Lighthouse
User Guide

Revision 2019.Q3.0

2019-07-12

TABLE OF CONTENTS

1. About this User Guide

This manual covers Lighthouse and is current as of 2019.Q3.0. When using a minor release (2019.Q3.1), there may or may not be a specific version of the user guide for that release. The current Lighthouse 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. All template types are supported.

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

TERM	DEFINITION
CELLULAR HEALTH	Status of the cellular connection of a node.
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.
INSTANCE	A single running Lighthouse.
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.
PRIMARY INSTANCE	The main instance of Lighthouse used for updating configuration and node enrollment.
REPLICATION	Automatic copying of the primary Lighthouse database to any connected dependent instances. Replication ensures that these instances mirror the same information and maintains connections to the same nodes.
ROLE	A set of access rights for a particular group. Three roles are defined within Lighthouse: Lighthouse Administrator, Node Administrator, and Node User.
SECONDARY/DEPENDENT INSTANCES	Redundant instances of Lighthouse that are used to access Lighthouse information and connected nodes.
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 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.

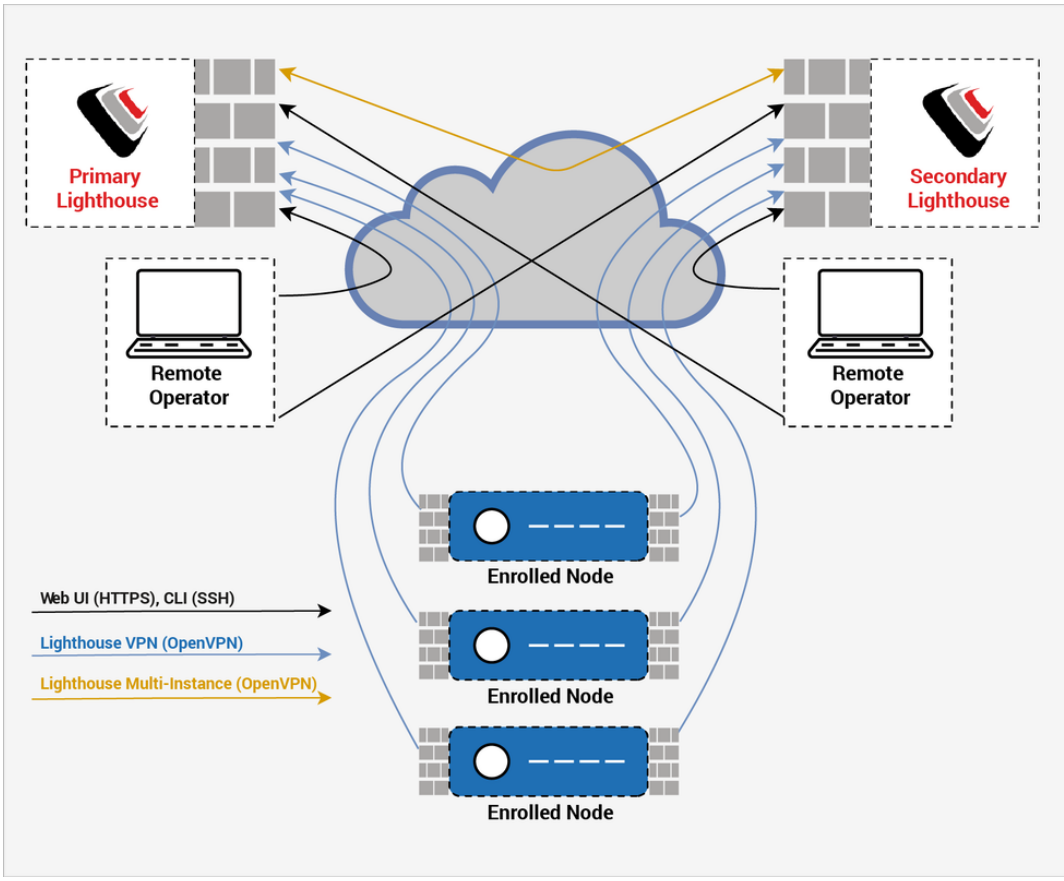
For large and very large deployments, contact us for guidance on the deployment options, including low and zero-touch enrollment. The performance and limitations are dependent on network deployment.

Also, Lighthouse VPN keepalive timeout needs to be modified according to the size of deployment.

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.



NOTE: This diagram depicts High Availability. Without a secondary Lighthouse set up, the diagram remains the same but without the secondary elements.

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 supports REST API versions v1, v1.1, v2, v3, v3.1 and v3.2, 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 v3.2 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.

2.2.4 Multiple Instance Feature

Starting with version 5.3, Lighthouse offers a Multiple Instance feature that allows you to set up a secondary or dependent instance of Lighthouse that automatically receives updates from a primary Lighthouse instance and maintains connections to all of its remote nodes.

Secondary instances are read-only. They may be used to view Lighthouse information specific to that instance, and to connect to its nodes via pmshell. Configuration changes must be performed on the primary instance, which will then update the information displayed on the secondary instance.

The multiple instance feature has the following limitations:

- Only one dependent Lighthouse may be enrolled.
- The primary and secondary Lighthouse must be 5.3 or later.
- Secondary Lighthouse instances are read-only. We recommended that you preconfigure instance specific settings such as hostname, external endpoints, and time zone on a secondary instance before adding it to the primary in a normal way through UI.
- Dependent Lighthouse instances must have zero nodes enrolled before being enrolled to the primary Lighthouse.
- Removing a dependent Lighthouse instance will initiate a factory reset.
- If external endpoints on the primary or secondary Lighthouses are updated after a secondary Lighthouse has been enrolled, it may break replication.
- Only Opengear nodes with a version that supports multiple instance will connect to the secondary instance, which means CS 4.4.1, or later and NGCS 19.Q2.0 or later. Nodes that don't support multiple instance will behave normally on the primary.
- The secondary instance UI offers a limited display.

See Chapter 10 for specific information on using the multiple instance feature.

3. Lighthouse VM installation

To host Lighthouse, the VM needs to be configured to support a 10GB SCSI disk. If you plan on running NetOps Secure Provisioning or SDI/IP Access, you will need a second 120GB disk.

3.1 Lighthouse VM components

Lighthouse VM is available in several formats:

- An Open Volume Format file — **`lighthouse-2019.Q3.0-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-2019.Q3.0-vmx.zip`** —inside a PKZip archive. This is for use with virtual machine managers from VMware.
- A raw (.hdd) file, **`lighthouse-2019.Q3.0-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-2019.Q3.0.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-2019.Q3.0-hyperv.zip`** — inside a PKZip archive. This is for use in Microsoft Hyper-V deployment.
- A Google Compute Engine (GCE) image — **`lighthouse-2019.Q3.0-gce.tar.gz`** — inside a GNU archive. This is for use in Google Compute Engine environment.
- A Microsoft Azure file, **`lighthouse-azure.zip`** for deploying on Azure.
- An Amazon Web Services bootstrap shell script **`aws_bootstrap.sh`** for deploying on AWS.
- An upgrade file for GCE image, **`lighthouse-2019.Q3.0-gce.lh_upg.`**
- An upgrade file, **`lighthouse-2019.Q3.0.lh_upg.`**
- The optional **`ironman-nom.vhd`** file for NetOps.

  NOTE: This NetOps disk is only required to run NetOps modules (including Secure Provisioning, and IP Access) or to run a Docker container on Lighthouse. To install:

  1. Upload the **`ironman-nom.vhd`** file for netops.
  2. Go to images.
  3. Create a new image.
  4. Choose the boot disk by navigating to the storage explorer and selecting the **`ironman.vhd`** disk image. *Optionally* add the netops disk under managed disks.

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.

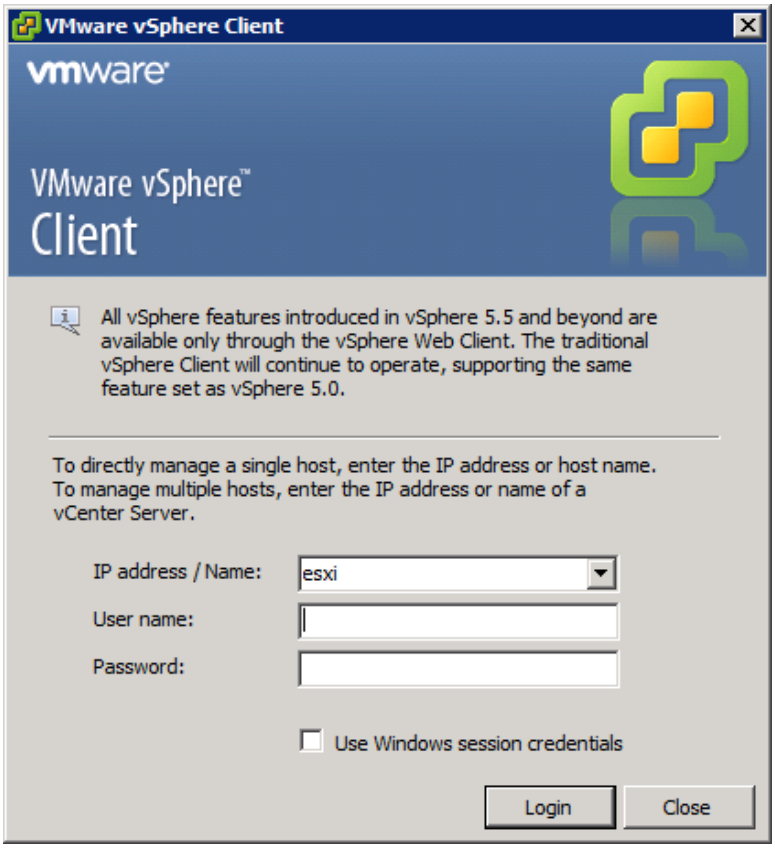
This procedure was tested using the VMware Sphere Client 6.0 running on Windows 7 Enterprise SP 1.

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

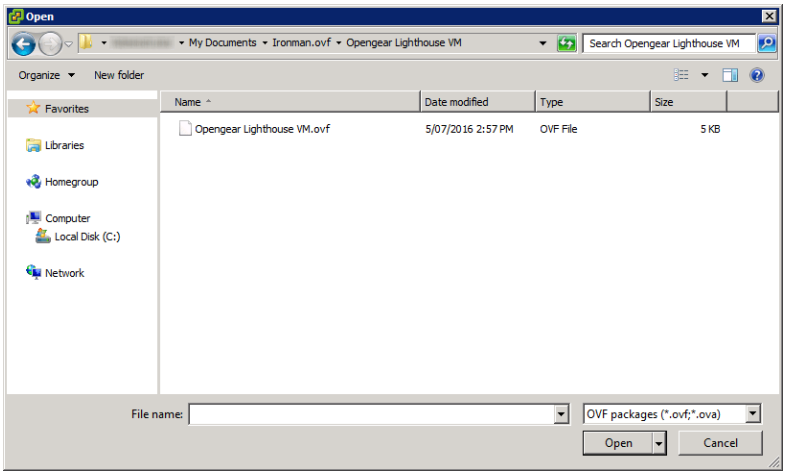
1. From the vSphere Client menu bar, choose **File > Deploy OVF Template**.

   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.

