Support Report**.

8.6 Modifying existing users

To modify an existing user:

1. Select **SETTINGS > User management > Local Users**
2. Click **Edit** in the **Actions** section of the user to be modified and make desired changes.
3. Click **Save User**.

The screenshot shows the 'Edit User' dialog in the Lighthouse interface. The left sidebar contains navigation options: MONITOR, MANAGE, CONFIGURE NODES, SETTINGS, Network Connections, User Management (selected), Groups, Local Users, Remote Authentication, Services, Date & Time, and System. The main content area is titled 'Edit User' and contains the following fields and table:

- Username:** sandyadmin
- Description:** sandyadmin
- Password:** (empty field)
- Confirm Password:** (empty field)
- Group Memberships:** A table with columns for Group Name, Description, and Members. The 'SandyAdmin' group is selected.
- User Enabled:** A checked checkbox.

Buttons for 'Cancel' and 'Save User' are located at the bottom right of the dialog.

Group Name	Description	Members
admin	Provides users with unlimited configuration and management privileges	2
netgrp	Primary group for users controlled by remote authentication sources	0
BNE_DC_users	bneusers	1
<input checked="" type="checkbox"/> SandyAdmin	SandyAdmin	1

The **Modify Users** dialog allows the user's **Description** to be changed and the user's **Password** to be reset. The username cannot be changed. To disable a user, uncheck the **Enabled** checkbox.

Disabled users cannot login to Lighthouse using either the Web-based interface or via shell-based logins (i.e. `sshusername-disabled@lighthouse-name-or-ip`). The user and the `/home/username-disabled` directory still exist in the Lighthouse VM file system.

8.7 Deleting users

To delete a user:

1. Select **SETTINGS > User management > Local Users**
2. Click **Delete** in the **Actions** section of the user to be modified.
3. Click **Yes** in the **Confirmation** dialog.

8.8 Disabling a Lighthouse root user

To disable a root user:

1. Make sure that another user exists that is in a group that has the **Lighthouse Administrator** role.
2. Select **SETTINGS > User management > Local Users**
3. Click **Disable** in the **Actions** section of the root user.
4. Click **Yes** in the **Confirmation** dialog.

To enable root user back log in with another user exists that is in a group that has the **Lighthouse Administrator** role and click **Enable** in the **Actions** section of the root user.

8.9 Configuring AAA

Lighthouse supports three AAA systems:

- LDAP (Active Directory and OpenLDAP)
- RADIUS
- TACACS+

Authentication works much the same with each, but group membership retrieval varies. The following sections detail the configuration settings for each provider and explain how group membership retrieval works.

To begin, select **SETTINGS > User Management > Remote Authentication**.

8.9.1 LDAP Configuration

Remote Authentication

Settings

Scheme: LDAP

Remote authentication servers	Address	Port (defaults to 389)
	<input type="text"/>	<input type="text"/> - <input type="text"/>

LDAP base DN: The distinguished name of the search base. For example: dc-my-company,dc-com

LDAP bind DN: The distinguished name to bind to the server with. The default is to bind anonymously.

Bind DN password:

Confirm password:

LDAP username attribute: The LDAP attribute that corresponds to the login name of the user (commonly "sAMAccountName" for Active Directory, and "uid" for OpenLDAP).

LDAP group membership attribute: The LDAP attribute that indicates group membership in a user record (commonly "memberOf" for Active Directory, and unused for OpenLDAP).

Ignore referrals: Disregard LDAP referrals to other servers

Apply

1. Select **LDAP** from the **Scheme** drop-down menu.
2. Add the **Address** and optionally the **Port** of the LDAP server to query.
3. Add the **Base DN** that corresponds to the LDAP system being queried.

For example, if a user's distinguished name is **cn=John Doe,dc=Users,dc=ACME,dc=com**, the **Base DN** is **dc=ACME,dc=com**

4. Add the **Bind DN**. This is the distinguished name of a user with privileges on the LDAP system to perform the lookups required for retrieving the username of the users, and a list of the groups they are members of.
5. Add the password for the binding user.
6. Add the **Username Attribute**. This depends on the underlying LDAP system. Use **sAMAccountName** for Active Directory systems, and **uid** for OpenLDAP based systems.
7. Add the **Group Membership Attribute**. This is only needed for Active Directory and is generally **memberOf**.
8. If desired, check **Ignore referrals** option. When checked, LDAP will not follow referrals to other remote authentication servers when logging users in to Lighthouse. If multiple remote authentication servers exist on the network, checking this option may improve login times.

NOTE: Multiple servers can be added. The LDAP subsystem queries them in a round-robin fashion.

8.9.2 RADIUS configuration

To configure RADIUS:

1. Select **SETTINGS > User Management > Remote Authentication**.

Remote Authentication

Settings

Scheme: RADIUS

Remote authentication servers	Address	Port (defaults to 1812)	
	192.168.250.20	1812	- +

Remote accounting servers	Address	Port (defaults to 1812)	
			- +

Server password: *****

Confirm server password: *****

Apply

2. In the **Settings** section, select **RADIUS** from the **Scheme** drop-down menu.
3. Add the **Address** and optionally the **Port** of the RADIUS authentication server to query.
4. Add the **Address** and optionally the **Port** of the RADIUS accounting server to send accounting information to.
5. Add the **Server password**, also known as the RADIUS Secret.

NOTE: Multiple servers can be added. The RADIUS subsystem queries them in a round-robin fashion.

To provide group membership, RADIUS needs to be configured to provide a list of group names via the Framed-Filter-Id attribute. The following configuration snippet shows how this can be configured for FreeRADIUS:

```
operator1 Auth-Type := System
    Framed-Filter-ID = ":group_name=west_coast_admin,east_coast_user:"
```

NOTE: The **Framed-Filter-ID** attribute must be delimited by the colon character.

8.9.3 TACACS+ configuration

To configure TACACS+:

1. Select **SETTINGS > User Management > Remote Authentication**.

Remote Authentication

Settings

Scheme: TACACS+

Remote authentication servers	Address	Port (defaults to 49)
	<input type="text"/>	<input type="text"/> - <input type="text"/>

TACACS+ login method: PAP
The method used to authenticate to the server. Defaults to PAP. To use DES encrypted passwords, select Login.

Server password:

Confirm server password:

TACACS+ service:
The service to authenticate with. This determines which set of attributes are returned by the server. Defaults to "raccess"

Apply

2. Select **TACACS+** from the **Scheme** drop-down menu.
3. Add the **Address** and optionally the **Port** of the TACACS+ authentication server to query.
4. Select the **Login Method**. **PAP** is the default method. However, if the server uses DES-encrypted passwords, select **Login**.
5. Add the **Server password**, also known as the TACACS+ Secret.
6. Add the **Service**. This determines the set of attributes sent back by the TACACS+ server

NOTE: Multiple servers can be added. The TACACS+ subsystem queries them in a round-robin fashion.

To provide group membership, TACACS+ needs to be configured to provide a list of group names This following configuration snippet shows how this can be configured for a tac_plus server:

```
user = operator1 {
    service = raccess {
        groupname = west_coast_admin, east_cost_user
    }
}
```

To do this with Cisco ACS, see [Setting up permissions with Cisco ACS 5 and TACACS+](#) on the Opendgear Help Desk.

9. Lighthouse central configuration

Templates are a centralized way of changing the configuration for enrolled Opendev console server nodes by pushing pre-defined configuration templates to selected nodes. Lighthouse supports the creation and execution of Users and Groups, Authentication and Script templates.

9.1 Creating new users and groups templates

Administrators can access **CONFIGURE NODES > Configuration Templating > Users and Groups Templates** to create, edit, and delete users and groups templates. Each template must contain at least one group.

Each template contains a list of user-defined groups and/or individual users. Each group has a defined role which determines what privileges group members have. User roles are defined by the groups they are a member of.

The available group roles are:

- **Node Administrator** — maps to the administrator role on the nodes.
- **Node User** — maps to the ports user role and the pmsHELL role on the nodes. Ports access can be restricted if required.

To create a new users and groups template:

1. Select **CONFIGURE NODES > Configuration Templating > Users and Groups Templates**.
2. Click the **+** button. The **New Users and Groups Template** dialog loads.

The screenshot shows the OpenGear Lighthouse Central Management interface. On the left is a navigation sidebar with sections: MONITOR, MANAGE, and CONFIGURE NODES. Under CONFIGURE NODES, there are options for Node Enrollment, Edit Tags, Edit Smart Groups, Edit Managed Device Filters, Configuration Templating (expanded), Apply Templates, Authentication Templates, and Script Templates. Below this is a section for 'Users and Groups Templates'. The main content area is titled 'New Users and Groups Template' and contains three sections: 'Template Details' with input fields for Name and Description; 'Set Group List' with a table header for Group Name and Actions, and a '+ ' button; and 'Set User List' with a table header for User Name and Actions, and a '+ ' button. At the bottom, there is a note: 'Note: To push users, the selected nodes need to be running firmware version 4.3.0 or later.' and two buttons: 'Cancel' and 'Save Template'.

3. Enter a **Name** and **Description** for a template in the **Template Details** section.
4. Click the **+** button in the **Set Group List** section to add a new group. The **Group Details** dialog loads.

The 'Group Details' dialog box contains the following fields: 'Group Name' (text input), 'Description' (text input), 'Role' (dropdown menu with 'Node User' selected), 'Restrict accessible Serial Ports' (checkbox checked), and 'Serial Ports range' (text input). A note next to the 'Serial Ports range' field states: 'A serial port number or range of ports. Ranges use the format start-finish (e.g., 1,3-5,8).' The dialog is styled with a light gray background and rounded corners.

Cancel Apply

5. Enter a **Group Name**, a **Description**, and select a **Role** for the group.

6. If **Node User** role is selected, the **Restrict accessible Serial Ports** checkbox and **Serial Ports range** appear.
7. Use the checkbox to restrict access and specify as port or range of ports in the **Serial Ports range** text box.
8. Click **Apply**.
9. Click the **+** button in the **Set User List** section to add new users. The **User Details** dialog loads.

The screenshot shows a 'User Details' dialog box with the following elements:

- Username**: A text input field.
- Description**: A text input field.
- Password**: A text input field.
- Confirm Password**: A text input field.
- Group Memberships**: A section containing a table with columns 'Group Name' and 'Description'. Below the table, it says 'No Groups have been created'.
- 0 / 0 Groups Selected**: A status indicator at the bottom of the dialog.

Cancel Apply

10. Enter a **Username**, a **Description**, and a **Password** for the user. Type the password again in the **Confirm Password** text box.
11. Optionally, click checkboxes next to the groups this user should belong to. Only groups from this template are available.
12. Click **Apply**.
13. Continue adding new groups and users until finished.
14. Click **Save Template**.

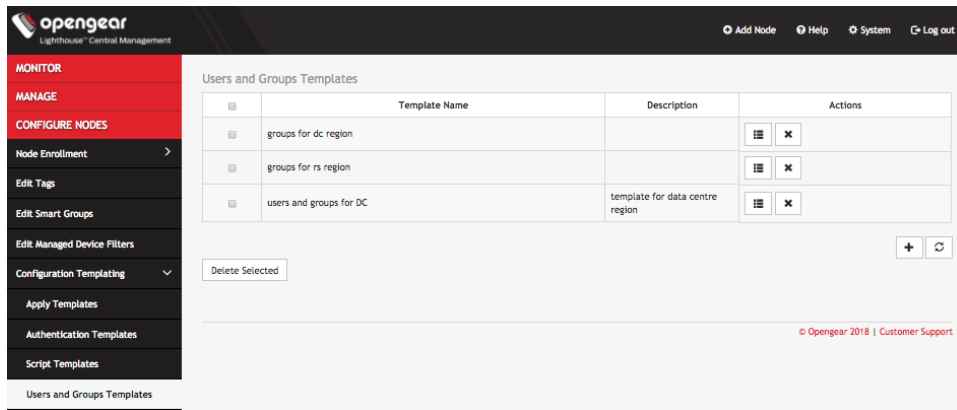
NOTE: When a **users and groups template** is pushed to a node, all custom groups on that node are replaced by groups defined in the template. If no users are in the new template, existing users will remain on the node. To push users, the selected nodes need to be running firmware version 4.3.0 or later.

9.2 Modifying existing users and groups templates

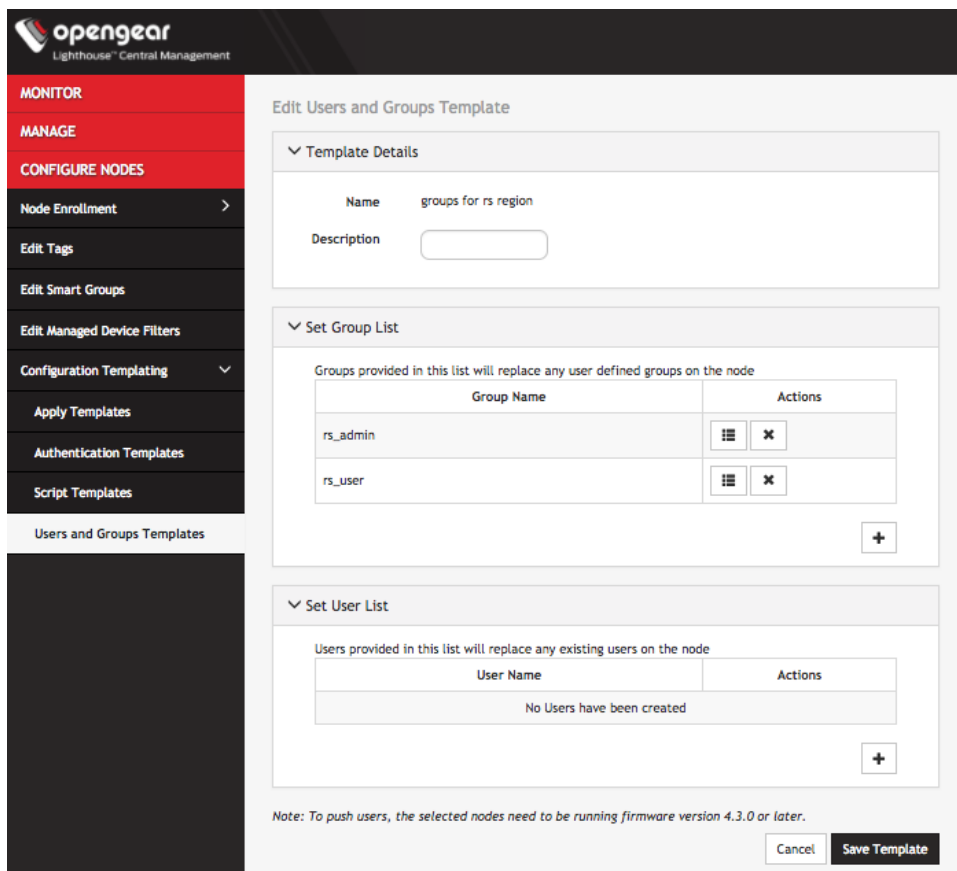
The **Edit Users and Groups Template** dialog allows a template's **Description**, **Group List**, and **User List** to be set and changed.