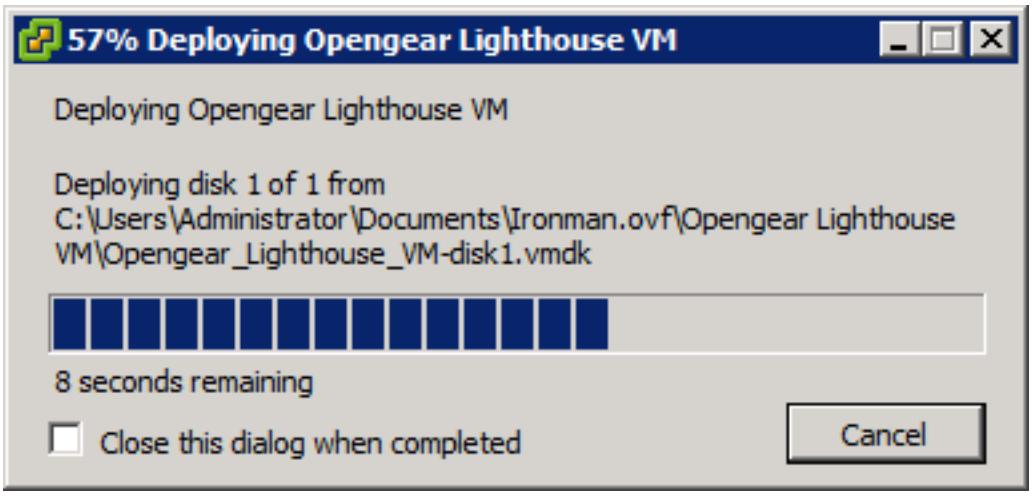
   Navigate to the directory containing the file `Opengear Lighthouse VM.ovf`.

   In the following screenshot, the file is located at `C:\Users\%USERNAME%\My Documents\Ironman.ovf\Opengear Lighthouse VM\`.

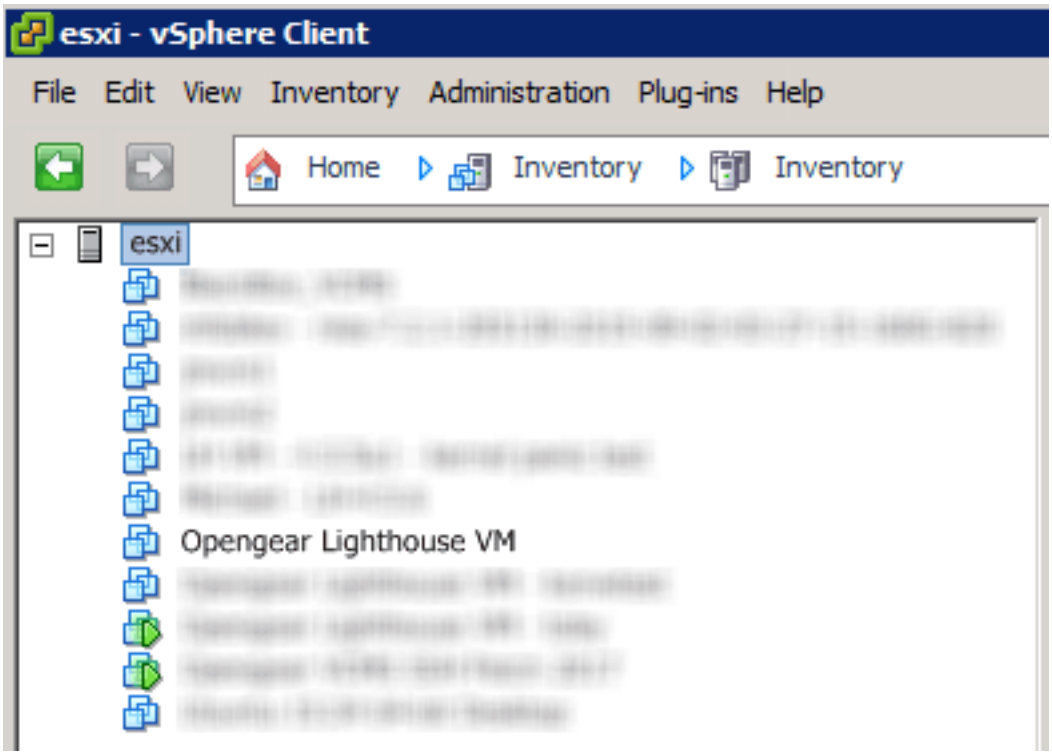


   Select `Opengear Lighthouse VM.ovf` and click **Open**.

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-2019.Q3.0-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 2019.Q3.0 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-2019.Q3.0.zip` archive to `C:\Users\%USERNAME%\Virtual Machines\.`
2. Right-click the `lighthouse-2019.Q3.0-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 2019.Q3.0 virtual machine to the VMware Workstation Pro virtual machines list. If more than one file appears, double-click `Ironman.vmx`.
5. The Lighthouse 2019.Q3.0 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 `ironman-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 Opengear 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 Opengear 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 Opengear 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-2019.Q3.0-ovf.zip` is in `C:\Users\%USERNAME%$\Downloads`.

1. Unzip `ironman-ovf.` It may appear as `lighthouse-2019.Q3.0-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 installed on the host macOS machine and the required PKZip archive, `lighthouse-2019.Q3.0-ovf.zip` is in `~/Downloads`.

1. Unzip `lighthouse-2019.Q3.0-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-2019.Q3.0-ovf.zip` is in `~/Downloads`.

1. Unzip `lighthouse-2019.Q3.0-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-2019.Q3.0-ovf.zip` is in `~/Downloads`.

1. Unzip `lighthouse-2019.Q3.0-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 the `.tar` archive, `lighthouse-2019.Q3.0-raw.hdd.tar` is in `~/Downloads`.

1. Expand `lighthouse-2019.Q3.0-raw.hdd.tar`. This extracts `lighthouse-2019.Q3.0-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-2019.Q3.0-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-2019.Q3.0-ovf.zip` is in `~/Downloads`.

1. Unzip `lighthouse-2019.Q3.0-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-2019.Q3.0-ovf.zip` is in `~/Downloads`.

1. Unzip `lighthouse-2019.Q3.0-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.

3.17 Azure environment

To use the Microsoft Azure environment:

1. Login to the Microsoft Azure portal at https://portal.azure.com
2. Under **Azure services** click the **Storage Accounts** icon.
3. Create a new storage account.
4. Navigate to the newly created storage account, click **storage explorer** and **create a new blob container.**
5. Upload the `ironman.vhd` image provided with the `lighthouse-azure.zip` file.
6. Go to the newly created image and click **Create VM**.
7. Ensure the selected image is correct.
8. Choose the desired virtual machine instance size.
9. Enter the details for the Microsoft Azure admin user with either password OR SSH key authentication.
10. If SSH key authentication is selected, the user will be created without a password and will be unable to access the UI.
11. To login to the Lighthouse UI, the user must then login via SSH with key authentication and configure their password using the ogpasswd utility (eg. sudo ogpasswd -u 'username' -p 'newpassword').
12. Login via SSH with a password will remain disabled for this user.
13. Select the inbound ports enabled for the Lighthouse instance (SSH, HTTPs, and optionally HTTP).
14. Navigate to the next page of configuration (Disks) and select the desired storage option for the boot disk.
15. Go to the **Review** page.
16. After validation passes, click **Create**.
17. Go to the Virtual Machines page, select the virtual machine and open the Serial Console. Lighthouse should now be deploying on Microsoft Azure.
18. To allow nodes to enroll in Lighthouse, you will need to add the following firewall rules in the Microsoft Azure virtual machine control panel:
    a. Go to the virtual machine configuration and select **Networking.**
    b. Add a rule to allow UDP connections from any source to port 1194 on the instance's internal network address (10.0.0.x).
    c. Add a rule to allow UDP connections from any source to port 1195 on the instance's internal network address (10.0.0.x).
    d. HTTPs and SSH should already be allowed from the initial setup If not, add them.
19. Confirm that the Azure instance public IP address has been added to external endpoints in **Settings > Administration.**

3.18 Amazon Web Services (AWS) environment

To use Lighthouse with AWS, you will need to create an account, create an AWS EC2 instance, and create an Amazon Machine Image. You will need to spin up a standard AWS EC2 instance with 30 gigs or more of disk space to ensure there is enough room for the necessary operations.

Amazon offers a walkthrough of the necessary steps here:
https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/get-set-up-for-amazon-ec2.html

To use the AWS environment, you will first need to sign up for AWS.

Sign Up for AWS

When you sign up for Amazon Web Services (AWS), your AWS account is automatically signed up for all services in AWS, including Amazon EC2. You are charged only for the services that you use.

If you have an AWS account already, skip to the next step, **Create an IAM User**.

To create an AWS account:

1. Visit https://portal.aws.amazon.com/billing/signup.
2. Enter your name, address, phone number, and billing information.
3. Enter a phone number to receive a verification code and enter this number on the portal.

Create an IAM User

To provide added security, Amazon recommends you use AWS Identity and Access Management (IAM) user account. Create an IAM user, and then add the user to an IAM group with administrative permissions or grant this user administrative permissions. You can then access AWS using a special URL and the credentials for the IAM user.

If you signed up for AWS but have not created an IAM user for yourself, you can create one using the IAM console. If you aren't familiar with using the console, see Working with the AWS Management Console for an overview.

To create an administrator user for yourself and add the user to an administrators group (console):

1. Use your AWS account email address and password to sign in as the AWS account root user to the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation pane, click **Users** and then click **Add user**.
3. For **User name**, enter **Administrator**.
4. Click the check box next to AWS Management Console access. Select **Custom password**, and then enter a new password.
5. Choose **Next: Permissions**.
6. Under **Set permissions**, click **Add user to group**.
7. Click **Create group**.
8. In the **Create group** dialog box, for **Group name** enter **Administrators**.
9. Click **Filter policies**, and then select **AWS managed -job function** to filter the table contents.
10. In the policy list, select the check box for **Administrator Access**. Then click **Create group**.

11. In the list of groups, select the check box for your new group. If necessary, click **Refresh** to see the group in the list.
12. Click **Next: Tags.** If desired, add metadata to the user by attaching tags as key-value pairs.
13. Click **Next: Review** to see the list of group memberships to be added to the new user. When you are ready to proceed, click **Create user**.
14. To sign in as this new IAM user, sign out of the AWS console, then use the following URL, where *your_aws_account_id* is your AWS account number without the hyphens (for example, if your AWS account number is **1234-5678-9012**, your AWS account ID is **123456789012**):

    https://your_aws_account_id.signin.aws.amazon.com/console/

15. Enter the IAM user name and password that you just created. When you're signed in, the navigation bar displays "*your_user_name* @ *your_aws_account_id*".

Create a Key Pair

A Linux instance has no password. AWS requires you to use a key pair to securely log in to your instance. You specify the name of the key pair when you launch your instance, then provide the private key when you log in using SSH.

If you haven't created a key pair already, you can create one using the Amazon EC2 console.

To create a key pair:

1. Sign in to AWS using the URL that you created in the previous section.
2. From the AWS dashboard, click **EC2** to open the Amazon EC2 console.
3. From the navigation bar, select a region for the key pair.

   NOTE: You can select any region that's available. However, key pairs are specific to a region; for example, if you plan to launch an instance in the US East (Ohio) Region, you must create a key pair for the instance in the US East (Ohio) Region.

4. In the navigation pane, under **NETWORK & SECURIT**Y, click **Key Pairs**.
5. Click **Create Key Pair**.
6. Enter a name for the new key pair in the **Key pair name** field of the **Create Key Pair** dialog box, and then click **Create**.
7. The private key file is automatically downloaded by your browser. The base file name is the name you specified as the name of your key pair, and the file name extension is `.pem`. Save the private key file in a safe place.

NOTE: This is the only chance for you to save the private key file. You'll need to provide the name of your key pair when you launch an instance and the corresponding private key each time you connect to the instance.

To use an SSH client to connect to your Linux instance, use the following command to set the permissions of your private key file so that only you can read it.

```
chmod 400 your_user_name-key-pair-region_name.pem
```

Connect to your instance using your key pair

To connect to your Linux instance with a Mac or Linux computer, specify the `.pem` file to your SSH client with the `-i` option and the path to your private key.

To connect to your Linux instance from a computer running Windows, you can use PuTTY, the Windows Subsystem for Linux, or AWS Systems Manager Session Manager. If you plan to use PuTTY, you'll need to convert the `.pem` file to a `.ppk` file.