To modify a template:

1. Select **CONFIGURE NODES > Configuration Templating > Users and Groups Templates**.



2. Click **Edit** in the **Actions** section of the template to be modified. The **Edit Users and Groups Template** dialog appears.



3. Make changes to the template's details, group list, or Individual user list as required.
4. Click the X button under Actions next to any groups or users which need to be removed.
5. Click **Save Template**.

9.3 Deleting users or groups from a template

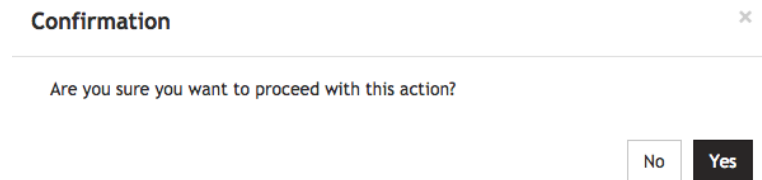
To delete a template:

1. Select **CONFIGURE NODES > Configuration Templating > Users and Groups Templates**.
2. Click the Edit button in the **Actions** section of the template.
3. Click the X button under Actions next to any groups or users which need to be removed.
4. Click **Save Template** to save the changes.

9.4 Deleting users and groups templates

To delete a template:

1. Select **CONFIGURE NODES > Configuration Templating > Users and Groups Templates**.
2. Click the X button in the **Actions** section of the template to be removed. The **Confirmation** alert box appears.



3. Click **Yes** in the **Confirmation** dialog. The users and groups template is deleted.

9.5 Creating new authentication templates

Only users assigned to the **Lighthouse Administrator** role can access **CONFIGURE NODES > Configuration Templating > Authentication Templates** and create authentication templates.

The supported modes are **Local**, **Radius**, **TACACS+**, and **LDAP**. For example, if an authentication template is configured to use **RADIUS** as an authentication source, that corresponds to **RADIUSDownLocal** with **Use Remote Groups** ticked on the downstream node.

To create a new authentication template:

1. Select **CONFIGURE NODES > Configuration Templating > Authentication Templates**.
2. Click the **+** button. The **New Authentication Template** dialog loads.

The screenshot shows the 'New Authentication Template' dialog in the OpenGear Lighthouse Central Management console. The left sidebar contains navigation options: MONITOR, MANAGE, CONFIGURE NODES, Node Enrollment, Edit Tags, Edit Smart Groups, Edit Managed Device Filters, Configuration Templating, Apply Templates, Authentication Templates, Script Templates, and Users and Groups Templates. The main content area is titled 'New Authentication Template' and contains two sections: 'Template Details' with 'Name' and 'Description' input fields, and 'Authentication Settings' with a 'Pre-populate from Lighthouse' button, a 'Scheme' dropdown menu (set to 'Local users only'), and a descriptive text: 'Pre-populate the template fields with the current Lighthouse remote authentication settings.' At the bottom right of the dialog are 'Cancel' and 'Save Template' buttons.

3. Enter a **Name** and **Description** for a template in the **Template Details** section.
4. Select a desired Scheme or click **Pre-populate** to pre-populate a template with the current Lighthouse remote authentication configuration.
5. Enter or update authentication settings if required. See *Configuring AAA* above for an example.
6. Click **Save Template**.

NOTE: When an authentication template is pushed to a node, the authentication settings at that node are replaced by the those defined in the authentication template.

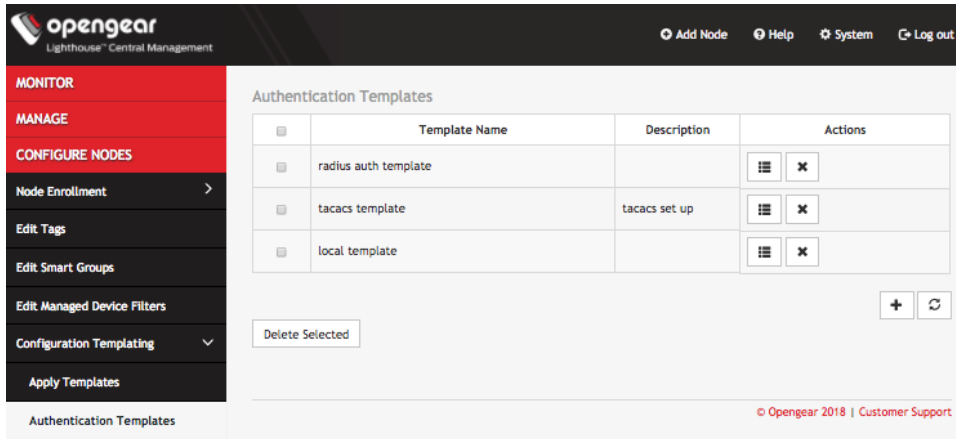
NOTE: The authentication templates do not support the full list of settings that the OpenGear console servers support. However, templates can be applied, and then additional settings configured manually.

9.6 Modifying existing authentication templates

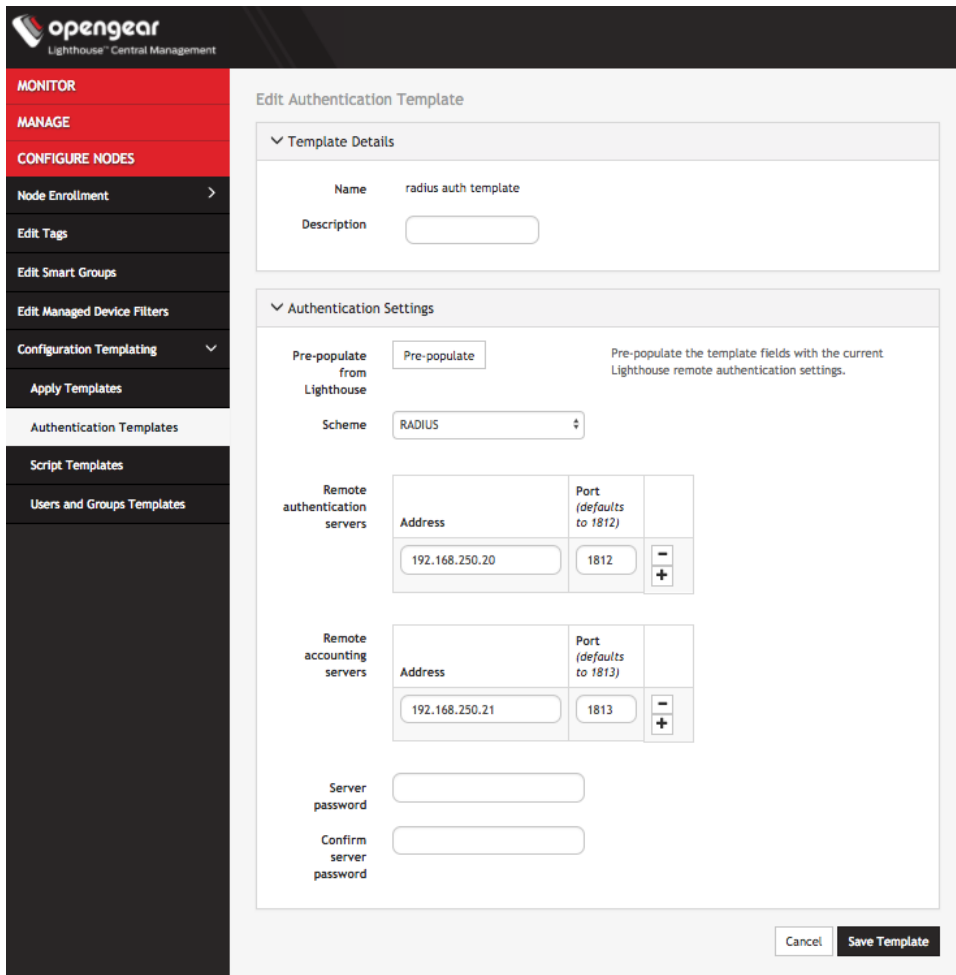
The **Edit Authentication Template** dialog allows the template's **Description** and **Authentication Settings** to be set and changed.

To modify an existing authentication template:

1. Select **CONFIGURE NODES > Configuration Templating > Authentication Templates**.



2. Click **Edit** in the **Actions** section of the template to be modified. The **Edit Authentication Template** dialog appears.



5. Make required changes.
6. Click **Save Template**.

9.7 Deleting authentication templates

To delete an authentication template:

1. Select **CONFIGURE NODES > Configuration Templating > Authentication Templates**.
2. Click **Delete** in the **Actions** section of the template to be removed. The **Confirmation** alert box appears.



3. Click **Yes** in the **Confirmation** dialog. The authentication template is deleted.

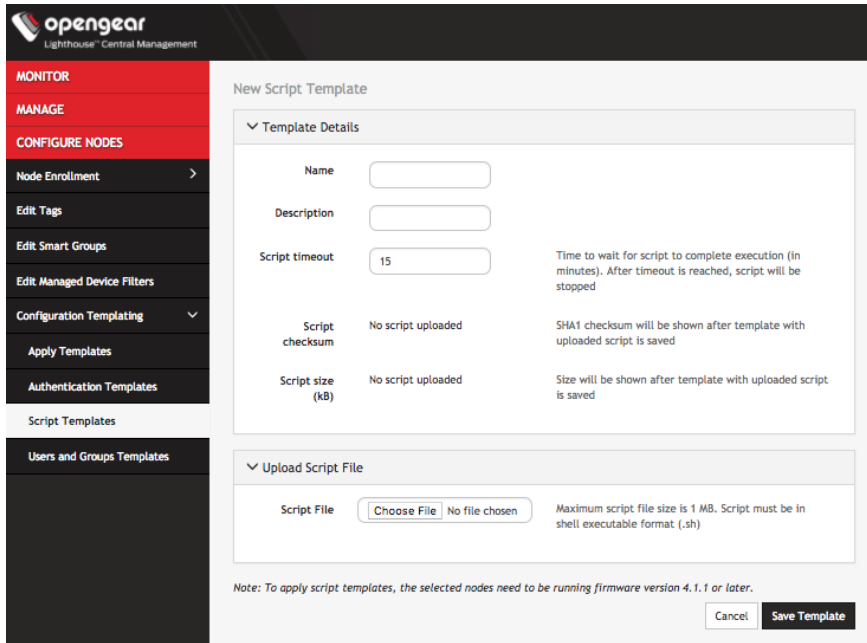
9.8 Creating new script templates

Users assigned to the **Lighthouse Administrator** role can access **CONFIGURE NODES > Configuration Templating > Script Templates** and create script templates.

Script Templates allow the user to upload arbitrary shell scripts to be run on a node. A script may set additional configuration settings not available in other templates or store additional files onto the node such as certificates, for example. The uploaded script must have a `.sh` extension and can't be more than 1MB in size. Other than those, there are no other restrictions on the script file to be uploaded. Once saved, the template stores the size and SHA1 checksum of the script. This can be used to verify the script contents of the template once saved. To apply script templates, the selected nodes need to be running firmware version 4.1.1 or later.

To create a new script template:

1. Select **CONFIGURE NODES > Configuration Templating > Script Templates**.
2. Click the **+** button. The **New Script Template** dialog loads.

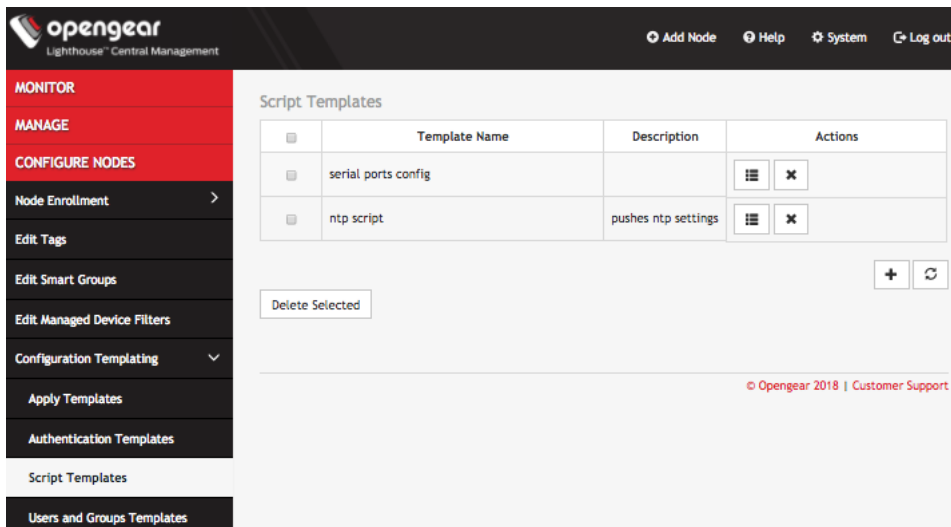


3. Enter a **Name** and **Description** for a template in the **Template Details** section.
4. To select a script to upload, click **Choose file**.
5. Click **Save Template**. **Script checksum** and **Script size** are shown after template with uploaded script is saved.

9.9 Modifying existing script templates

The **Edit Script Template** dialog allows the template's **Description**, **Script timeout**, and **Script File** to be uploaded. To modify an existing script template:

1. Select **CONFIGURE NODES > Configuration Templating > Script Templates**.



2. Click **Edit** in the **Actions** section of the template to be modified. The **Edit Script Template** dialog appears.

The screenshot shows the 'Edit Script Template' dialog in the OpenGear Lighthouse Central Management interface. The dialog is titled 'Edit Script Template' and has a sidebar on the left with navigation options: MONITOR, MANAGE, CONFIGURE NODES, Node Enrollment, Edit Tags, Edit Smart Groups, Edit Managed Device Filters, Configuration Templating, Apply Templates, Authentication Templates, Script Templates, and Users and Groups Templates. The main content area is divided into two sections: 'Template Details' and 'Upload Script File'. The 'Template Details' section contains a table with the following information:

Name	serial ports config	
Description	<input type="text"/>	
Script timeout	<input type="text" value="5"/>	Time to wait for script to complete execution (in minutes). After timeout is reached, script will be stopped
Script checksum	b4fe78ebcb12b104fdc05 153909201c8f72908e9	SHA1 checksum will be shown after template with uploaded script is saved
Script size (kB)	867 bytes	Size will be shown after template with uploaded script is saved

The 'Upload Script File' section contains a 'Script File' field with a 'Choose File' button and the text 'No file chosen'. To the right of this field, it says 'Maximum script file size is 1 MB. Script must be in shell executable format (.sh)'. At the bottom of the dialog, there is a note: 'Note: To apply script templates, the selected nodes need to be running firmware version 4.1.1 or later.' and two buttons: 'Cancel' and 'Save Template'.

3. Make required changes.
4. Click **Save Template**.

9.10 Deleting script templates

To delete a script template completely:

1. Select **CONFIGURE NODES > Configuration Templating > Script Templates**.
2. Click **Delete** in the **Actions** section of the template to be removed. The **Confirmation** alert box appears.

Confirmation

Are you sure you want to proceed with this action?

3. Click **Yes** in the **Confirmation** dialog. The script template is deleted.

9.11 Apply Templates

Users with **Lighthouse Administrator** privileges (i.e. users with the **Lighthouse Administrator** role or users who are members of groups with the **Lighthouse Administrator** role) can access **CONFIGURE NODES > Configuration Templating > Apply Templates** and execute templates affecting any node.

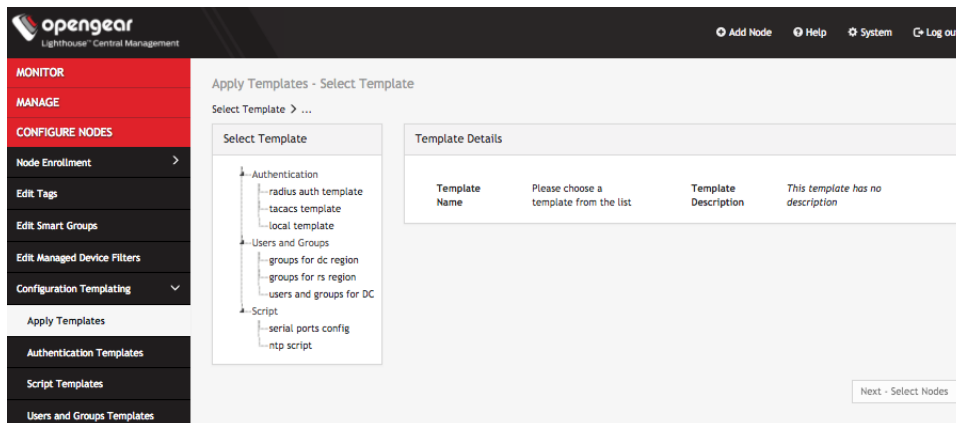
Users with Node Administrator privileges (i.e. users with the Node Administrator role or users who are members of groups with the Node Administrator role) can access **CONFIGURE NODES > Configuration Templating > Apply Templates** and execute templates affecting nodes in Smart Groups linked to their role.

Apply Templates consists of four stages, each one a step in the overall wizard. The steps are:

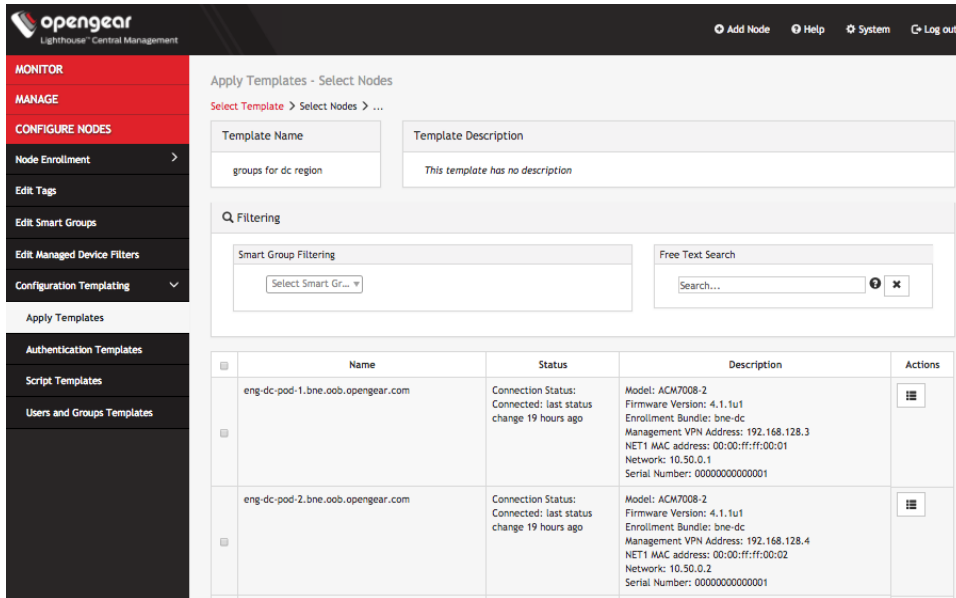
1. Select Template.
2. Select Nodes.
3. Preflight. This test run simulates what happens if the template is pushed to the selected nodes.
4. Execution.

To apply a template:

1. Select **CONFIGURE NODES > Configuration Templating > Apply Templates**.



2. Select a template from the existing template tree. **Template Details** populates with details from the selected template.
3. Click **Next — Select Nodes**. The **Select Nodes** stage loads.



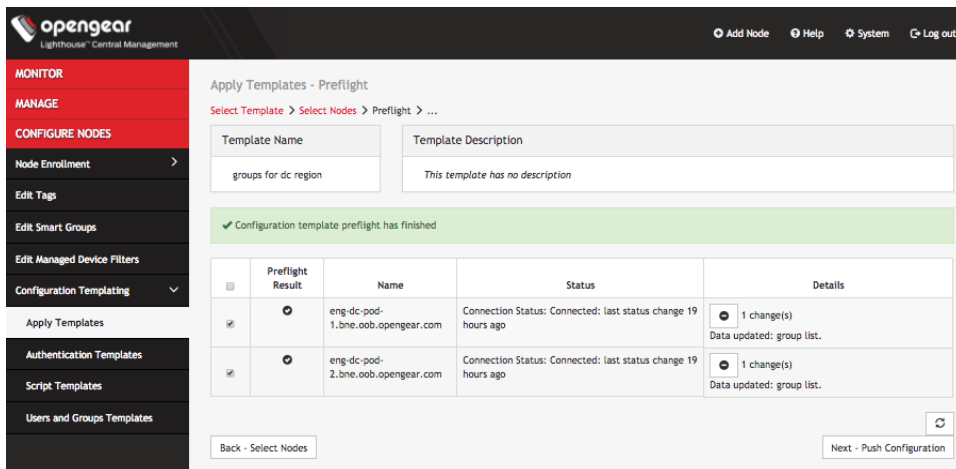
4. Select nodes from the list of enrolled nodes. **Smart Group Filtering** and **Free Text Search Filtering** can be used to narrow down the results.

The screenshot above shows filtering being used to set the list of enrolled nodes to match the set of nodes an administrator wishes to deal with.

NOTE: Third-party nodes are not supported for template execution.

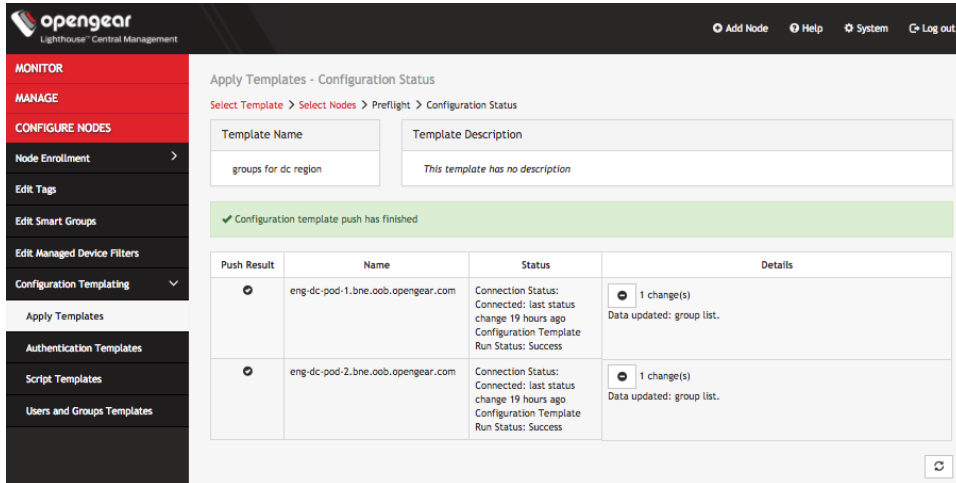
5. Click **Next — Preflight**. The **Preflight** stage loads. This stage requires manual refresh to retrieve updated **Preflight Result** and **Details**.

After all nodes finish preflight, a success message appears and **Next — Push Configuration** becomes active.



6. Select desired nodes for template execution and click **Next — Push Configuration**. The **Configuration Status** stage loads. This stage requires manual refresh to retrieve updated **Push Result** and **Details**.

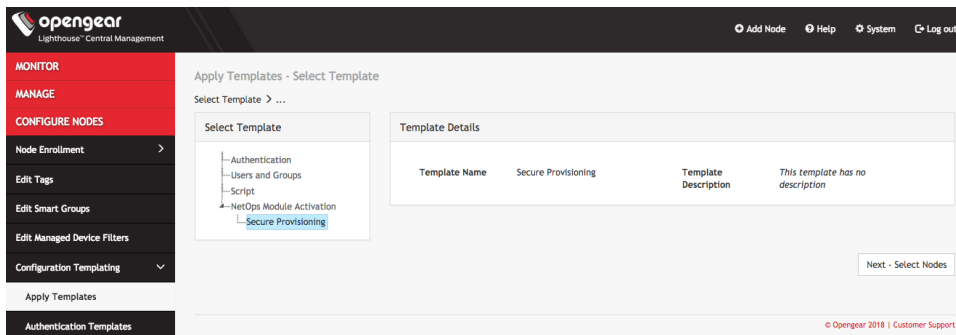
After all nodes finish the template push a success message appears.



9.12 Manually Activate Secure Provisioning via Template

Users assigned to the **Lighthouse Administrator** role can manually apply the **Secure Provisioning NetOps Module** to desired OM2200 nodes.

1. As a Lighthouse administrator, choose **CONFIGURE NODES > Configuration Templating > Apply Templates**
2. Click **Secure Provisioning** under **NetOps Module Activation**.



3. Click **Next – Select Nodes**
4. Choose the desired OM2200 nodes by clicking the checkboxes next to them.
5. Click **Next – Preflight**. Refresh to ensure the preflight check has succeeded.
6. When preflight is complete, click **Next - Push Configuration**.

10. NetOps Automation

Opengear NetOps Automation is a system that automates the configuration and operation of network infrastructure in data center and remote edge locations. This chapter explains how to configure and use **Provisioning for NetOps Automation**.

The system is composed of these parts:

- **Opengear Lighthouse software** - The central management interface and automation controller.
- **Managed Devices** - Target infrastructure, e.g. network switches from Cisco or Cumulus
- **Opengear OM2200 Nodes** - Remote management appliances, connected, to managed device management NICs via Ethernet (and optionally managed device consoles via RS-232 or USB serial).
- **NetOps Modules** - Software components that automate specific operational scenarios (e.g. network device provisioning), deployed as Docker containers. Modules become active on Lighthouse when a license for that module is installed. They can then be manually or automatically activated on OM2200 nodes.

This chapter assumes that you are using Lighthouse version 5.2.2 or later, and an OM2200 appliance running firmware 18.Q3.0 or later.

10.1 Secure Provisioning for NetOps Automation

The Secure Provisioning module for NetOps Automation allows you to provision a new network remotely, securely, and automatically. Using Secure Provisioning for NetOps Automation, network turn up no longer requires network engineering staff to perform initial configuration tasks on site even when there is no existing LAN or WAN in place. Remote hands rack, stack, and cable the infrastructure, then Secure Provisioning for NetOps Automation automates the rest of the turn up process.

The Provisioning module leverages these technologies:

- **ZTP (Zero Touch Provisioning)**: The process by which managed devices in their unconfigured state request and are delivered initial setup resources over the local management network.
- **Human-readable YAML language**: Provides simplified configuration of managed device ZTP configuration parameters.
- **Git source control**: Managed Device resources such as initial configuration files and OS images are automatically stored in a versioned, auditable repository.
- **Ansible automation framework**: Automatically propagates device resources and configures on-site ZTP services.

The Secure Provisioning module combines a centrally orchestrated, vendor-neutral ZTP service with on-site node LAN and WAN connectivity, to automate the provisioning process end to end.

NOTE: This chapter diverges for UI and CLI-based workflows.

10.2 Initial Setup

Import the supplied Lighthouse virtual appliance into your VMware, VirtualBox or Linux KVM hypervisor.

The latest NetOps modules are included with the Lighthouse release in a separate virtual disk. To use a module in Lighthouse, the disk needs to be added to the VM instance as a second IDE disk. If you are installing a fresh Lighthouse instance based on VMX, OVF/OVA, or Hyper-V, the NetOps disk will already be included in the archive and VM configuration. Boxes with Fedora or CentOS as host currently do not support adding multiple disks.

10.2.1 Manually install NetOps virtual disk

For fresh installs based on raw HDD files (QEMU, Google Compute Engine), or for upgrading an existing Lighthouse install, the NetOps disk must be manually added following these steps:

VMware vSphere 6.0 Client

1. Make sure the Lighthouse virtual machine is powered off.
2. Right-click the Lighthouse virtual machine and select **Edit Settings**.
3. On the **Virtual Hardware** tab, select **New Hard Disk** from the **New Device** drop-down and click **Add**.
4. Choose the **Opengear_Lighthouse_VM-disk2.vmdk** file.

VMware Workstation Pro and VMware Workstation Player

1. Make sure the Lighthouse virtual machine is powered off.
2. Right-click the Lighthouse virtual machine and select *Settings*.
3. Click **Add** at the bottom, then select **Hard Disk** and click **Next**.
4. Choose **SCSI** and click **Next**.
5. Choose **Use an existing virtual disk** and click **Next**.
6. Choose the **Opengear_Lighthouse_VM-disk2.vmdk** file and click **Finish**.

Hyper-V using Hyper-V Manager

1. Make sure the Lighthouse virtual machine is powered off.
2. Right-click the Lighthouse virtual machine and select **Settings**.
3. Select the disk controller that the primary disk is on (IDE or SCI). In the right pane, select **Hard Drive** and click **Add**.
4. Click **Browse** and choose the **Ironman-nom.vhd** disk and click **Apply**.