Create a Virtual Private Cloud (VPC)

EC2 instance types require that you launch your instances in a VPC. If you have a default VPC, you can skip this section and move to the next task, Create a Security Group. To determine whether you have a default VPC, open the Amazon EC2 console and look for Default VPC under Account Attributes on the dashboard. If you do not have a default VPC listed on the dashboard, you can create a nondefault VPC using the steps below.

To create a nondefault VPC:

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. From the navigation bar, select a region for the VPC. Select the same region in which you created your key pair.
3. On the VPC dashboard, choose **Launch VPC Wizard**.
4. On the **Step 1: Select a VPC Configuration** page, make sure **VPC with a Single Public Subnet** is selected, and click **Select**.
5. On the **Step 2: VPC with a Single Public Subnet** page, enter a name for your VPC in the **VPC name** field. Leave the other default configuration settings and click **Create VPC**. On the confirmation page, click **OK**.

Create a Security Group

You must add rules to a security group that enable you to connect to your instance from your IP address using SSH.

You'll need the public IPv4 address of your local computer. If you are connecting through an Internet service provider (ISP) or from behind a firewall without a static IP address, you need to find out the range of IP addresses used by client computers.

To create a security group:

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. From the navigation bar, select a region for the security group. Select the same region in which you created your key pair.
3. Click **Security Groups** in the navigation pane.
4. Click **Create Security Group**.
5. Enter a name for the new security group and a description.
6. In the **VPC** list, select your VPC. If you have a default VPC it is marked with an asterisk (*).
7. On the **Inbound** tab, create the following rules (choose **Add Rule** for each new rule), and then click **Create**:

a. Choose **HTTP** from the **Type** list, and make sure that **Source** is set to **Anywhere (0.0.0.0/0)**.
b. Choose **HTTPS** from the **Type** list, and make sure that **Source** is set to **Anywhere (0.0.0.0/0)**.
c. Choose **SSH** from the **Type** list. In the **Source** box, choose **My IP** to automatically populate the field with the public IPv4 address of your local computer. Or choose **Custom** and specify the public IPv4 address of your computer or network in CIDR notation.

Launch an Instance

You can launch a Linux instance using the AWS Management Console:

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. From the console dashboard, click **Launch Instance**.
3. The **Choose an Amazon Machine Image (AMI)** page displays a list of basic configurations that serve as templates for your instance.
4. On the **Choose an Instance Type** page, click the hardware configuration of your instance.

   NOTE: You'll need to choose at least a medium-sized image or larger.

5. Click **Review and Launch** to let the wizard complete the other configuration settings for you.
6. On the **Review Instance Launch** page, under **Security Groups**, the wizard created and selected a security group for you. You can use this security group, or you can select the security group that you created when getting set up using the following steps:
   a. Choose **Edit security groups**.
   b. On the **Configure Security Group** page, ensure that **Select an existing security group** is selected.
   c. Select your security group from the list of existing security groups, and then click **Review and Launch.**
7. On the **Review Instance Launch** page, choose **Launch**.
8. When prompted for a key pair, select **Choose an existing key pair**, then select your key pair.

   NOTE: Don't select **Proceed without a key pair**. If you launch your instance without a key pair, then you can't connect to it.

9. Click the acknowledgement check box, and then choose **Launch Instances**.
10. Click **View Instances** to close the confirmation page and return to the console.
11. It can take a few minutes for the instance to be ready so that you can connect to it. Check that your instance has passed its status checks in the **Status Checks** column.

Install Lighthouse on AWS EC2

After creating the AWS EC2 instance:

1. Locate the `aws_bootstrap.sh` script, provided in the current 19.Q3.0 Lighthouse release folder.
2. Connect via SCP and copy `aws_bootstrap.sh` onto the AWS EC2 instance.
3. While SSHed to your instance on AWS, run the built in `aws configure` command.
4. Provide an access key with administrative privileges.

NOTE: At a minimum, the access key sufficient permissions to create, attach, delete, and snapshot EBS volumes as well as create an Amazon Machine Image (AMI).

5. Download the Lighthouse image from the Opengear FTP site.
6. Copy the raw_hdd image to the AWS EC2 instance and untar the file. Optionally, you can untar and then copy the file.
7. From AWS, run the `aws_bootstrap.sh` script with the appropriate parameters to tell it where to find the untarred Lighthouse image on the instance.

   NOTE: *aws-bootstrap.sh* creates an AMI from a Lighthouse image and has the following options:
   -f FILENAME Use the specified local file to create the image
   -r URI Download the image file from the specified URI
   -d DEVICE Attach temporary disk images to the specified device (eg, xvde)
   -n NAME The name to use for generated images (default: Lighthouse)
   -h Display help message

8. When complete, you'll have an AMI called **Lighthouse** you can use to create a Lighthouse instance with any hardware configuration you require.
9. To set a password for the root user on Lighthouse:

   a. Open the **Configure instance details** page of the AMI launch process.
   b. Under the **Advanced Details** section, add a root password using the **userdata** field in the format *password=Whatever123*. If you do not, you will have to log in via SSH to set it.

      NOTE: Optionally, you can specify a custom startup script in the **Advanced Details** section with *script_uri=http://my.domain/my_script.sh*. This script will be run once on first boot. Different user options should be provided on separate lines.

10. When done, the EC2 instance can be shut down and removed. Future instances can be created from the AMI.

NOTE: Currently AWS support is limited to:

- All standard Lighthouse operations
- Running on the AWS platform
- Providing aws-cli tools for interaction with AWS
- Loading the provided SSH key for the root user
- Running custom scripts on startup (see above)
- Providing a root password via userdata (see above)

At this time we do not support:

- Using AWS's database services
- Using AWS's redis services
- Using any of AWS's scalability functionality