Hyper-V using Powershell

Use the `Add-VMHardDiskDrive` cmdlet, e.g:

```
# Add-VMHardDiskDrive -VMName lighthouse -ControllerType IDE -
ControllerNumber 0 -ControllerLocation 1 -Path 'C:\LocalVMs\Virtual Hard
Disks\Ironman-nom.vhd'
```

where:

- *lighthouse* is the name of the virtual machine
- *IDE* is the disk controller type
- *C:\LocalVMs\Virtual Hard Disks\Ironman-nom.vhd* is the path to the NetOps disk image

Virtualbox

1. Make sure the Lighthouse virtual machine is powered off.
2. Extract the **Opengear_Lighthouse_VM-disk2.vmdk** file from the `lighthouse-5.2.2-ovf.zip` archive.
3. Select the Lighthouse instance and click **Settings**.

4. In the Storage tab, select the **main disk controller (SCSI)** and click the **Add Hard Disk** button.
5. Click **Choose Existing** and select the `Opengear_Lighthouse_VM-disk2.vmdk` file.
6. Click **OK**.

Virtual Machine Manager

1. Make sure the Lighthouse virtual machine is powered off.
2. Extract the raw NetOps `lh_hdd` disk from `lighthouse-5.2.2-raw-hdd-nom.tar`.
3. Select the Lighthouse instance and click **Open**.
4. Click the **Information** button at the top and then **Add Hardware** at the bottom.
5. Make sure **Storage** is selected, then choose **Select or create custom storage**.
6. Click **Manage** then **Browse Local**. Find the extracted `lh_hdd` disk. Make sure the **Disk Bus** is set to **IDE**. Click **Finish**.
7. Make sure that the **Storage format** under **Advanced Options** is set to **raw**.

Google Compute Engine

1. Once the Lighthouse virtual machine is created, make sure it is powered off.
2. To create an image from the `lighthouse-gce-nom.tar.gz` file, follow the instructions here: https://cloud.google.com/compute/docs/images/import-existing-image#import_image.
3. In the **Google Cloud Console**, navigate to **VM Instances**, then select the Lighthouse instance to open its details and click **Edit**.
4. Under **Additional Disks** click **Add Item**, then create a new disk based on the newly created **nom** image.

Install the OM2200

1. Connect the NET1 Ethernet to a network port to a network port via which OM2200 node can reach the Lighthouse VM.
2. Connect power to one or both of the OM2200's AC inlets.
3. By default, OM2200 node requests a DHCP address and has a static address of 192.168.7.2/24. Confirm that you can reach the OM2200's address via ping, SSH and HTTPS.

Enroll the OM2200 as a Lighthouse node

1. Launch an HTTPS browser session to Lighthouse with a root or administrator account.
2. On the top right of the screen, click **Add Node**.
3. Select **An Opengear device**, the second option in the Product dropdown list.

4. Enter the OM2200's **Network Address**, **Username (root)** and **Password (default)**.
5. Check **Auto-approve node** then click **Apply**.

6. Select **CONFIGURE NODES > Node Enrollment > Enrolled Nodes** then click the Refresh button on the bottom of the page to confirm that enrollment has completed.

The OM2200 now has a secure Lighthouse VPN (OpenVPN) tunnel back to Lighthouse.

NOTE: For automatic deployment, activate the Secure Provisioning module with a license (see Chapter 7.1). Then either:

- Check **Always Activate** on the **CONFIGURE NODES > NetOps Modules > Manage Modules** page.
- Or create an enrollment bundle with the Secure Provisioning module assigned to it and enroll using this bundle.

When activation completes, the **CONFIGURE NODES > Device Provisioning** menu becomes available. You can now securely provision managed devices.

For manual deployment, see Chapter 9.12, *Manually Activate Secure Provisioning via Template*.

10.2.2 Connect target device

Secure Provisioning currently supports provisioning devices from these vendors:

- Opendgear
- Cisco
- Juniper
- Arista
- HPE/Aruba
- Huawei
- Cumulus
- Pica8

NOTE: Additional devices may be supported using custom DHCP configuration. To request built-in support for additional devices, visit opengear.com/support.

1. Connect a supported managed device's management NIC directly to the OM2200's NET2 Ethernet, any port of its built-in Ethernet switch, or via an intermediary management switch.
2. Power on the managed device.
3. Ensure the managed device is in ZTP mode, this typically requires the device to have its configuration erased/reset to factory defaults.

10.3 Device Provisioning configuration

All system configuration is performed via Lighthouse. The configuration necessary to provision a device consists of two elements, the **Device Resource Bundle** and **Node Inventory**.

10.3.1 Device Resource Bundle

A **Device Resource Bundle** contains the resource files, such as a configuration file, and OS upgrade image. These are loaded onto the managed device via ZTP (DHCP + TFTP). This may be a full, final

configuration, or a baseline configuration to allow the device to be managed by an upstream configuration service.

As each vendor's ZTP process is slightly different, Device Resource Bundles allow you to select the Device Type. This generates the appropriate ZTP server configuration (DHCP options), any necessary intermediary provisioning scripts and enables device-specific ZTP features, such as serial number matching.

By default, Device Resource Bundles are targeted to all managed devices of the selected Device Type. Bundles may be targeted to specific managed devices by specifying one or more device MAC addresses (including range and reverse match), or in some case by specifying one or more device serial numbers.

10.3.2 Node Inventory

A **Node Inventory** is a static or dynamic list of nodes and a corresponding list of Device Resource Bundles. This defines how Device Resource Bundles are distributed around your network.

You may skip this step and distribute a bundle to all nodes, e.g. if you have generic baseline configurations across all managed devices of a specific type, or if you're leveraging MAC or serial matching bundle-by- bundle.

Resource Bundles may be distributed using any of these three methods:

1. Push to all nodes
2. Push to a static list of nodes, selected individually by node ID
3. Push to a dynamic list of nodes, linked to a Lighthouse Smart Group of nodes

You may mix distribution methods, e.g. a set of basic resources shared across all your network sites, plus a set of site-specific resources distributed to certain nodes only.

10.3.3 Create device configuration

To provision a managed device, you must supply device resources. Device resources consist of an initial configuration file for the device to install, and optionally an operating system image for the device to upgrade itself with.

Device resource file formats are specific to the target vendor. Provisioning for NetOps Automation provisions these files but does not generate them. For example, a simple Arista initial configuration file may look like:

demo_arista.cfg

```
hostname nom-demo-switch
!
interface Management1
  description ZTP_Mgmt_Interface
  ip address 10.0.0.123/24
!
banner login
Welcome to $(hostname)!
```

```
      / |
     \| \|
( ) \ `---. Provisioned by
( )  _ | |   Opengear NetOps Automation
( )  | |
( )  _ . | _ |

EOF
!
end
```

A simple Cisco IOS XR initial configuration may look like:

```
!! IOS XR
!
hostname nom-demo-router
!
username admin
  group root-lr
  group cisco-support
  secret 5 $1$Qk9Y$x/GCXsUPrXYQw1s5GCdW30
!
interface MgmtEth0/RP0/CPU0/0
  description ZTP_Mgmt_Interface
  ip address 10.0.0.200 255.255.255.0
!
banner motd ^Welcome to $(hostname)!
```

```
      / |
     \| \|
( ) \ `---. Provisioned by
( )  _ | |   Opengear NetOps Automation
( )  | |
( )  _ . | _ |
^
!
end
```

10.3.4 Using templated resources

As well as static files, you may create templated ZTP script files. In certain cases, these script files may need to reference site-specific values such as the IP address of the remote node's TFTP server, which may vary from node to node.

In most cases, Secure Provisioning automatically generates and serves any intermediate scripts required by the vendor's ZTP process, to load the managed device initial configuration.

For *Pica8* and *Cumulus* devices, initial configuration is performed directly by this script, so it is not auto-generated.

In creating this script, you may use the templated `{{ nom_remote_server }}` variable replaced by ZTP server address of the OM2200 node.

cumulus_setup.sh

```
#!/bin/bash
curl tftp://{{nom_remote_server}}/cumulus_interfaces >
/etc/network/interfaces
```

Templated resources are referred to as **Script Files (UI)** or **script_files (CLI)** whereas static resources are referred to as **Configuration Files (UI)** or **config_files (CLI)**.

10.4 UI-based workflow

Each NetOps Module provides a simple web UI for configuration and status monitoring. This UI is designed primarily for manual operation, evaluation and testing. For comprehensive automation, refer to the CLI-based workflow section below.

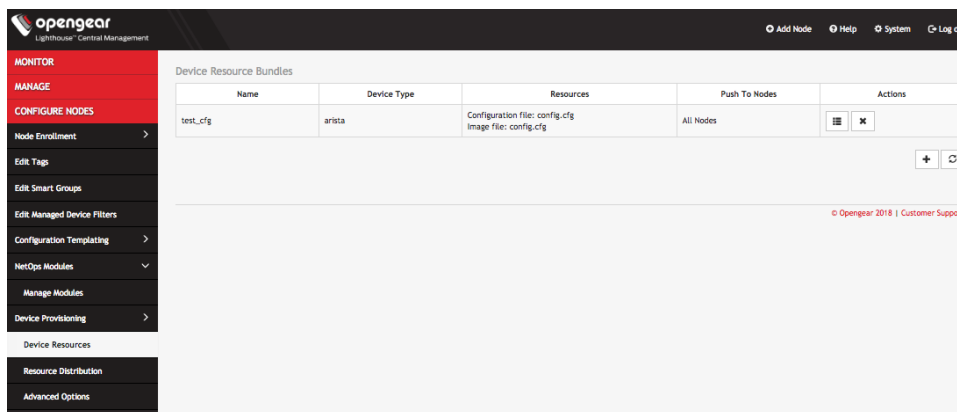
To begin:

1. Launch an HTTPS browser session to Lighthouse
2. Login as a Node Administrator or Administrator user.

NOTE: Lighthouse Administrators and root can push resources, but it is not recommended. We recommend that Node Administrator users upload and push resources via the UI and push them to nodes they administrate using either static or dynamic inventories. Users will only see resources they have uploaded.

10.4.1 Create Device Resource Bundle

1. From the menu, select **CONFIGURE NODES > Device Provisioning > Device Resources**



2. Click the **Add** icon

New Device Resource Bundle

Device Resource Details

Name

Device Type

Configuration File Upload a device configuration or select from uploaded files.
 No file chosen

Image File Upload a device firmware/OS image file or select from uploaded files.
 No file chosen

MAC Addresses - + Provision devices matching the specified MAC address(es). Specify each address in full, using a wildcard (e.g. 00:10:FA:C2:BF:*), or negate to exclude from the match (e.g. !01:23:45:67:89:AB).

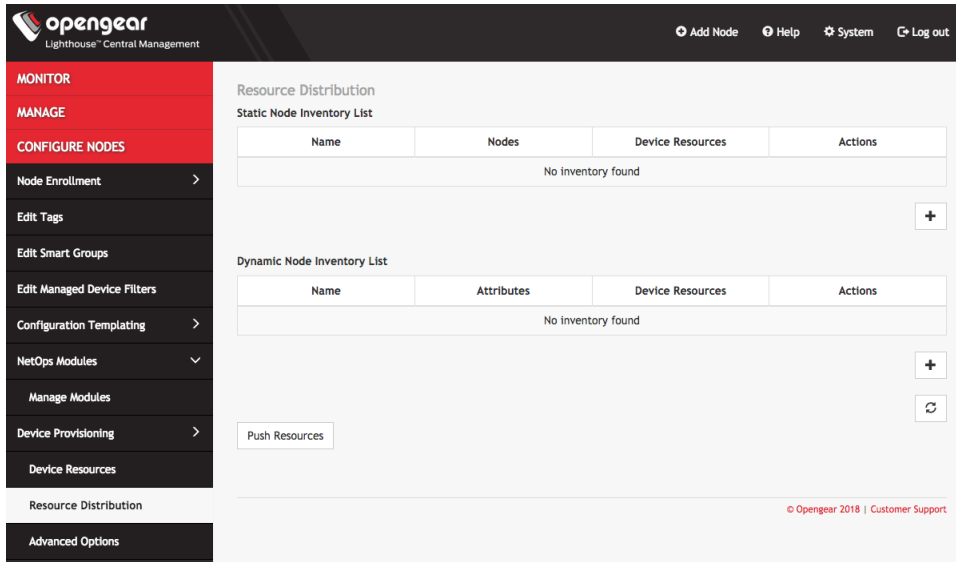
Push to all nodes

3. Choose a **Name** to identify this bundle, e.g. *demo_arista*
4. Select the **Device Type** corresponding to a target managed device
5. Each device type may display different resource fields documented below, this is dependent on the device-specific ZTP feature set
 - **Configuration File** is the initial configuration file for the device to load via ZTP, select a previously uploaded file or click the button to upload a new file
 - **Image Files** is the initial software image for the device to load via ZTP, select a previously uploaded file or click the button to upload a new file
 - **Script File** is used by devices required templated resources, i.e. Cumulus and Pica8 – see earlier note about templated resources, select a previously uploaded file or click the button to upload a new file. Optionally target this bundle at devices matching the specified **MAC Addresses**, click the **Add** and **Remove** icons when specifying multiple
6. Each MAC address may be specified in full using a wildcard (e.g. 00:10:FA:C2:BF:*), or negated to exclude from the match (e.g. !01:23:45:67:89:AB)
7. Optionally target this bundle at devices matching the specified **Serial Numbers**. Click the **Add** and **Remove** icons when specifying multiple
8. Optionally, check **Push to all nodes** to distribute this bundle to all nodes. If you select this option for all bundles, you may skip the *Define Resource Distribution* step
9. Click **Save**

10.4.2 Define Resource Distribution

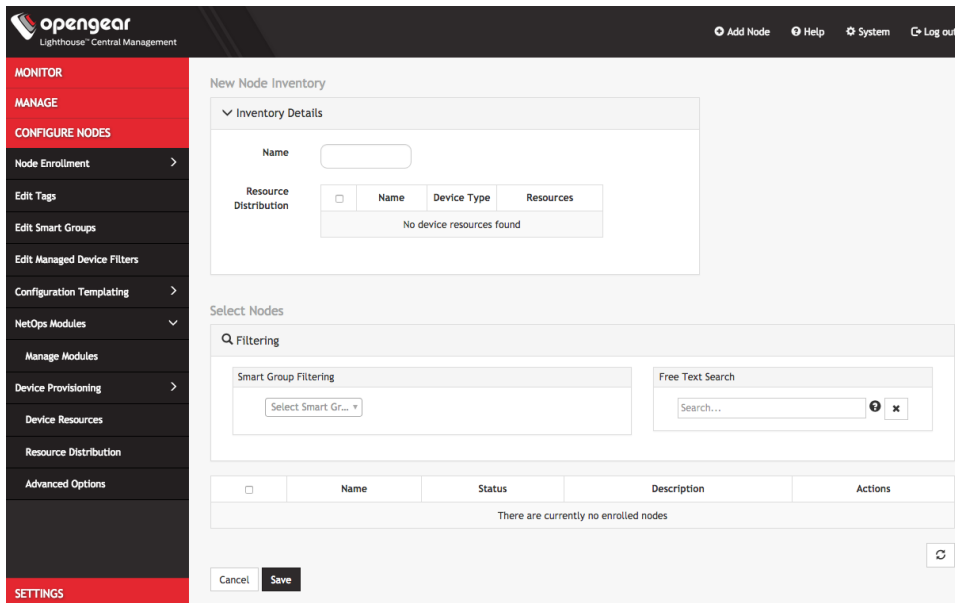
See the *Node Inventory* section for an overview of available distribution methods.

From the menu, select **CONFIGURE NODES > Device Provisioning > Resource Distribution**



To define a **Static Node Inventory**:

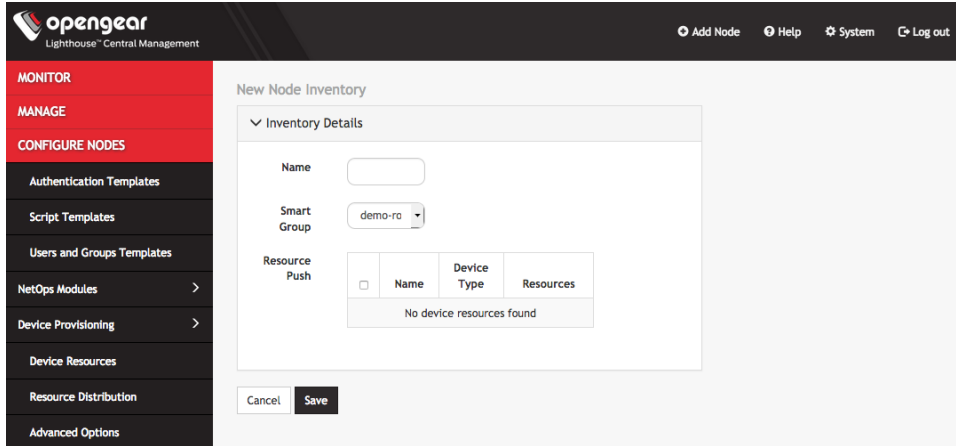
1. Choose **CONFIGURE NODES > Device Provisioning > Resource Distribution**. Under the **Static Node Inventory List** click the **Add** icon



2. Choose a **Name** to identify this bundle, e.g. *BranchInventory*
3. Select the bundles to distribute, as listed in the **Resource Distribution** table
4. **Select Nodes** to which these bundles are to be distributed
5. The **Free Text Search** and **Smart Groups** options help you locate nodes for inclusion in the static inventory. These filters are not applied dynamically to the inventory going forward
6. Click **Save**

To define a **Dynamic Node Inventory**:

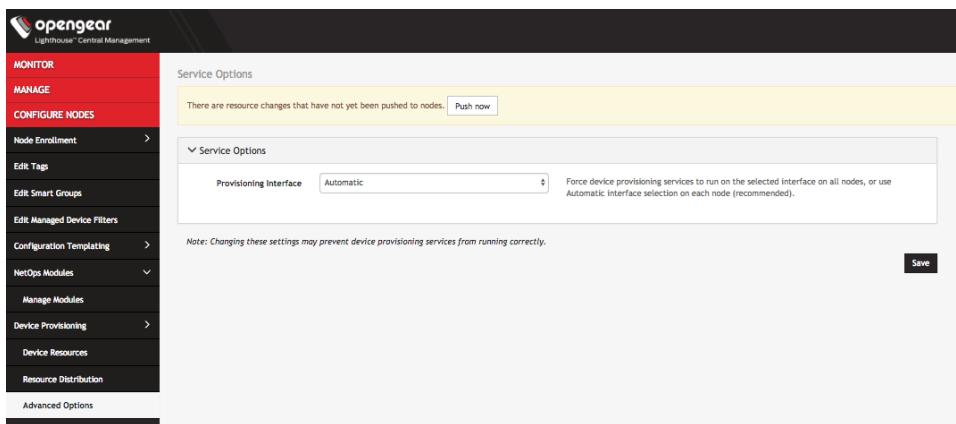
1. Create a Lighthouse Smart Group, as documented in the *Creating Smart Groups* section of this User Guide.
2. Under **Resource Distribution > Dynamic Node Inventory List**, click the **Add** icon



3. Choose a **Name** to identify this bundle, e.g. *LabInventory*
4. Select the **Smart Group** to link to this inventory, this **Smart Group** search is dynamically evaluated to a list of nodes each time resources are pushed
5. Select the bundles to distribute, as listed in the **Resource Distribution** table

10.4.3 Push Resources

Changes to resource bundle or distribution configuration are not applied to nodes immediately, they must be explicitly pushed to nodes.



When changes are detected, a **Push now** button appears on **Device Provisioning** UI pages.

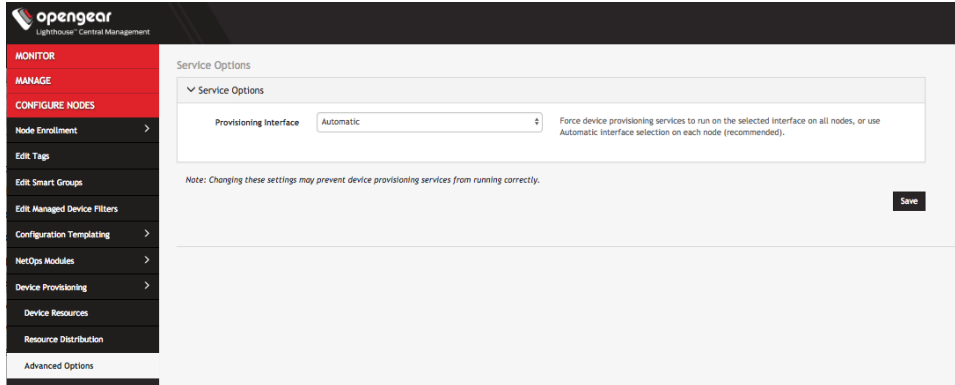
Resources may always be resynchronized by choosing **CONFIGURE NODES > Device Provisioning > Resource Distribution** and clicking **Push Resources**.

The push operation distributes resource bundles as configured, then restarts ZTP services on remote nodes. Push status is displayed above **Device Provisioning** UI pages.

10.4.4 Manage Device Provisioning on Each Interface

To manage how device provisioning behaves on each interface:

1. Select **CONFIGURE NODES > Device Provisioning > Advanced Options**



2. Choose the desired interface or choose **Automatic** for automatic interface selection on each node.
3. Click **Save**.

10.5 CLI-based workflow

Advanced users or users may choose to manage device resources and resource distribution with direct access to the central file repository on Lighthouse.

All necessary resource and configuration files are uploaded to Lighthouse using the **Secure Copy** protocol, ie. scp, WinScp or similar - or advanced users may prefer to use git directly.

If you have adopted DevOps-style configuration management using your own source repository (such as git or Subversion) and/or configuration deployment using continuous integration (such as Jenkins or GitLab) this interface also provides a convenient way to hook the OpenGear system into these tools and workflows. For example, a configuration commit in the upstream system could automatically proliferate to the Lighthouse file repository, and then in turn to the downstream nodes.

NOTE: Changes made via Lighthouse UI or API will overwrite those made by direct repository access and vice versa, therefore UI-based or CLI-based operation should be considered mutually exclusive modes of operation.

10.5.1 Create configuration YAML

The next step is to assign resource files to specific device types (collectively known as *device resources*), and to assign device resources to be deployed to specific OM2200 nodes.

1. Use a YAML file to bundle device resources and control the distribution of device resources from Lighthouse to the OM2200 nodes.
2. In the *ztp-test* directory, create a file with the `.yaml` or `.yml` extension using the following format:

ztp-test.yml

```

device_resources:
  demo-arista:
    device_type: arista
    config_file: 'demo_arista.cfg'
    image_file: 'arista_eos.swi'

deployment:
  all:
    - demo-arista

```

NOTE: Indentation is meaningful in YAML, and you must use space characters to indent instead of tabs.

The **device_resources** list groups and assigns resource files to particular device types (i.e. resource bundles)

Choose an identifier for each resource bundle item, e.g. *demo-arista*

For each item provide the *device_type*, as well as one or more resources, i.e. *config_file*, *image_file* or *script_file*

- **device_type** matches this device resource item to all devices from the specified vendor – it may be one of:
 - *cisco*
 - *cisco_xr*
 - *juniper*
 - *arista*
 - *aruba*
 - *huawei*
 - *cumulus*
 - *pica*
 - *opengear*
- **config_file** is the initial configuration file for the device to load via ZTP, as present in the *downloads* directory
- **image_file** is the initial software image for the device to load via ZTP, as present in the *downloads* directory

NOTE: HPE/Aruba devices do not support the image upgrade via ZTP.

NOTE: The Cisco Autoinstall process does not support image upgrade via ZTP. To automate image upgrade, you must supply a TCL script file rather than a configuration file.

- **script_file** is used by devices required templated resources, i.e. Cumulus and Pica8 – see earlier note about templated resources
- **mac_address** optionally target this bundle at the listed MAC address(es), which may be specified in full, using a wildcard (e.g. 00:10:FA:C2:BF:*), or negated to exclude from the match (e.g. !01:23:45:67:89:AB)
- **serial_number** optionally target this bundle at the listed serial number(s)

Device resource items are then assigned to OM2200 nodes using the **deployment** and optionally the **node_inventory** lists.

See the *Node Inventory* section earlier in this document for an overview of available distribution methods.

The **node_inventory** list defines groups of nodes

Choose an identifier for each inventory, e.g. *BranchInventory* or *LabInventory*

To define a static inventory:

- Create a list named **static**
- List nodes by node ID, e.g. *nodes-1*
- View nodes IDs by running this command on Lighthouse:

```
node-info --all
```

To define a static inventory:

- Create a Lighthouse Smart Group, as documented in the *Creating Smart Groups* section of the Lighthouse User Manual
- Create a key named **smartgroup** with a value of the Smart Group name, this Smart Group search is dynamically evaluated to a list of nodes each time resources are pushed

The **deployment** list assigns device resources to the node inventories defined above, or all nodes

- The built-in **all** identifier selects all nodes
- Otherwise deployment identifiers correspond to **node_inventory** identifiers, e.g. *BranchInventory*
- Assign device resources by listing device resource items, e.g. *demo-arista*
- You may have multiple device resources per deployment

A more comprehensive YAML file may look like:

more-devices.yml

```
device_resources:
  access-switch:
    device_type: juniper
    config_file: 'jn-switch35.config'
    image_file: 'jinstall-ex-4200-13.2R1.1-domestic-signed.tgz'
    mac_address:
      - '00:00:0c:15:c0:*'
      - '!00:00:0c:15:c0:99'
  branch-router:
    serial_number:
      - 'SAD15300D4W'
      - 'FOC1749N1BD'
      - 'AVJ18163A52'
    config_file: 'branch_xr.cfg'
    device_type: cisco_xr
  demo-arista:
```

```

device_type: arista
config_file: 'demo_arista.cfg'
image_file: 'arista_eos.swi'

node_inventory:
  BranchInventory:
    static:
      - nodes-1
      - nodes-2
      - nodes-10
  LabInventory:
    smartgroup: LabNodes

deployment:
  all:
    - demo-arista
  LabInventory:
    - access-switch
  BranchInventory:
    - branch-router
    - access-switch

```

10.5.2 Upload configuration and resources

Assemble device resources on your PC or laptop in preparation for upload:

1. Create a new directory or folder of your choosing, e.g. *ztp-test*
2. Inside *ztp-test*, create a new directory or folder called: *downloads*
3. Copy device resources into *downloads*

If you require templated resources (see earlier note about templated resource):

4. Inside *ztp-test*, create a new directory or folder called *templates*
5. Copy any resources requiring templating into *templates*

Your locally assembled files will now look something like:

```

.
|__ ztp-test
|   |__ ztp-test.yml
|   |__ downloads
|       |__ arista_eos.swi
|       |__ demo_arista.cfg

```

Secure copy the entire *ztp-test* directory to Lighthouse port 2222, to the */srv/central-auto/* directory and authenticating as *root*, e.g. using the `scp` command where 192.168.0.1 to the IP address of Lighthouse:

```

cd ztp-test
scp -P 2222 -rp ./* root@192.168.0.1:/srv/central-auto/

```

Secure Provisioning now automatically propagates the device resources to the OM2200 nodes specified by the YAML, and automatically configures and starts or restarts ZTP services on the OM2200 nodes.

At this point, target device will begin the ZTP process and become provisioned.

10.5.3 Direct git repository access

Advanced users may choose to access the Secure Provisioning git repository on Lighthouse directly, rather than using scp. This has the advantage of supporting commit messages and integrates with upstream git or other continuous integration systems.

Example commands to initialize a local copy of the repository where 192.168.0.1 is the IP address of Lighthouse:

```
ssh-copy-id root@192.168.0.1
cd ztp-test
git init
git remote add origin root@192.168.0.1:2222/srv/central
git add *
git commit -m "Initial commit of ZTP resources"
git push origin master
```

10.5.4 Direct DHCP configuration

As well as (or instead of) the YAML config file, ISC DHCP configuration snippets and ZTP resource files may be manually added by replicating the directory structure that the Secure Provisioning module uses internally.