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:
   2019.Q3.0
   ```

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:

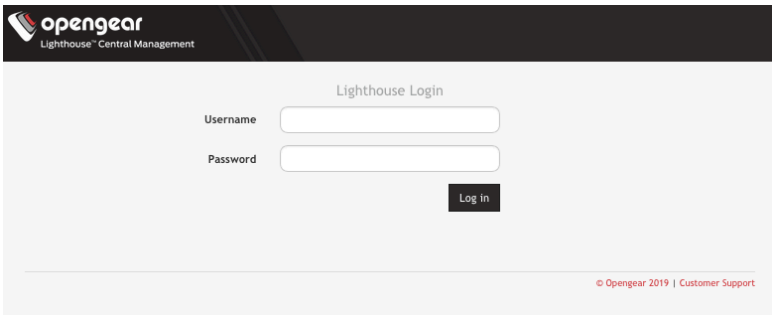
- The static address, `192.168.0.1`, or
- The address it is assigned by any DHCP server it finds. Type **ifconfig** command to see which IP address the VM has been allocated by DHCP.
- Static IP address on another subnet, requiring IP address, mask, gateway set using ogconfig-cli commands.

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.

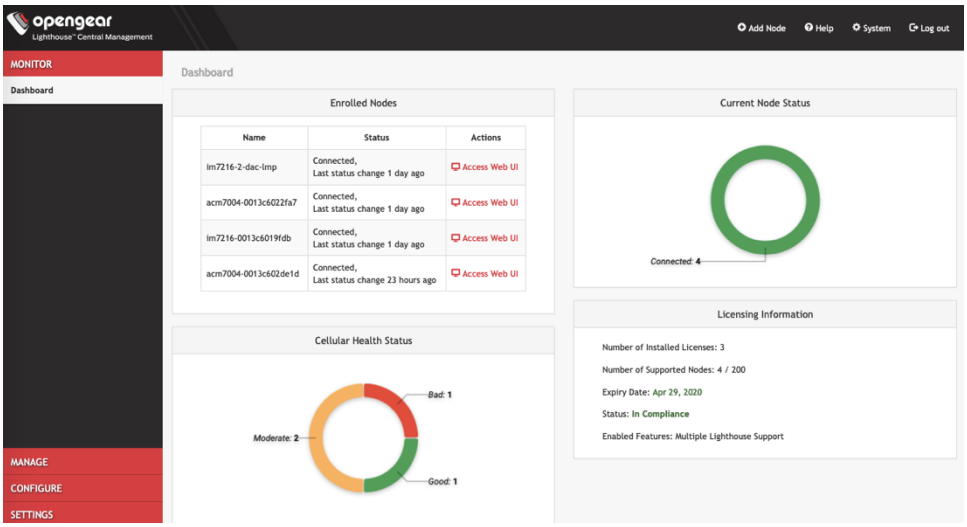


5.3 Login to Lighthouse

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.

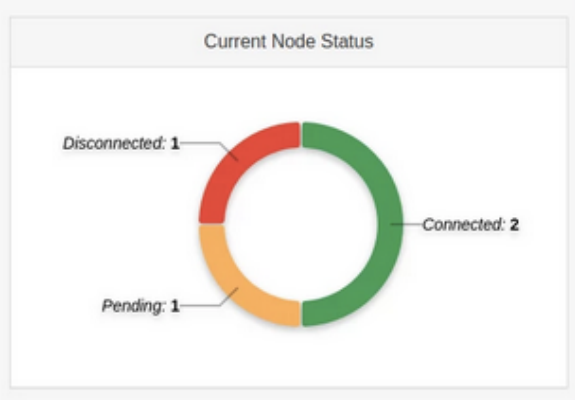
The elements that appear on the **Dashboard** page depend on the privileges granted to the currently logged in user.



For root users, the Dashboard displays **Enrolled Nodes**, **Cellular Health Status**, **Current Node Status**, and **Licensing Information.**

Cellular Health is clickable and will take you to the **MANAGE > Nodes > Node Web UI** page where you can view the **Cellular Health** column with information on each node.

Current Node Status is clickable. Nodes may be **Connected**, **Disconnected,** or **Pending**.



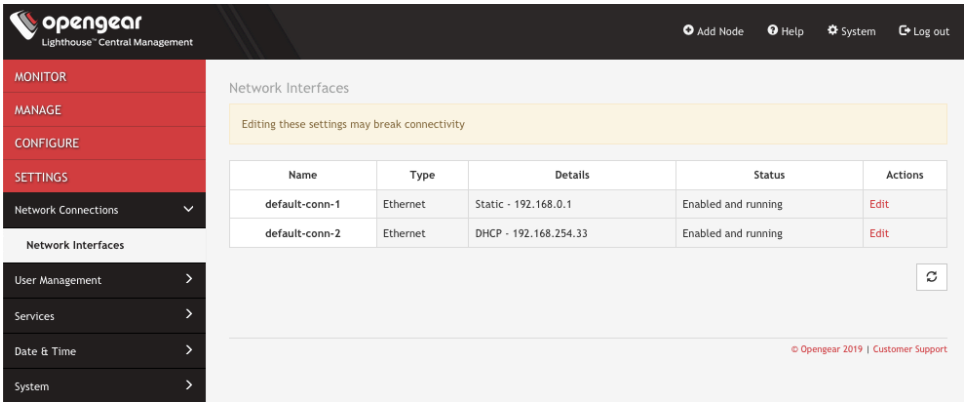
Clicking on:

- **Pending** opens **Configure > Node Enrollment > Pending Nodes**
- **Connected** opens **Manage > Nodes > Node Web UI** filtered to show only connected nodes
- **Disconnected** opens **Manage > Nodes > Node Web UI** filtered to show only disconnected nodes

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**



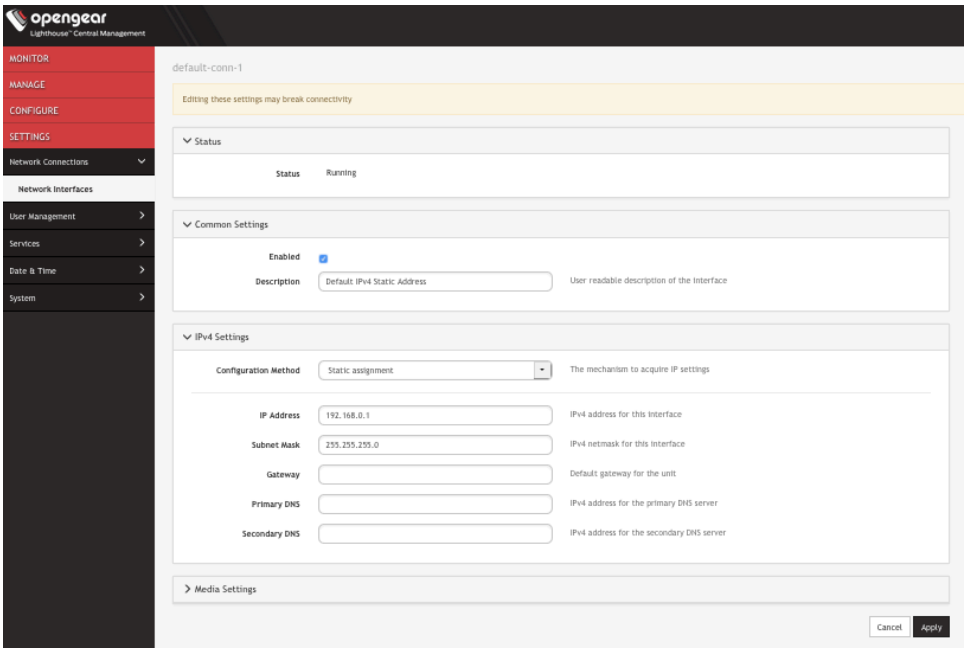
This displays two connections: static and DHCP interfaces.

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

- *default-static* is `net1:static1`
- *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**.

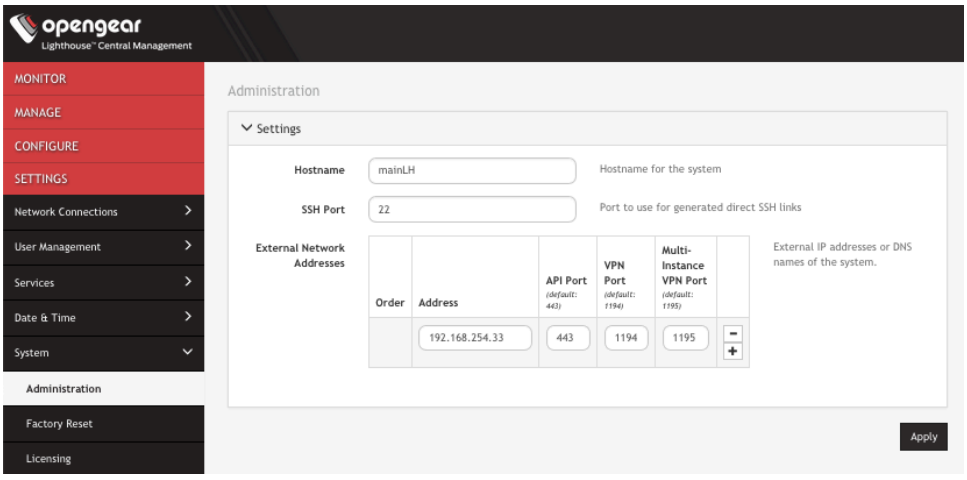


NOTE: Don't change the configuration method. Instead, disable the interface which will not be used by unchecking the **Enabled** checkbox. If **default-static** and **default-DHCP** are changed to the same configuration method (i.e. both are set to **Static assignment** or both are set to **DHCP**) neither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



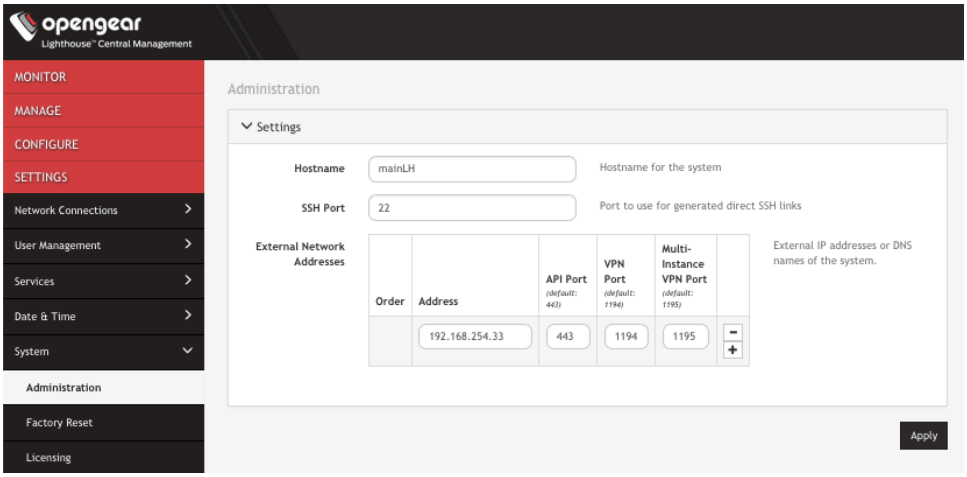
3. Click **Apply**.

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.

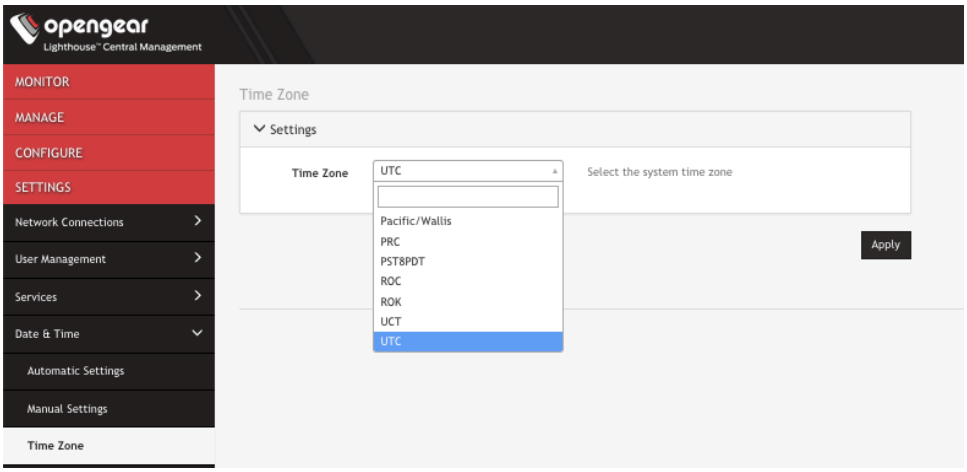
The external IP addresses are sent to a remote node during enrollment in the following order:

1. `net1:dhcp`
2. `net1:static1`
3. The IP address connected to the default gateway.

5.7 Setting the Lighthouse internal clock

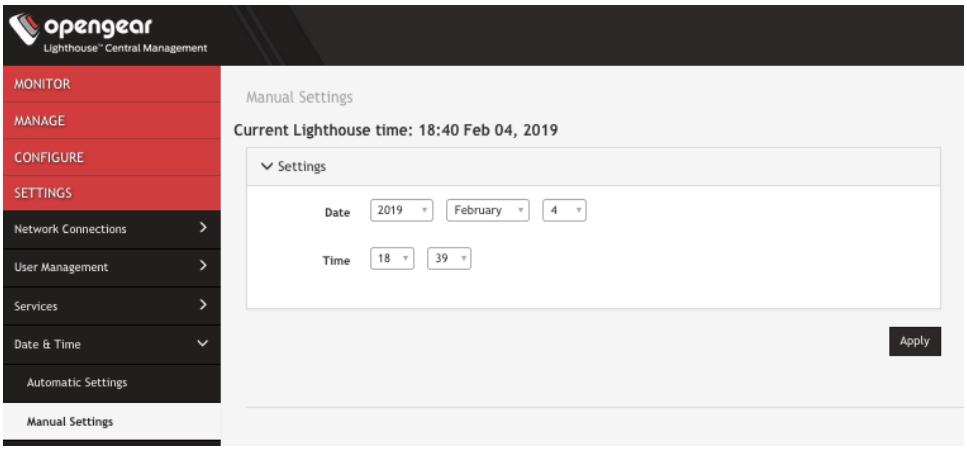
To set the time zone:

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**.



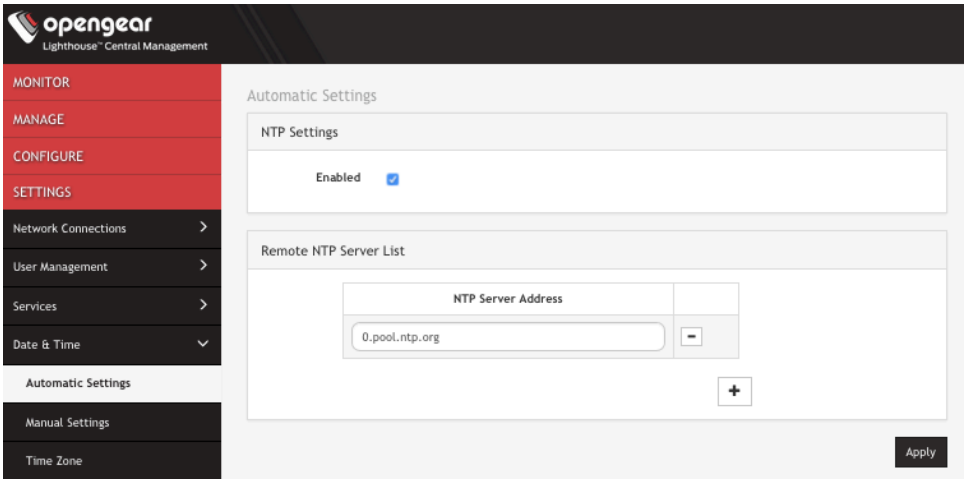
To set the correct time and date, either:

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



or

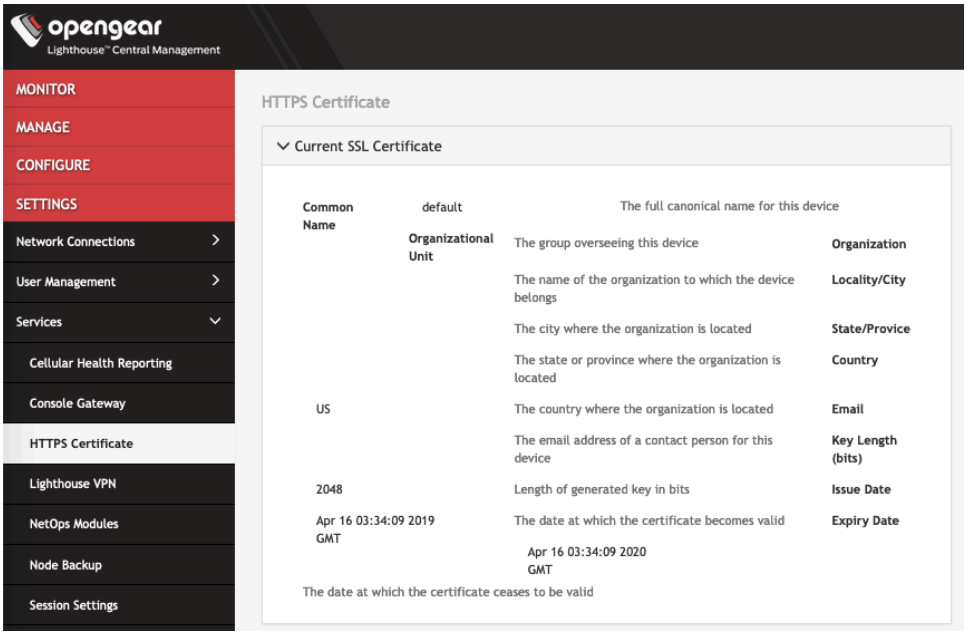
1. Select **SETTINGS > Date & Time > Automatic Settings**.
2. Click the **Enabled** checkbox.
3. Enter a working NTP Server address in the **NTP Server Address** field.
4. Click **Apply**.



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.

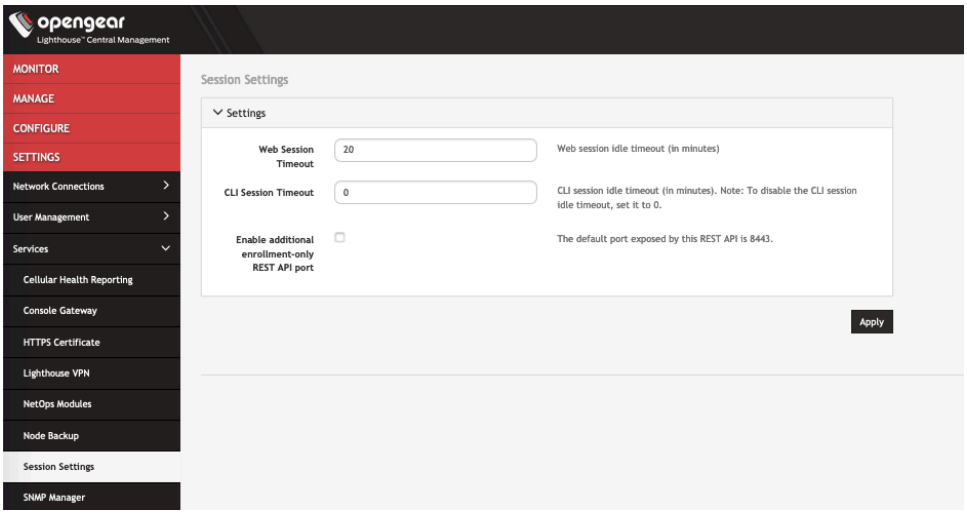


Below this listing is a **Certificate Signing Request** form, which can be used to generate a new SSL certificate.

5.9 Examine or modify Lighthouse Session Settings

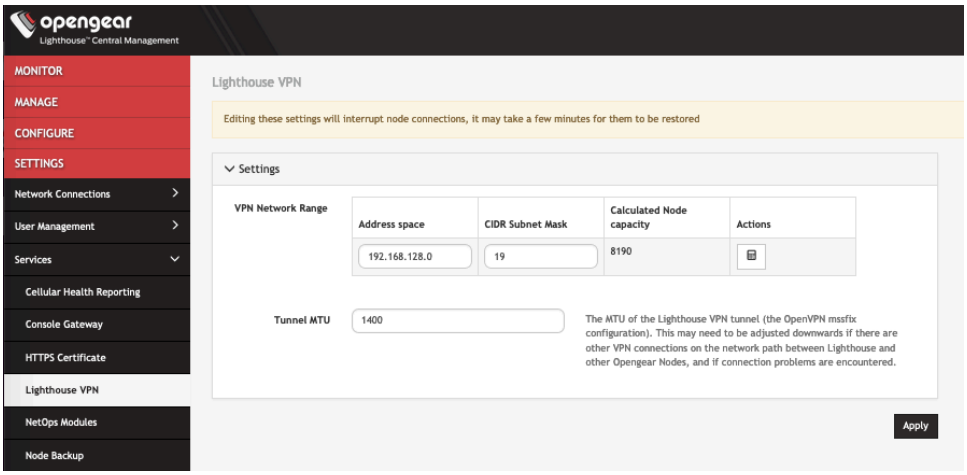
To modify Web and CLI session settings select **SETTINGS > Services > Session Settings**.

- **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

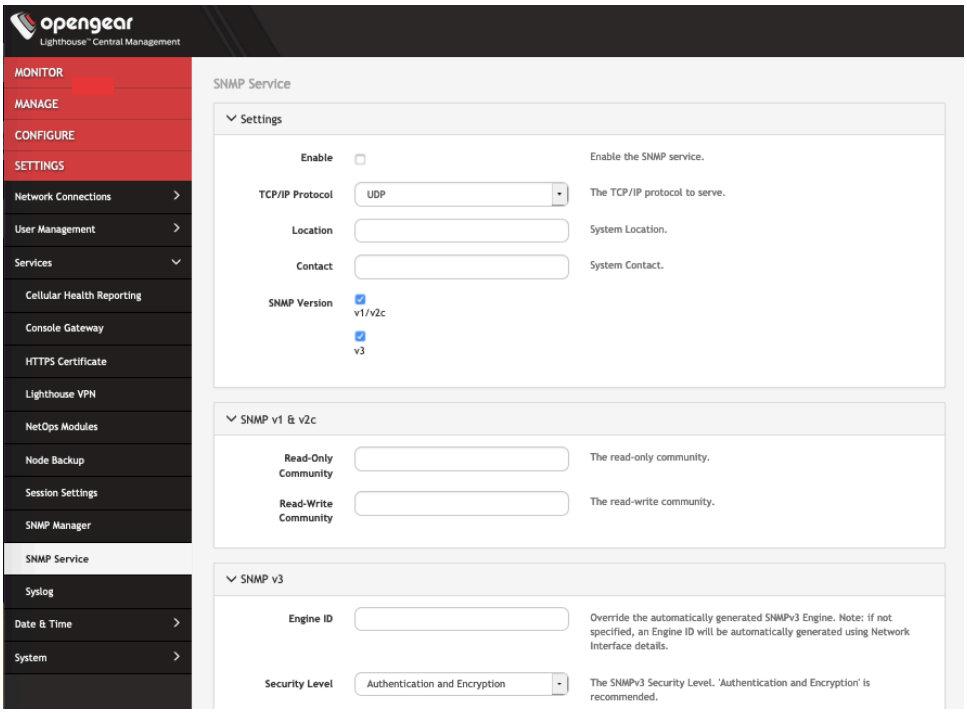
The MTU setting can be configured for traffic that is travelling through the Lighthouse VPN in an attempt to solve MTU path discovery problems. To examine the MTU of the Lighthouse VPN tunnel, or to modify it, select **SETTINGS > Services > Lighthouse VPN**. Allowed values are between 1280 and 1500.



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.

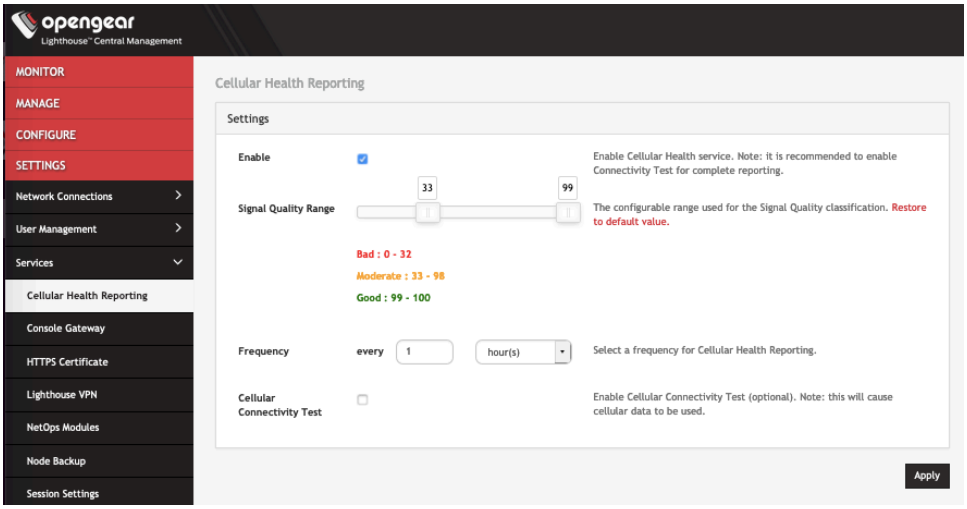


To enable SNMP Service,

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 Cellular Health Settings

Administrative users can control the cellular health reporting settings under **SETTINGS > Services > Cellular Health Reporting**.



- Uncheck the box if you wish to disable the Cellular Health monitoring.
- Adjust the sliders on this screen to define what you consider **Good**, **Bad**, and **Moderate** signal quality to be. This will change the Cellular Health information displayed in various node lists and on the Dashboard.
- Control how frequently Lighthouse will check the signal quality.
- Finally, you can run periodic cellular connectivity tests which make sure the cellular can actually connect.

NOTE: Cellular Health Reporting requires Console Server firmware version 4.5 or greater.

5.13 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 query for licensing information.
ogLhLicenseStatus:
- ogLhLicInstalled
- ogLhLicSupported
- ogLhLicExpiry
- ogLhLicStatus
- ogLhLicFeatureName

You can also query for enrolled node cellular health information.

ogLhNodeCellularHealth

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:

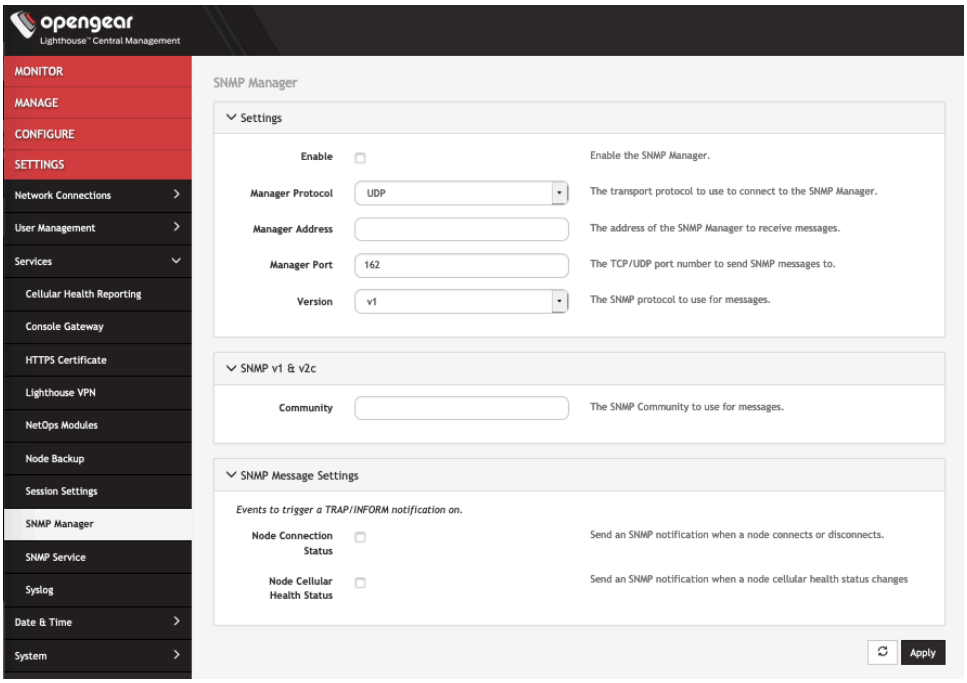
```
snmpwalk -m ALL -v1 -c public 192.168.1.1 ogLhNodeSerialNumber.192.168.128.2
snmpget -m ALL -v1 -c public 192.168.1.1 ogLhNodeSerialNumber.192.168.128.2
```

Get cellular health for all enrolled nodes:

```
snmpwalk -m ALL -c public -v 1 192.168.124.143 ogLhNodeCellularHealth
OG-LIGHTHOUSE-MIB::ogLhNodeCellularHealth.192.168.128.2 = INTEGER: good(4)
OG-LIGHTHOUSE-MIB::ogLhNodeCellularHealth.192.168.128.3 = INTEGER: good(4)
OG-LIGHTHOUSE-MIB::ogLhNodeCellularHealth.192.168.128.4 = INTEGER: bad(2)
OG-LIGHTHOUSE-MIB::ogLhNodeCellularHealth.192.168.128.5 = INTEGER: unknown(0)
OG-LIGHTHOUSE-MIB::ogLhNodeCellularHealth.192.168.128.6 = INTEGER: bad(2)
```

5.14 SNMP Manager Settings

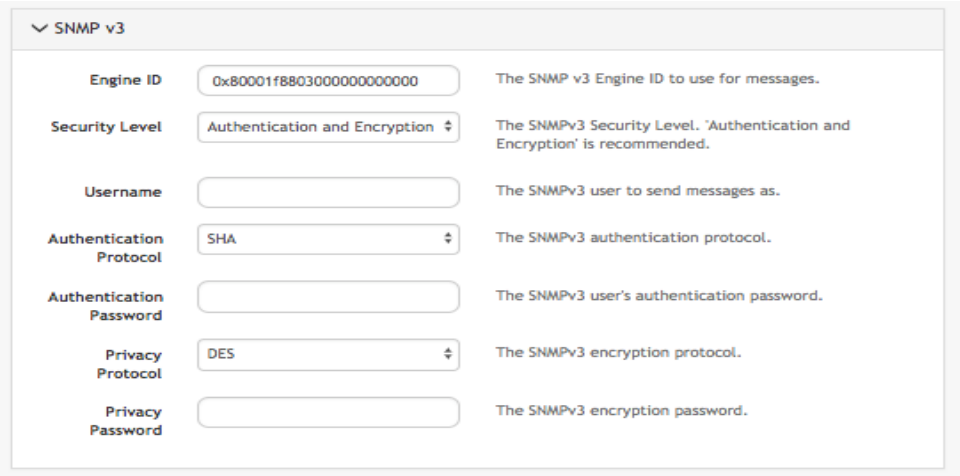
Administrative users can configure the SNMP Manager settings. Select **SETTINGS > Services > SNMP Manager**. The SNMP Manager allows SNMP TRAP/INFORM messages to be sent from Lighthouse to a configured server any time a node connection status is changed.



To enable the SNMP Manager,

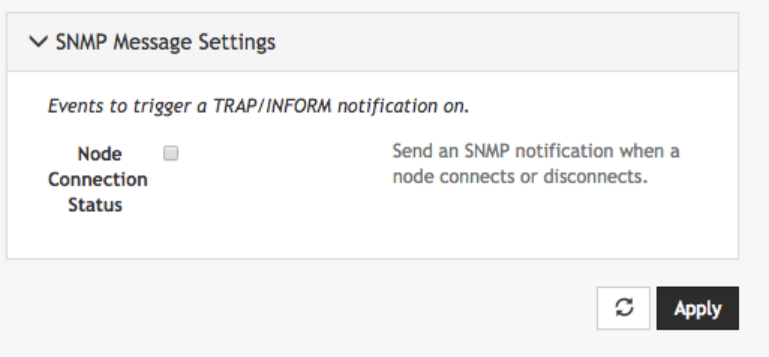
1. Under the **Settings** section, select the **Enable** checkbox.
2. Choose **UDP** or **TCP** as the **Manager Protocol** drop-down.
3. Enter the **Manager Port** to receive SNMP messages.
4. Check the SNMP protocol **Version** from the **v1, v2c**, **v3** drop-down.
5. Choose the **SNMP Message Type**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

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



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:

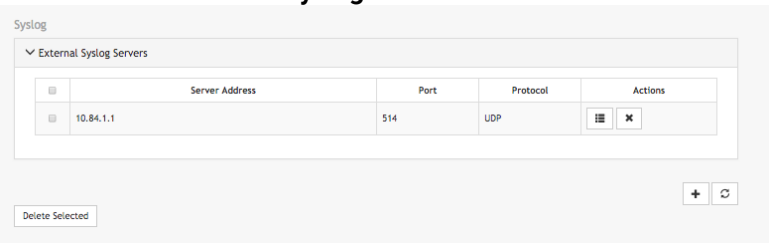
nodeStatusNotif
- ogLhNodeName
- ogLhNodeIndex
- ogLhNodeConnStatus

Structure of notifications for third-party nodes:

thirdPartyNodeStatusNotif
- ogLhThirdPartyNodeIndex
- ogLhThirdPartyNodeName
- ogLhThirdPartyNodeAddress
- ogLhThirdPartyNodeConnStatus

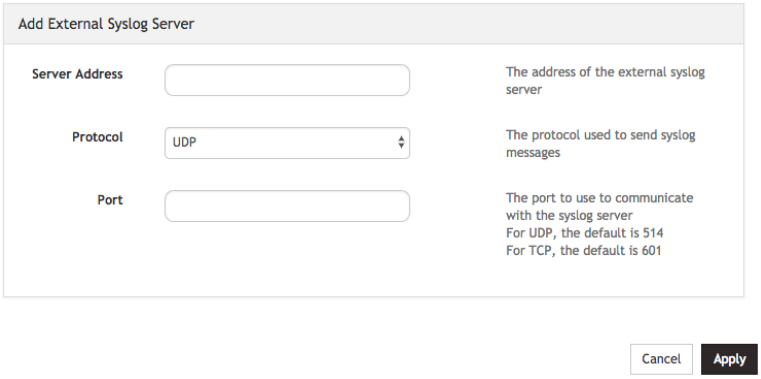
5.15 Syslog export

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.



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**.

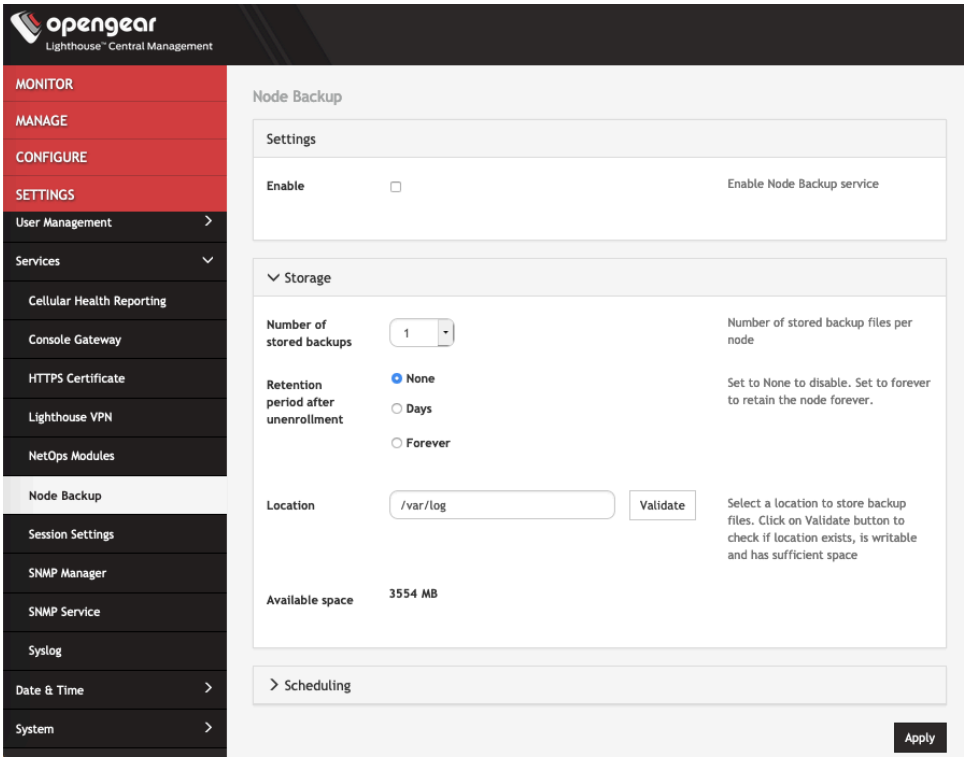
To edit an existing syslog server, click the **Edit** button  under **Actions**. Delete a server by clicking the **x** button  .

5.16 Node Backup

NOTE: Node backup requires firmware 4.6 or later.

Administrative users can enable automatic node backup. Up to 10 backups can be stored on a rolling basis.

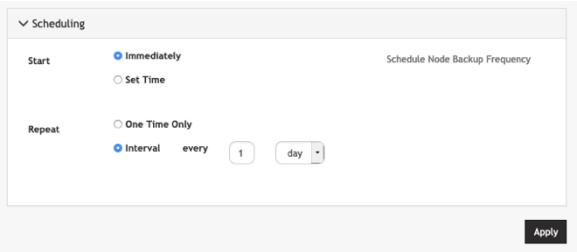
1. Select **SETTINGS > Services > Node Backup**.



2. Click the **Enable** checkbox to turn on this service.
3. Under the Storage section, choose the **Number of stored** backups you wish to keep.
4. Choose how long you wish these backups to be stored after unenrollment. Selecting **Days** opens a field that allows you to enter a number.
5. Enter the **Location** you wish to store the backup files, then click **Validate** to make sure the location exists and has enough space to store them.
6. Click **Apply** or set a schedule.

To set an automated schedule for performing node backups:

1. Click on Scheduling to open the scheduling section.



2. For the **Start** time, choose either **Immediately** or choose **Set Time** to open editable **Date** and **Time** fields.
3. Choose how often you wish to **Repeat** the backup by adjusting the values for **Interval**.

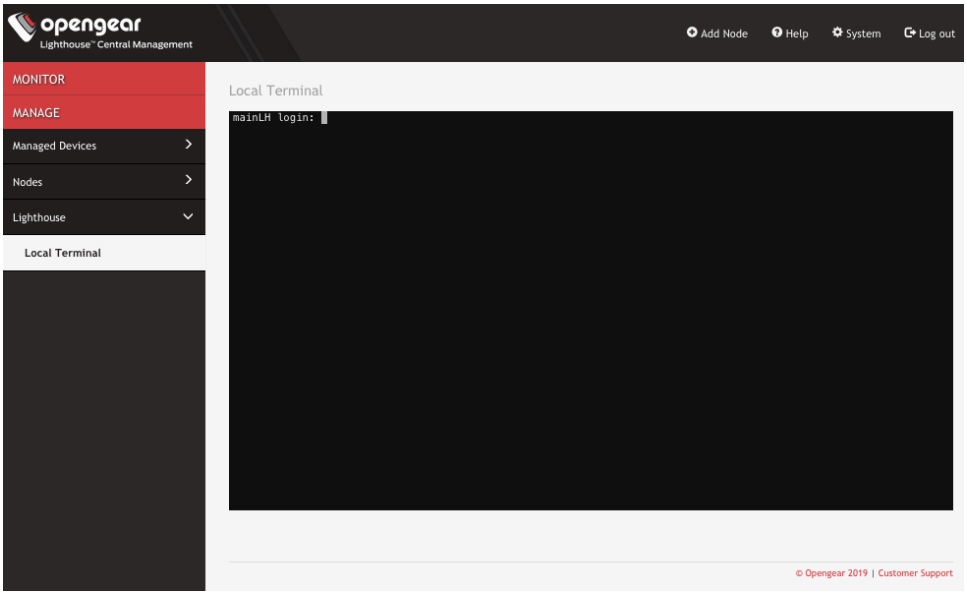
NOTE: You can modify these options by returning to **SETTINGS > Services > Node Backup** at any time.

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

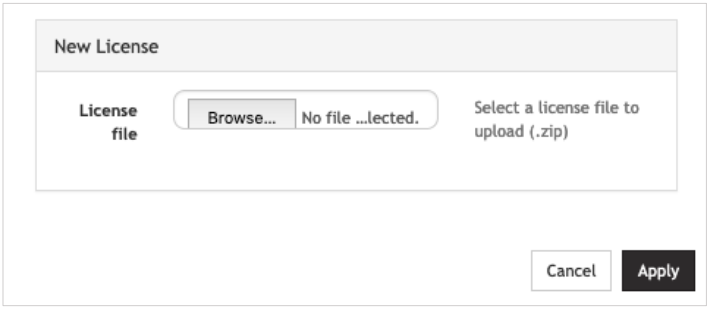
Lighthouse includes support for managing third-party remote nodes at no cost. Support for third-party remote nodes is not built-in to a new Lighthouse instance, however: it is added via a license.

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

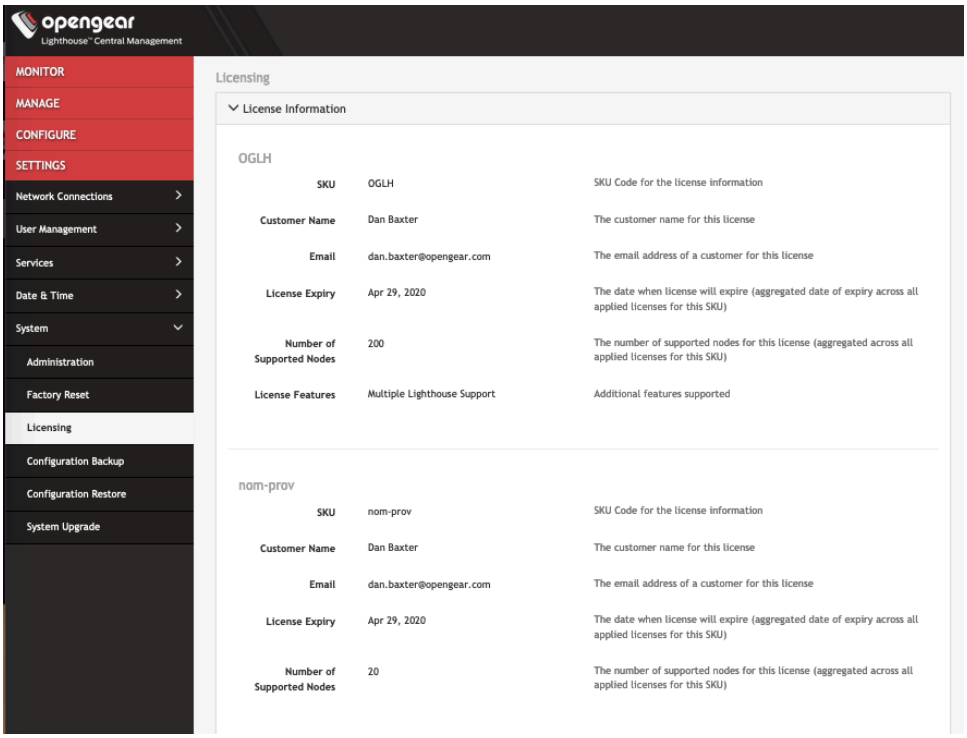
1. Select **SETTINGS > System > Licensing**
2. Click the **+** button on the bottom right of the page. A **New License** dialog opens.



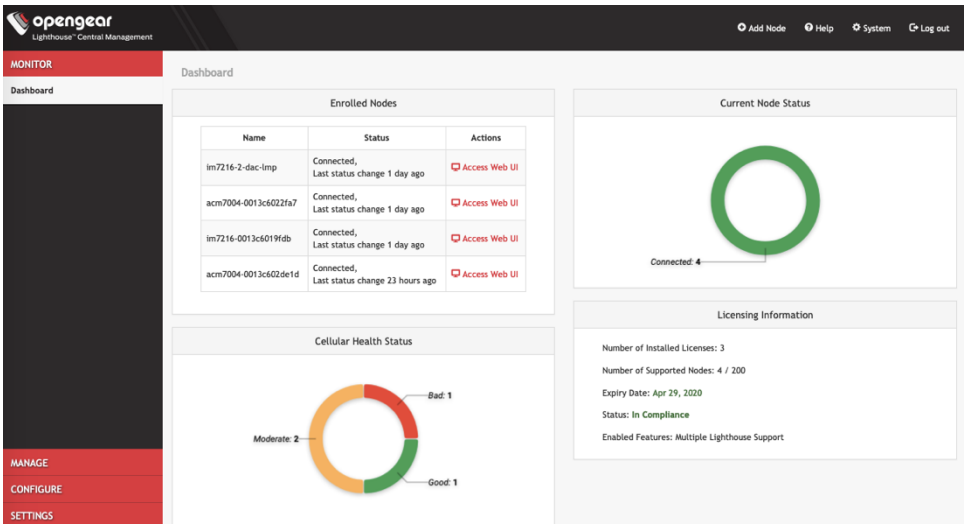
3. Select the license file to upload.
4. Click **Apply**.

7.1.2 Showing installed licenses in the Lighthouse UI

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



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 `STD OUT` (or, using the `-o` switch, a file).

For example:

```
# oglicdump
{
  "OGLH": {
    "contact": {
      "email": "test@test.com",
      "name": "test",
      "phone": "test"
    },
    "features": {
      "additional": {
        "multipleinstance": "1",
        "thirdpartynodes": "1"
      },
      "maintenance": 1548806400,
      "nodes": 20
    }
  }
}
```

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 > 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.

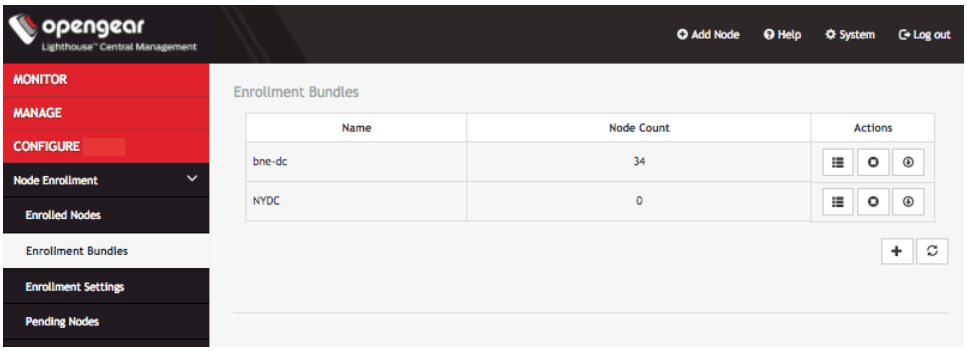
Associating templates with an enrollment bundle allows to run a set of templates on a node, after it has been enrolled. Any template defined on the Lighthouse can be added to an enrollment bundle, and each bundle supports any number of templates.

NOTE: NetOps modules (see NetOps User Guide) can also be associated with enrollment bundles.

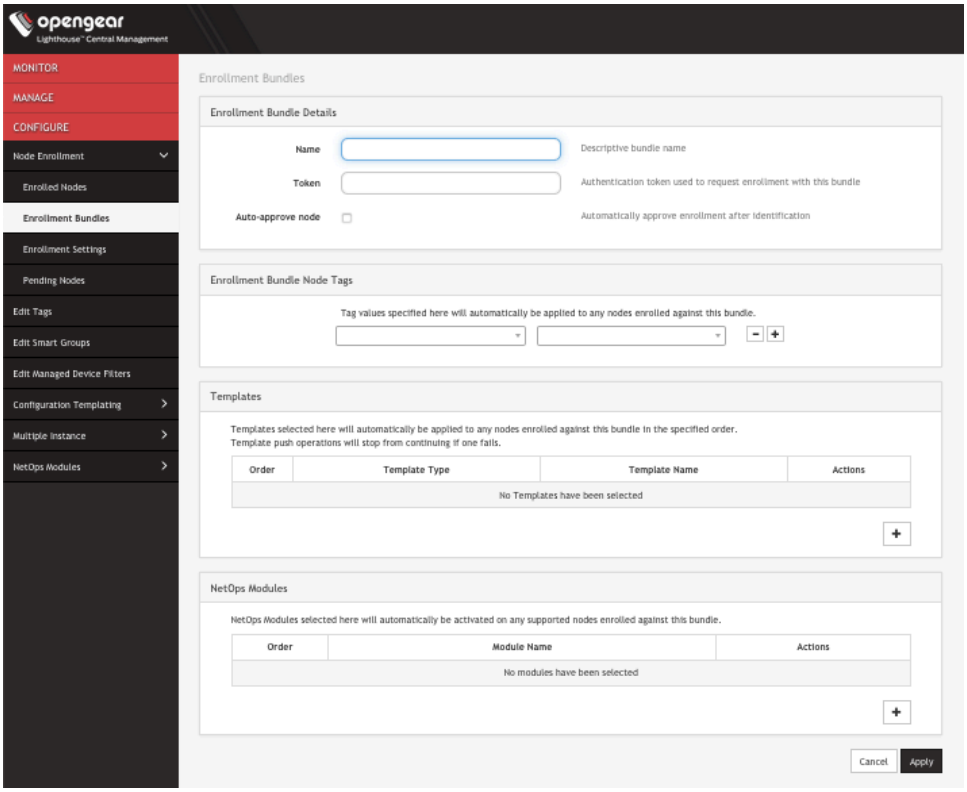
7.2.3 Creating an enrollment bundle

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

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



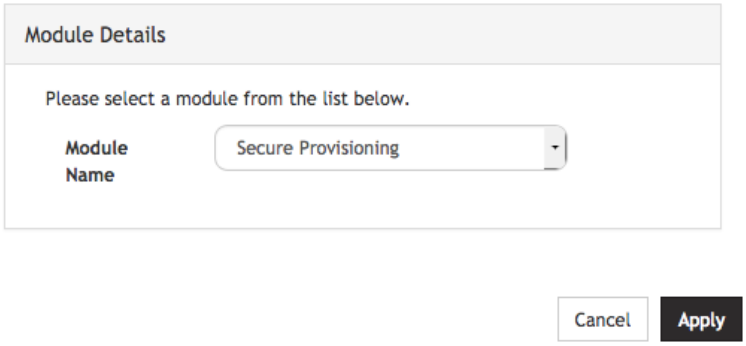
2. Click the **+** button. The **Enrollment Bundle Details** page appears.



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.

6. 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.



7. 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:

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

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

1. Click the **+** button to add a new pair of drop-down menus.
2. Select a value from the **Template Type** menu. The selected template type filters the available names to those templates of that type.
3. Select a value from the **Template Name** menu.
4. Repeat until all desired type-name pairs are displayed.
5. Click **Apply**.
6. 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:

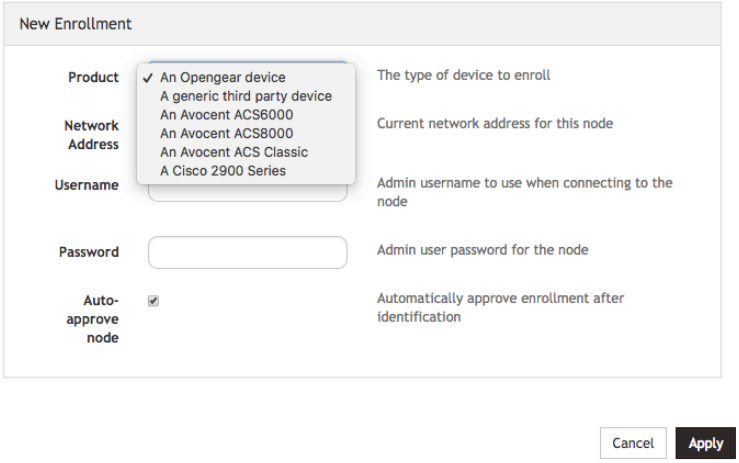
```
script=configure_ports.sh
image=acm7000-3.16.6.image
external_endpoints=192.168.1.2:4444,192.168.1.3:4445
```

7.2.5 Enrollment via Lighthouse Web UI

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

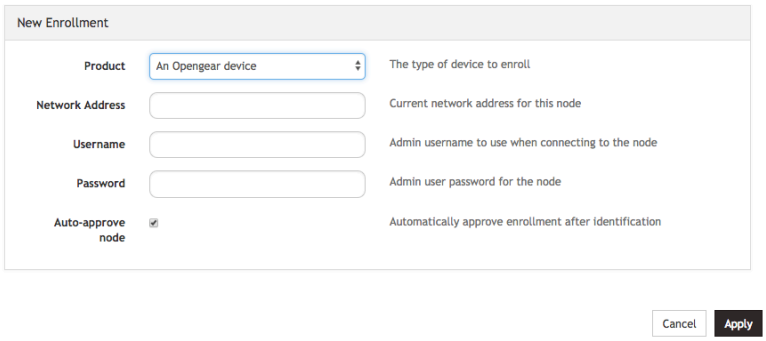
1. Select the **Add Node** shortcut in the top menu bar to bring up the new enrollment dialog.
2. Select the **Product** type from the **Product** drop-down menu.
3. Available options in the **Product** drop-down menu are:

    - An Opengear device
    - A generic third-party device
    - An Avocent ACS6000
    - An Avocent ACS8000
    - An Avocent ACS Classic
    - A Cisco 2900 Series



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.

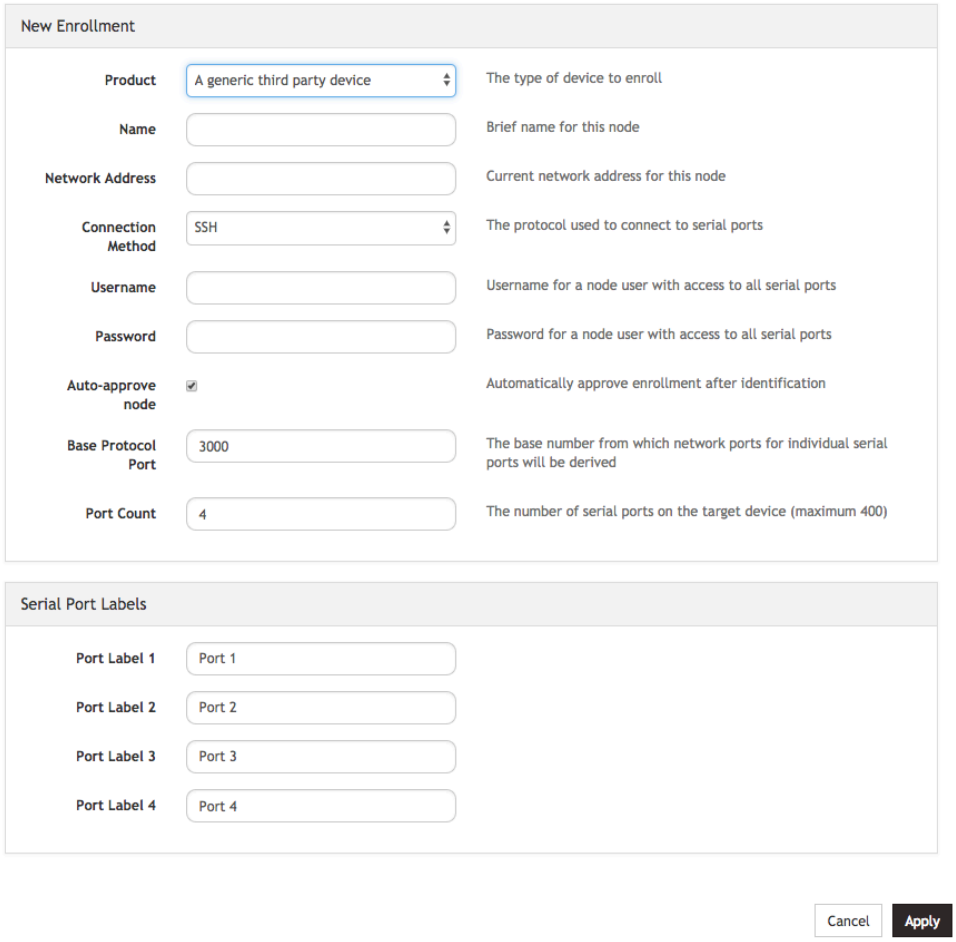
4. Enter the **Name**, **Network Address**, **Username**, and **Password** of the node being enrolled. The **Username** and **Password** fields are for the login credentials required by the remote node being enrolled, *not* the login credentials used to login to the Lighthouse instance.