For example, if a device type is not supported by the YAML config or requires different DHCP options, a DHCP snippet may be manually added to provide this, e.g.:

new-vendor.conf

```
class "new-vendor-class" {
    match if (option vendor-class-identifier = "new-vendor");
    option bootfile-name "new-vendor.cfg";
}
```

Manually added files must be placed in the subdirectory corresponding to the nodes they will be deployed to, i.e. for resources deployed to all nodes, the files must be in the **all** subdirectory. Within this subdirectory, files must be placed in the following:

- DHCP snippets are placed in the **dhcpd** directory
- Resource files such as device configuration or image files are placed in the **downloads** directory
- Templated files such as scripts are placed in the **templates** directory

Directly added files are pushed together with YAML-generated files to the OM2200 nodes. An example local directory structure is shown below with a YAML config file from the earlier example, as well as manual *new-vendor* files added to the **all** deployment directory:

```
.
|__ ztp-test
|__ ztp-test.yml
|__ templates
```

```
| |__ cumulus_setup.sh
|__ downloads
| |__ demo_arista.cfg
| |__ cumulus_interfaces
| |__ arista_eos.swi
|__ all
|__ templates
|__ downloads
| |__ new-vendor.cfg
|__ dhcpd
|__ new-vendor.conf
```

The files are uploaded to the central Secure Provisioning repository, using Secure Copy or git, in the same way as the earlier example:

Secure Copy:

```
cd ztp-test
scp -P 2222 -rp .//* root@192.168.0.1:/srv/central-auto/
```

git:

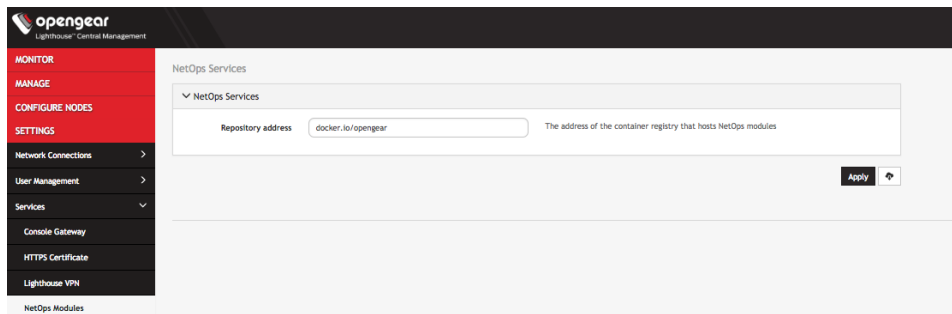
```
ssh-copy-id root@192.168.0.1
cd ztp-test
git init
git remote add origin root@192.168.0.1:2222/srv/central
git add *
git commit -m "Initial commit of ZTP resources"
git push origin master
```

10.6 NetOps Module management

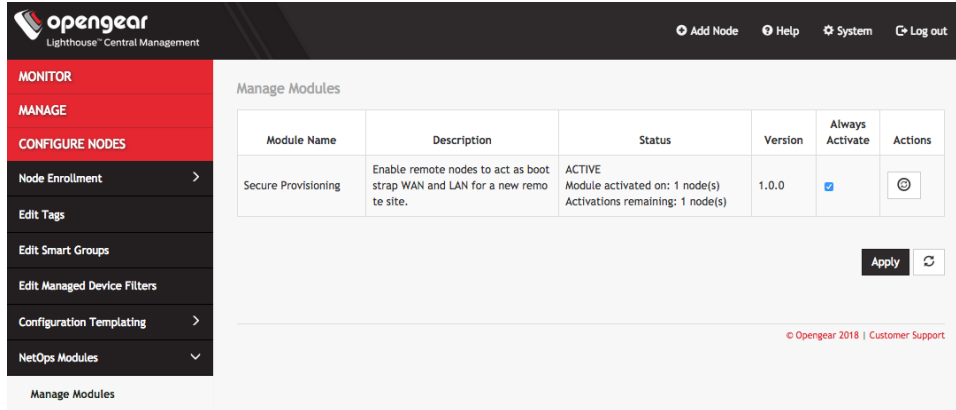
The Provisioning Module is stored in a private repository hosted at Docker Hub. Periodically, module updates and new modules may become available.

You may upgrade the Provisioning Module without independently of upgrading Lighthouse software. To upgrade modules from the UI:

1. Launch an HTTPS browser session to Lighthouse and login as a Lighthouse Administrator user
2. From the menu, select **SETTINGS > Services > NetOps Modules**



3. Click the **Synchronize** icon
4. From the menu, select **CONFIGURE NODES > NetOps Modules > Manage Modules**. This page displays currently installed modules.



5. Click the **Redeploy** icon under **Actions**.

You should see the message *Success: Module update process started, updating may take awhile*. Click the Refresh button on the bottom right to update the status.

To upgrade modules from the CLI:

1. Login to the Lighthouse as root or become root by running `sudo -i`
2. Synchronize Lighthouse with the upstream module repository by running:
`/etc/scripts/netops_sync_handler`
3. Install the updated modules on Lighthouse and the nodes by running:

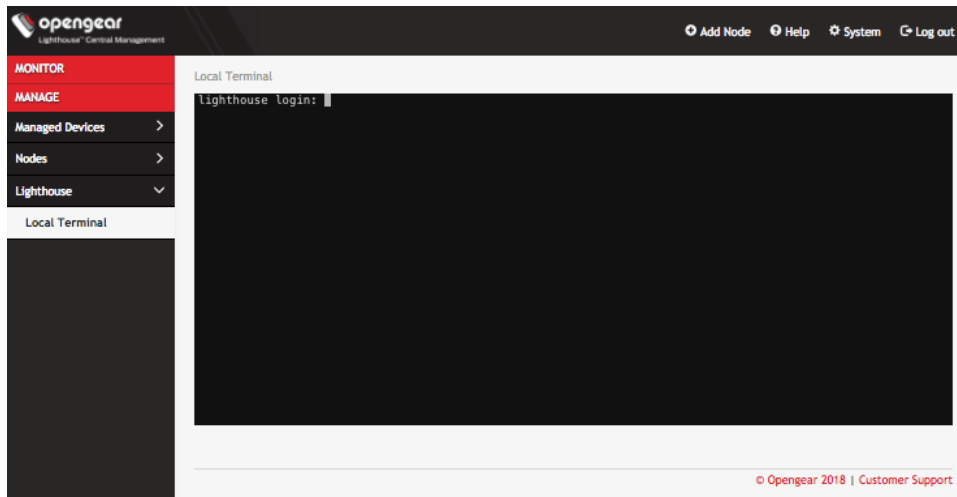
```
token=$(curl -k -L -d '{"username":"root","password":"lighthouse-root-password"}' "https://127.0.0.1/api/v3.0/sessions/" | python -c 'import sys, json; print json.load(sys.stdin)["session"]')
curl -k -H "Authorization: Token $token"
"https://127.0.0.1/api/v3.0/netops/modules/dop/redeploy"
```

NOTE: Replace *lighthouse-root-password* with the password you've set for the root user on Lighthouse.

11. Command line tools

Lighthouse includes a web-based terminal. To access this bash shell instance:

1. Select **MANAGE > Lighthouse > Local Terminal**.



2. At the presented login prompt, enter an administrator's username and press **Return**.
3. A `password:` prompt appears. Enter the administrator's password and press **Return**.
4. A bash shell prompt appears.

This shell supports most standard bash commands and also supports copy-and-paste to and from the terminal.

Lighthouse-specific shell-based tools are listed below.

```
node-command --list-nodes
```

Example `node-command` Output

```
== node-command ID 2017-05-19T14:08:33.360164_29534 ==  
14:08:33 [SUCCESS] BNE-R01-ACM7004-5 192.168.128.2:22  
OpenGear/ACM7004-5 Lighthouse 3b90d826 -- Tue May 9 13:42:16 EST 2017  
  
14:08:33 [SUCCESS] BNE-R02-IM7216 192.168.128.3:22  
OpenGear/IM72xx Lighthouse 3b90d826 -- Tue Jul 5 13:42:16 EST 20167
```

11.1 node-info

`node-info` is a shell-based tool for pulling more detailed information from console servers.

Example `node-info` output

```
$ node-info -A  
BNE-R01-ACM7004-5  
  address: 192.168.128.2  
  id: nodes-1  
  ssh port: 22  
  description: Brisbane Rack 1  
  enrollment status: Enrolled  
  connection status: Connected
```

```
BNE-R02-IM7216
  address: 192.168.128.3
  id: nodes-2
  ssh port: 22
  description: Brisbane Rack 2
  enrollment status: Enrolled
  connection status: Connected
```

11.2 node-upgrade

`node-upgrade` is a tool for running bulk firmware upgrades on managed console servers.

By passing in required information — such as the firmware version to upgrade to, the location of the firmware image to upgrade with, and the nodes to upgrade — via appropriate flags, `node-upgrade` can upgrade the firmware on multiple console servers and report results back to `STD OUT` with a single command.

`node-upgrade` accepts twelve flags as follows:

<code>-h --help</code>	Display this message
<code>-q --quiet</code>	Suppress command output
<code>-b --batch</code>	Suppress node-command output
<code>-l --list-nodes</code>	List all nodes matching query, or all nodes if none selected
<code>-i --node-id=ID</code>	Select node by config ID
<code>-n --node-name=name</code>	Select node by name
<code>-a --node-address=address</code>	Select node by VPN address
<code>-g --smartgroup=name</code>	Select nodes by the smart group they resolve to
<code>-A --all</code>	Select all available nodes
<code>-f --firmware-dir</code>	The directory of the firmware file(s)
<code>-v --version</code>	The firmware version to upgrade to
<code>-z --ignore-version</code>	Ignore firmware version warnings for upgrade

An example node-upgrade run

The following is an example `node-upgrade` command. It sets `/mnt/nvram/` as the directory `node-upgrade` looks to for the firmware image used as the source for all the firmware upgrade attempts. Every console server being managed from the active Lighthouse instance is targeted for an upgrade and the target console servers are set to upgrade to firmware 4.1.0.

```
# node-upgrade -A -f /mnt/nvram -v 4.1.0
```

When run, `node-upgrade` returns information to `STD OUT`, such as the following:

```
Upgrading firmware for device family: ACM550X
Upgrading firmware for device family: CM71XX
Upgrading firmware for device family: CM7196
Upgrading firmware for device family: ACM7004-5
```

```
Upgrading firmware for device family: IM72XX
im7208: flashing firmware file: im72xx-4.1.0.flash
[FAILURE] acm5508: not upgraded to OpenGear/ACM5508-2 version 4.1.0.
Reason for failure: No firmware available for ACM550X device family.
[FAILURE] cm7148: not upgraded to OpenGear/CM7148-2-DAC version 4.1.0.
Reason for failure: netflash failed due to the same firmware currently
on the device.
[FAILURE] cm7196: not upgraded to OpenGear/CM7196A-2-DAC version
4.1.0. Reason for failure: netflash failed due to the same firmware
currently on the device.
[FAILURE] acm7004: not upgraded to OpenGear/ACM7004-5-LMR version
4.1.0. Reason for failure: netflash failed due to the same firmware
currently on the device.
[SUCCESS] im7208: upgraded to OpenGear/IM7208-2-DAC-LR version 4.1.0.
```

`node-upgrade` returns status codes 0 (success) or 1 (failure) when particular conditions are met.

Exit code 0 (success) is returned under the following conditions:

- Success
- Successful upgrade of all nodes.
- No nodes selected for upgrade.
- No firmware found in nominated directory.

Exit code 1 (failure) is returned under the following conditions:

- Missing or invalid command line options.
- The current user is not authorized to execute commands on a node.
- The specified firmware directory was invalid (i.e. because it does not exist or is not readable).
- At least one node upgrade failed.

11.3 ogadduser

`ogadduser` is a shell-based tool for creating users.

Basic `ogadduser` usage syntax is as follows:

```
$ ogadduser -u testuser -p mypassword -g admin
```

NOTE: When a new user is created via `ogadduser`, an entry is added to the `syslog`.

11.4 ogconfig-cli

`ogconfig-cli` allows users to inspect and modify the configuration tree from the command line. It is transactional in nature, allowing users to ensure their configuration is correct before pushing it to the configuration server.

As the root user, start the tool with:

ogconfig-cli

11.4.1 Commands to try from within the ogconfig-cli tool

- help
- get .
- print . 2
- print users[0].username
- find users enabled false

11.4.2 Config searches using ogconfig-cli

Simple config searches can be performed from inside `ogconfig-cli` with the `find` command.

NOTE: The element being searched must be a list, otherwise the command returns an error.

The syntax is:

```
find <path of list to search> <element to search for> <value to search for>
```

For example, to find enabled users use:

```
ogcfg > find users enabled true
```

Or to find the enabled ports on a particular node set:

```
ogcfg> find nodes[0].ports mode 'ConsoleServer'
```

11.4.3 Changing a configuration from within ogconfig-cli

From inside `ogconfig-cli`:

```
ogcfg> set system.hostname "opengear-lighthouse-new"  
ogcfg> push  
ogcfg> quit
```

To see that the change has taken effect:

```
$ cat /etc/hostname
```

A configuration change doesn't take effect until it is pushed to the configuration server. For example, from inside `ogconfig-cli`:

```
ogcfg> set system.hostname "opengear-lighthouse-new-again"  
ogcfg> print system.hostname  
ogcfg> quit
```

To verify that the change did not yet take effect:

```
$ cat /etc/hostname
```

11.4.4 Configuration validation from within `ogconfig-cli`

Configuration is validated before being applied so that an incorrect configuration cannot be accidentally set. For example, from inside `ogconfig-cli`, setting an invalid ethernet link speed is rejected:

```
ogcfg> set system.net.physifs[0].ethernet.link_speed "1GB"
ogcfg> push
Commit failed
  Messages:   String is not in the list of allowed values
              Push command failed

ogcfg> quit
```

11.4.5 Modify LHVPN keepalive timeout for different sized deployments with `ogconfig-cli`

The `lhvpn` timeout (in seconds) should be adjusted depending on the number of nodes to ensure stable connections are maintained. We recommend these settings:

- Fewer than 100 nodes: timeout = 60
- 100 to 599 nodes: timeout = 120
- 600 to 1199 nodes: timeout = 240
- 1200 to 2200 nodes: timeout = 360

The `lhvpn` timeout can be modified by running the following commands, where `<timeout_val>` is the number of seconds:

```
ogcfg> set services.lhvpn.server.keepalive.timeout <timeout_val>
ogcfg> push
```

NOTE: VPN connections will be restarted after pushing a new timeout value.

11.4.6 Support for mounting the hard disks with `ogconfig-cli`

Extra hard disks can be mounted in the Lighthouse VM by adding them to the configuration. Each new disk needs to have a partition created and formatted. Partitions can be created using `fdisk` or `cgdisk`, and should be formatted using the `ext4` filesystem, using the `mkfs.ext4` command:

```
root@lighthouse:~# mkfs.ext4 /dev/sdb1
```

The directory in which to mount the filesystem must be created. In general, new filesystems should be mounted in the provided mountpoint of `/mnt/aux`. Any other filesystems should be mounted within the filesystem mounted here.

Add the information to the configuration system using `ogconfig-cli` as follows, modifying the path for the specific situation.

```
ogcfg> var m !append system.mountpoints map
{8435270-fb39-11e7-8fcf-4fa11570959}: Map <>
ogcfg> set {m}.node "/dev/sdb1"
{b8c37c6-fb39-11e7-971c-23517b19319}: String </dev/sdb1>
ogcfg> set {m}.path "/mnt/aux"
{1fb50d8-fb39-11e7-994c-0f10b09cbd4}: String </mnt/aux>
ogcfg> push
OK
```

11.5 oglicdump

`oglicdump` is a shell-based tool for displaying and saving the current third-party licensing status of a Lighthouse instance.

When used without a switch, `oglicdump` writes the current status to `STD OUT`.

To write this status out to a file, or in machine readable form, or as a raw license container string, or to write out a sub-set of the licensing information (such as licenses for a given SKU), use one of the switches `oglicdump` supports:

<code>-h</code>	Displays this help.
<code>-v</code>	Display version information
<code>-o <file></code>	File to write out to. Default is stdout.
<code>-s <SKU></code>	Specific SKU code to dump out. Default is all SKU codes.
<code>-f <feature></code>	Specific feature value to dump out. This is only valid in conjunction with <code>-s</code> .
<code>-c</code>	Output contacts only. This is only valid in conjunction with <code>-s</code> .
<code>-m</code>	Output machine readable, as in compact formatted.
<code>-r</code>	Output the raw license container strings from config.

11.6 cron

`Cron` service can be used for scheduled `cron` jobs runs. Daemon can be managed via the `/etc/init.d/crond` interface, and `cron` tables managed via `crontab`. `Crontab` supports:

Usage:

```
crontab [options] file
crontab [options]
crontab -n [hostname]
```

Options:

<code>-u <user></code>	define user
<code>-e</code>	edit user's crontab
<code>-l</code>	list user's crontab
<code>-r</code>	delete user's crontab
<code>-i</code>	prompt before deleting
<code>-n <host></code>	set host in cluster to run users' crontabs
<code>-c</code>	get host in cluster to run users' crontabs
<code>-x <mask></code>	enable debugging

To perform start/stop/restart on crond service:

```
/etc/init.d/crond start
```

Cron doesn't need to be restarted when crontab file is modified, it examines the modification time on all crontabs and reload those which have changed.

To verify the current crond status:

```
/etc/init.d/crond status
```

To check current cron jobs running with the following command to list all crontabs:

```
crontab -l
```

To edit or create a custom crontab file:

```
crontab -e
```

This opens a personal cron configuration file. Each line can be defined as one command to run. The following format is used:

```
minute hour day-of-month month day-of-week command
```

For example, append the following entry to run a script every day at 3am:

```
0 3 * * * /etc/config/backup.sh
```

Save and close the file.

11.7 sysflash

sysflash is the shell-based tool for upgrading a Lighthouse instance's system.

Basic syntax is as follows:

```
# sysflash [flags] [path/to/system-image.lg_upg | Percent-encoded URL to firmware-image.lg_upg]
```

NOTE: URLs must be Percent-encoded and image filenames cannot include spaces.

sysflash includes eight flags which modify the standard upgrade behavior as well as the `-h` or `--help` flag, which returns all the available flags and their effects:

<code>-b, --board-name <name></code>	Override board name (currently <code>lighthouse-vm</code>)
<code>-B, --board-revision <version></code>	Override board revision (currently <code>1.0</code>)
<code>-V, --vendor <vendor></code>	Override vendor (currently <code>opengear</code>)
<code>-I, --no-version-check</code>	Do not check software version for upgradability
<code>-m, --no-migration</code>	Do not migrate current config. Start fresh.
<code>-v, --verbose</code>	Increase verbosity (may repeat)

<code>-o, --no-boot-once</code>	Do not modify bootloader (implies <code>--no-reboot</code>)
<code>-r, --no-reboot</code>	Do not reboot after upgrading
<code>-h, --help</code>	Print this help

11.8 Selecting nodes using shell-based tools

There are a number of ways to select nodes, also known as console servers, as targets on which to run a command. These can be used multiple times, or together, to select a range of console servers:

Select individually by name, address, Lighthouse VPN address, config index or smart group (as per `--list-nodes` output):

```
node-command --node-name BNE-R01-IM4248
node-command --node-address 192.168.0.33
node-command --node-index nodes-1
node-command --smartgroup="model-acm"
```

11.8.1 Select all nodes

```
node-command --all
```

11.8.2 Running commands on selected nodes

Once nodes are selected, the commands to be run for each can be given. These are run on each managed node in parallel. Any command which can be run from a node shell can be run on each managed node.

NOTE: All commands are run as root.

For example, to check the version on two specific, configured nodes, selecting one by name and the other by index, run the following command:

```
node-command --node-name BNE-R01-ACM7004-5 --node-index nodes-2 cat
/etc/version
```

NOTE: When using non-trivial selection arguments, check which target nodes have been selected on the initial command pass by using the `--list-nodes` switch rather than the final command.

12. System upgrades

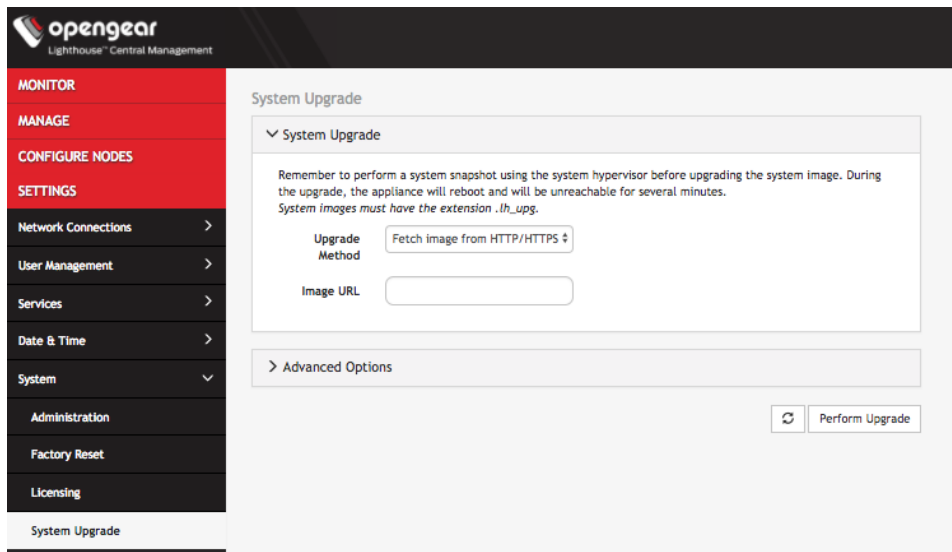
A Lighthouse appliance's system can be upgraded using a `.lh_upg` image file.

Once the upgrade is complete, the Lighthouse instance reboots. It is unavailable during the reboot process.

12.1 Upgrading the system from within Lighthouse

To upgrade a Lighthouse instance's system using the Lighthouse UI:

1. Select **SETTINGS > System > System Upgrade**.
2. Select the **Upgrade Method**, either **Fetch image from HTTP/HTTPS Server** or **Upload Image**.



If upgrading via **Fetch image from HTTP/HTTPS Server**:

1. Enter the URL for the system image in the **Image URL** text-entry field.
2. Click **Perform Upgrade**.

Or if upgrading via **Upload Image**:

1. Click the **Choose file** button.
2. Navigate to the directory containing the `system-upgrade-image.lh_upg` file.
3. Select the `system-upgrade-image.lh_upg` file and press **Return**.
4. Click **Perform Upgrade**.

NOTE: The **Advanced Options** section, which expands to present an **Upgrade Options** text-entry field, should only be used if a system upgrade is being performed as part of an Opengear Support call.

Once the upgrade has started, the **System Upgrade** page displays feedback as to the state of the process.

A system upgrade attempt returns the error **System version was not higher than the current version** if the selected image file is not a more recent version than the installed version.

12.2 Upgrading the Lighthouse system via the Local Terminal

Lighthouse includes a shell-based tool — `sysflash` — that allows a user with administrative privileges to upgrade the instance's system from the **Local Terminal**.

To upgrade Lighthouse instance's system using the Lighthouse **Local Terminal**:

1. Select **MANAGE > Lighthouse > Local Terminal**.
2. At the `[hostname] login:` prompt, enter an administrator username and press **Return**.
3. At the `Password:` prompt, enter the administrator's password and press **Return**.
4. To use `sysflash` in conjunction with a `.lh_upg` file available via an HTTP or HTTPS server:

At the Local Terminal bash shell prompt, enter a URL. **It must be URL-encoded:**

```
sysflash http[s]://%3A%2F%2Fdomain.tld%2Fpath%2Fto%2Ffirmware-upgrade-image.lh_upg
```

5. Press **Return**.

To use `sysflash` in conjunction with a `.lh_upg` file available via the local file system:

1. At the Local Terminal bash shell prompt enter:

```
sysflash /path/to/system-upgrade-image.lh_upg.
```

2. Press **Return**.

NOTE: `sysflash` includes several flags that allow for variations in the standard system upgrade process. These flags should not be used unless directed to do so by Opendgear Support.

Flags are listed by running either of the following at a Local Terminal bash shell prompt:

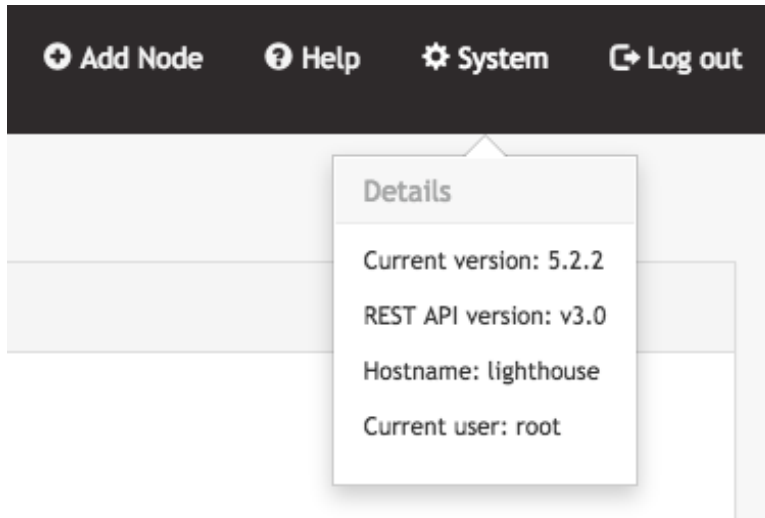
- `sysflash -h` or
- `sysflash --help`
- The same listing is presented in the `sysflash` entry of the Command line tools chapter above.

13. Troubleshooting

13.1 Finding the current Lighthouse instance version

13.1.1 Using the web UI

1. Click **System** on the top right of the Lighthouse instance's web UI.
2. The **Details** menu appears, listing the Lighthouse instance's **Current version**, **REST API version**, **Hostname**, and **Current user**.



13.1.2 Via the local Lighthouse shell

1. Click **MANAGE > Lighthouse > Local Terminal**
2. At the `[hostname] login:` prompt, enter an administrator username and press **Return**.
3. At the `Password:` prompt, enter the administrator's password and press **Return**.
4. At the bash shell prompt, enter `cat /etc/version` and press **Return**.

The current Lighthouse instance's version is returned to `STD OUT`. For example:

```
root@lighthouse:~# cat /etc/version
5.2.2
```

NOTE: The procedure above uses the Web UI to reach the Lighthouse Local Terminal. This is not the only way to reach the Lighthouse shell and `cat /etc/version` works in any circumstance where an administrator has access to the Lighthouse shell. For example, many of the Virtual Machine Manager applications that can run a Lighthouse instance offer virtual console access. If this is available and an administrator logs in to the Lighthouse shell via this console, the command string works as expected.

13.1.3 Other information sources related to a Lighthouse instance's version

Two other command strings can be useful when specifics about a particular Lighthouse instance are needed.

Both these commands can be run by an administrator with access to a running Lighthouse instance's bash shell.

First is `cat /etc/sw*`. This command concatenates the following four files to `STD OUT`:

```
/etc/sw_product  
/etc/sw_variant  
/etc/sw_vendor  
/etc/sw_version
```

For example:

```
# cat /etc/sw*  
lighthouse  
release  
opengear  
5.2.2
```

Second is `cat /etc/issue`. `/etc/issue` is a standard *nix text file which contains system information for presenting before the system's login prompt. On a Lighthouse instance, `etc/issue` contains the vendor, and the Ironman/Lighthouse version

```
# cat /etc/issue  
Opengear Lighthouse 5.2.2 \n \l
```

13.2 Technical support reports

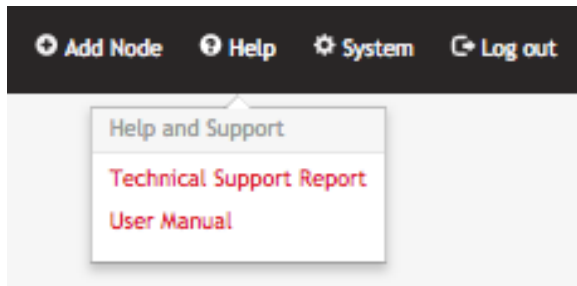
Lighthouse 5 can generate a technical support report that includes Lighthouse configuration information and the current system log for the Lighthouse VM.

In the case of contacting the Opengear Technical Support, the support technician may ask for this report.

13.2.1 Generate a support report via the Lighthouse interface

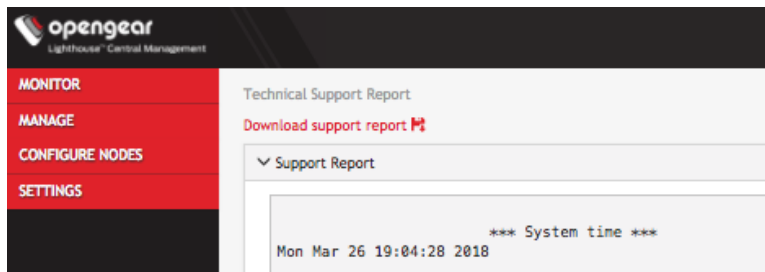
To generate a complete configuration and status report regarding a given Lighthouse VM:

1. Select **Help > Technical Support Report**.



Lighthouse generates this support report on demand and the report includes the current system log. This process can take several minutes.

2. Click **Download support report**.



This downloads a PKZip archive to the local system. The archive's filename is structured as follows:

```
support-[host-name]-[iso-8601-order-date-and-time-stamp].zip
```

It contains two files:

- `system.txt` — the configuration information also presented in the **Technical Support Report** window.
- `messages` — the current Lighthouse VM system log.

The two files are also presented in the **Support Report** text box below the **Download support report** link. Because the report includes the current system log, this is a long but scrollable presentation and is searchable using the web browser's built-in search function.

13.2.2 Generate a support report via the local terminal

To generate a complete configuration and status report regarding a given Lighthouse VM:

1. Select **MANAGE > Lighthouse > Local Terminal**.
2. At the `[hostname] login: prompt`, enter an administrator username and press **Return**.
3. At the `password: prompt`, enter the administrator's password and press **Return**.
4. At the bash shell prompt, enter

```
support-report -z > /tmp/support.zip
```

and press **Return**

The `-z` switch generates the same combined file produced by the **Download support report** link noted in the Lighthouse UI-specific procedure.

NOTE: In the example above, the redirect saves the generated PKZip file to `/tmp/support.zip`. However, be aware that the `/tmp` directory is deleted during a reboot, so the file might be saved to a different location.

Here are two options for copying the file from Lighthouse:

- Use SCP from a Mac or Windows client. As `scp` only requires `ssh` access, no additional configuration is required on Lighthouse for this to work.

```
$ scp root@192.168.0.2:/tmp/support.zip .
root@192.168.0.2's password:
support.zip      100% 321  604.0KB/s   00:00
```

For Windows users, WinSCP on Win10 also works.

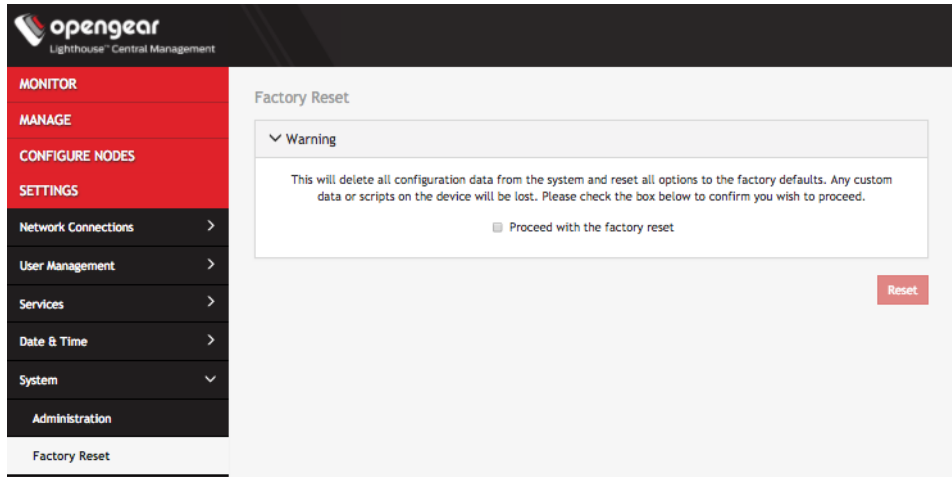
- Use the FTP client on Lighthouse to copy the file to an FTP server. Passive mode must be used for this to work. Example:

```
root@LH5-UK-Lab:/tmp# ftp
ftp> open 192.168.0.216
Connected to 192.168.0.216.
220 im7200-demo-uk FTP server (GNU inetutils 1.4.1) ready.
Name (192.168.0.216:root): fred
331 Password required for fred.
Password:
230- *** Opendgear UK Demo IM7216 ***
230 User fred logged in.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> passive
Passive mode on.
ftp> bin
200 Type set to I.
ftp> put support.zip
227 Entering Passive Mode (192,168,0,216,208,166)
150 Opening BINARY mode data connection for 'support.zip'.
226 Transfer complete.
4132664 bytes sent in 0.128 seconds (32262492 bytes/s)
ftp> quit
221 Goodbye.
```

13.3 Returning a Lighthouse instance to factory settings

To return an enrolled console server to its factory settings using Lighthouse:

1. Login to the Lighthouse web-based interface as **root**. Other users, even those with full administrative privileges, do not have the permissions required to reset the Lighthouse VM to its factory settings.
2. Select **SETTINGS > System > Factory Reset**.



3. Select the **Proceed with the factory reset** checkbox.
4. Click **Reset**.

Running the following shell script as root performs a full factory reset:

```
/usr/bin/factory_reset
```

This script prompts for confirmation before performing the factory reset. The factory reset procedure and the shell script are equivalent to logging in to a console server's web-based management interface (see *Connecting to a console server's web-management interface* above) and doing the following:

1. Select **Administration**
2. Check the **Config Erase** checkbox.
3. Click **Apply**.

NOTE: Returning a console server to its factory settings in this fashion does **not** un-enroll the server from the Lighthouse VM.

NOTE: The latest User Manual can be downloaded from the [Opengear documentation](https://opengear.com/support/documentation) page at opengear.com/support/documentation. It can be accessed by **Help > User Manual** link in the top bar menu.

14. Technical support

Purchaser is entitled to twelve (12) months free telephone support and free e-mail support (worldwide) from date of purchase provided that the Purchaser first register their product(s) with Opengear by filling in the on-line form at opengear.com/product-registration.

Direct telephone, help-desk and e-mail support is available from 09:00 to 20:00, US Eastern Time (UTC - 5 or UTC -4). Other support options are at opengear.com/support.html.

Opengear's standard warranty includes free access to Opengear's [Knowledge Base](#) as well as any application notes, white papers and other on-line resources that may become available from time to time.

Opengear reserves the right to stop support for products no longer covered by warranty.

15. End-user license agreements

15.1 Opendgear end-user license agreement

READ BEFORE USING THE ACCOMPANYING SOFTWARE

YOU SHOULD CAREFULLY READ THE FOLLOWING TERMS AND CONDITIONS BEFORE USING THE ACCOMPANYING SOFTWARE, THE USE OF WHICH IS LICENSED FOR USE ONLY AS SET FORTH BELOW. IF YOU DO NOT AGREE TO THE TERMS AND CONDITIONS OF THIS AGREEMENT, DO NOT USE THE SOFTWARE. IF YOU USE ANY PART OF THE SOFTWARE, SUCH USE WILL INDICATE THAT YOU ACCEPT THESE TERMS.

You have acquired a product that includes Opendgear (“Opendgear”) proprietary software and/or proprietary software licensed to Opendgear. This Opendgear End User License Agreement (“EULA”) is a legal agreement between you (either an individual or a single entity) and Opendgear for the installed software product of Opendgear origin, as well as associated media, printed materials, and “online” or electronic documentation (“Software”). By installing, copying, downloading, accessing, or otherwise using the Software, you agree to be bound by the terms of this EULA. If you do not agree to the terms of this EULA, Opendgear is not willing to license the Software to you. In such event, do not use or install the Software. If you have purchased the Software, promptly return the Software and all accompanying materials with proof of purchase for a refund.

Products with separate end user license agreements that may be provided along with the Software are licensed to you under the terms of those separate end user license agreements.

LICENSE GRANT. Subject to the terms and conditions of this EULA, Opendgear grants you a nonexclusive right and license to install and use the Software on a single CPU, provided that, (1) you may not rent, lease, sell, sublicense or lend the Software; (2) you may not reverse engineer, decompile, disassemble or modify the Software, except and only to the extent that such activity is expressly permitted by applicable law notwithstanding this limitation; and (3) you may not transfer rights under this EULA unless such transfer is part of a permanent sale or transfer of the Product, you transfer at the same time all copies of the Software to the same party or destroy such materials not transferred, and the recipient agrees to this EULA.

No license is granted in any of the Software’s proprietary source code. This license does not grant you any rights to patents, copyright, trade secrets, trademarks or any other rights with respect to the Software.

You may make a reasonable number of copies of the electronic documentation accompanying the Software for each Software license you acquire, provided that, you must reproduce and include all copyright notices and any other proprietary rights notices appearing on the electronic documentation. Opendgear reserves all rights not expressly granted herein.

INTELLECTUAL PROPERTY RIGHTS. The Software is protected by copyright laws, international copyright treaties, and other intellectual property laws and treaties. Opendgear and its suppliers retain all ownership of, and intellectual property rights in (including copyright), the Software components and all copies thereof, provided however, that (1) certain components of the Software, including SDT Connector, are components licensed under the GNU General Public License Version 2, which Opendgear supports, and (2) the SDT Connector includes code from JSch, a pure Java implementation of SSH2 which is licensed under BSD style license. Copies of these licenses are detailed below and Opendgear will provide source code for any of the components of the Software licensed under the GNU General Public License upon request.

EXPORT RESTRICTIONS. You agree that you will not export or re-export the Software, any part thereof, or any process or service that is the direct product of the Software in violation of any applicable laws or regulations of the United States or the country in which you obtained them.

U.S. GOVERNMENT RESTRICTED RIGHTS. The Software and related documentation are provided with Restricted Rights. Use, duplication, or disclosure by the Government is subject to restrictions set forth in

subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 or subparagraphs (c) (1) and (2) of the Commercial Computer Software – Restricted Rights at 48 C.F.R. 52.227-19, as applicable, or any successor regulations.

TERM AND TERMINATION. This EULA is effective until terminated. The EULA terminates immediately if you fail to comply with any term or condition. In such an event, you must destroy all copies of the Software. You may also terminate this EULA at any time by destroying the Software.

GOVERNING LAW AND ATTORNEY'S FEES. This EULA is governed by the laws of the State of Utah, USA, excluding its conflict of law rules. You agree that the United Nations Convention on Contracts for the International Sale of Goods is hereby excluded in its entirety and does not apply to this EULA. If you acquired this Software in a country outside of the United States, that country's laws may apply. In any action or suit to enforce any right or remedy under this EULA or to interpret any provision of this EULA, the prevailing party will be entitled to recover its costs, including reasonable attorneys' fees.

ENTIRE AGREEMENT. This EULA constitutes the entire agreement between you and Opengear with respect to the Software, and supersedes all other agreements or representations, whether written or oral. The terms of this EULA can only be modified by express written consent of both parties. If any part of this EULA is held to be unenforceable as written, it will be enforced to the maximum extent allowed by applicable law, and will not affect the enforceability of any other part.

Should you have any questions concerning this EULA, or if you desire to contact Opengear for any reason, please contact the Opengear representative serving your company.

THE FOLLOWING DISCLAIMER OF WARRANTY AND LIMITATION OF LIABILITY IS INCORPORATED INTO THIS EULA BY REFERENCE. THE SOFTWARE IS NOT FAULT TOLERANT. YOU HAVE INDEPENDENTLY DETERMINED HOW TO USE THE SOFTWARE IN THE DEVICE, AND OPENGEAR HAS RELIED UPON YOU TO CONDUCT SUFFICIENT TESTING TO DETERMINE THAT THE SOFTWARE IS SUITABLE FOR SUCH USE.

LIMITED WARRANTY Opengear warrants the media containing the Software for a period of ninety (90) days from the date of original purchase from Opengear or its authorized retailer. Proof of date of purchase will be required. Any updates to the Software provided by Opengear (which may be provided by Opengear at its sole discretion) shall be governed by the terms of this EULA. In the event the product fails to perform as warranted, Opengear's sole obligation shall be, at Opengear's discretion, to refund the purchase price paid by you for the Software on the defective media, or to replace the Software on new media. Opengear makes no warranty or representation that its Software will meet your requirements, will work in combination with any hardware or application software products provided by third parties, that the operation of the software products will be uninterrupted or error free, or that all defects in the Software will be corrected.

OPENGEAR DISCLAIMS ANY AND ALL OTHER WARRANTIES, WHETHER EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. OTHER THAN AS STATED HEREIN, THE ENTIRE RISK AS TO SATISFACTORY QUALITY, PERFORMANCE, ACCURACY, AND EFFORT IS WITH YOU. ALSO, THERE IS NO WARRANTY AGAINST INTERFERENCE WITH YOUR ENJOYMENT OF THE SOFTWARE OR AGAINST INFRINGEMENT. IF YOU HAVE RECEIVED ANY WARRANTIES REGARDING THE DEVICE OR THE SOFTWARE, THOSE WARRANTIES DO NOT ORIGINATE FROM, AND ARE NOT BINDING ON, OPENGEAR.

NO LIABILITY FOR CERTAIN DAMAGES. EXCEPT AS PROHIBITED BY LAW, OPENGEAR SHALL HAVE NO LIABILITY FOR COSTS, LOSS, DAMAGES OR LOST OPPORTUNITY OF ANY TYPE WHATSOEVER, INCLUDING BUT NOT LIMITED TO, LOST OR ANTICIPATED PROFITS, LOSS OF USE, LOSS OF DATA, OR ANY INCIDENTAL, EXEMPLARY SPECIAL OR CONSEQUENTIAL DAMAGES, WHETHER UNDER CONTRACT, TORT, WARRANTY OR OTHERWISE ARISING FROM OR IN CONNECTION WITH THIS EULA OR THE USE OR PERFORMANCE OF THE SOFTWARE. IN NO EVENT SHALL OPENGEAR BE LIABLE FOR ANY AMOUNT IN EXCESS OF THE LICENSE FEE PAID TO OPENGEAR UNDER THIS EULA. SOME STATES AND COUNTRIES DO NOT ALLOW THE LIMITATION OR EXCLUSION OF

LIABILITY FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS LIMITATION MAY NOT APPLY TO YOU.

15.2 GNU general public license (GPL), version 2

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the

terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms. To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the “copyright” line and a pointer to where the full notice is found.

one line to give the program’s name and a brief idea of what it does.

Copyright © year name of author

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright © *year name of author* Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type ‘show w’. This is free software, and you are welcome to redistribute it under certain conditions; type ‘show c’ for details.

The hypothetical commands ‘show w’ and ‘show c’ should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than ‘show w’ and ‘show c’; they could even be mouse-clicks or menu items—whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a “copyright disclaimer” for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest
in the program ‘Gnomovision’
(which makes passes at compilers) written
by James Hacker.

signature of Ty Coon, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU [Lesser General Public License](#) instead of this License.

16. Standard Warranty

Opengear, Inc., its parent, affiliates and subsidiaries, (collectively, "Opengear") warrant your Opengear product to be in good working order and to be free from defects in workmanship and material (except in those cases where the materials are supplied by the Purchaser) under normal and proper use and service for the period of four (4) years from the date of original purchase from an Authorized Opengear reseller. In the event that this product fails to meet this warranty within the applicable warranty period, and provided that Opengear confirms the specified defects, Purchaser's sole remedy is to have Opengear, in Opengear's sole discretion, repair or replace such product at the place of manufacture, at no additional charge other than the cost of freight of the defective product to and from the Purchaser. Repair parts and replacement products will be provided on an exchange basis and will be either new or reconditioned. Opengear will retain, as its property, all replaced parts and products. Notwithstanding the foregoing, this hardware warranty does not include service to replace or repair damage to the product resulting from accident, disaster, abuse, misuse, electrical stress, negligence, any non-Opengear modification of the product except as provided or explicitly recommended by Opengear, or other cause not arising out of defects in material or workmanship. This hardware warranty also does not include service to replace or repair damage to the product if the serial number or seal or any part thereof has been altered, defaced or removed. If Opengear does not find the product to be defective, the Purchaser will be invoiced for said inspection and testing at Opengear's then current rates, regardless of whether the product is under warranty.