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.

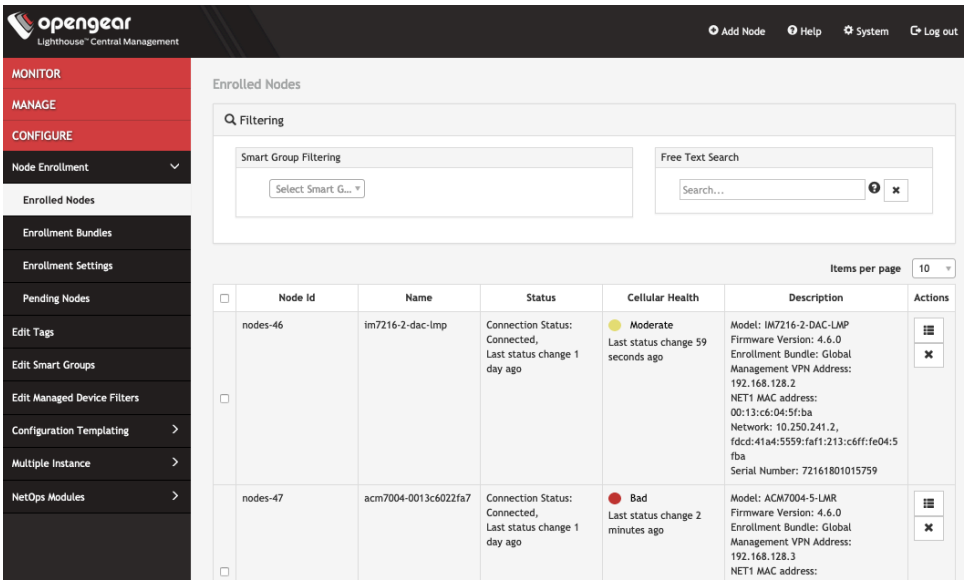
5. To enroll a generic third-party device, there are three more required fields: **Connection Method**; **Base Protocol Port**; and **Port Count**.

   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.



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 > 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:

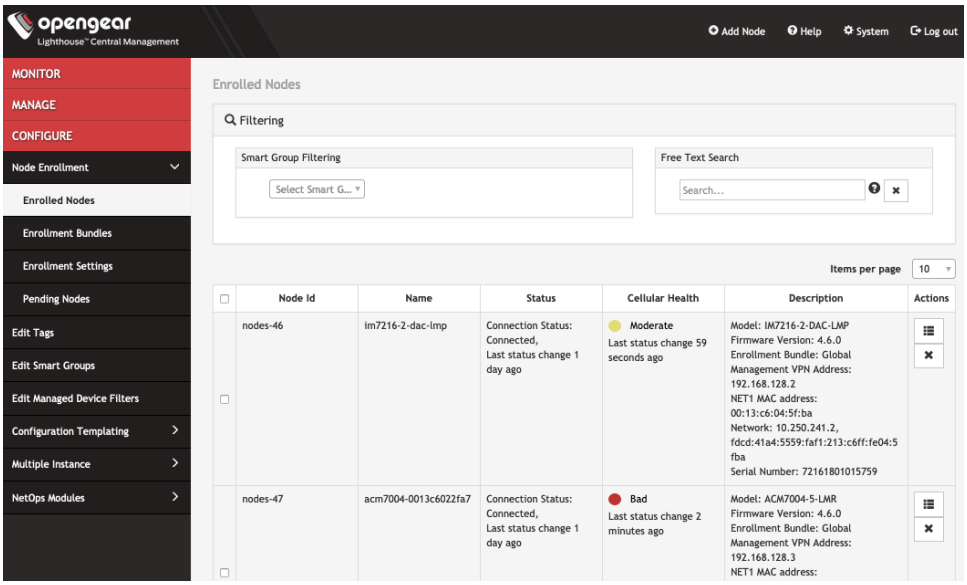
1. Select **CONFIGURE > Node Enrollment > Enrollment Bundles**. A list of existing **Enrollment Bundles** appears.
2. In the **Actions** column of the particular bundle, click the **download** button, a downward arrow in a circle.
3. Depending on the browser's configuration, a `manifest.og` file is either 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 > 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 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.

The **Cellular Health** column displays the node's current cellular status. If this state is **Good|Moderate|Bad**, the color indicator and the text are clickable links that open a popup containing detailed health information.

This popup includes the following information:

- Cellular IP Address (ipv4 and ipv6)
- Cellular interface status (Up|Down)
- Conditions
- Signal Quality
- RSSI
- Connection Type
- Sim Issues
- Connectivity Test (Passed|Failed|Connectivity Test Disabled)

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

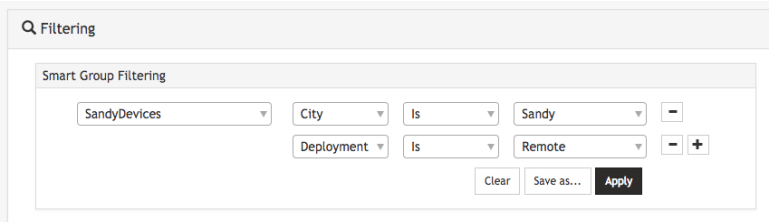
returns any nodes that have both *CM7148* AND *4.1.1* in searchable fields (e.g. *CM7148* in the name field and *4.1.1* in the firmware version field).

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:



In the example above, the **CONFIGURE > 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*.

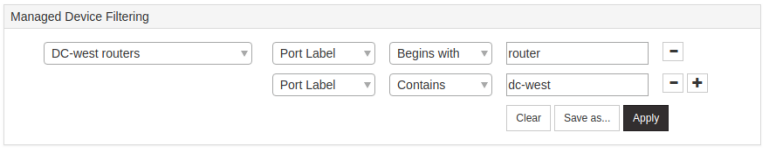
To add more filtering options:

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:



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.

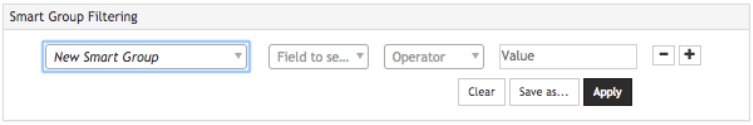
A given User Group can be linked to a particular Smart Group. When a Group is linked in this fashion, members of the Group inherit rights over all nodes in the group based on the Group's Role. See *Modifying existing groups* for how to set a Group's Role and Linked Smart Group.

Smart Groups can also be used to filter visible nodes on pages that display enrolled nodes (such as **CONFIGURE > 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 > 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.



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.

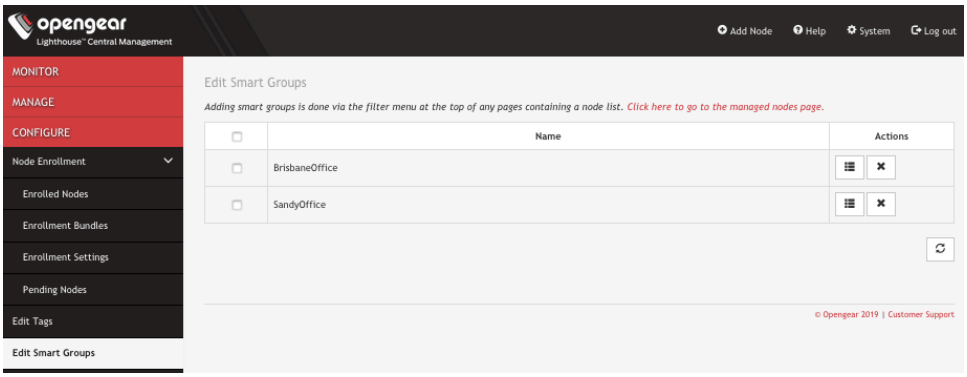
4. Click the **Operator** drop-down to select the operator to apply to the **Value**. In general, the **Is** operator is the most useful.
5. Select the **Value** to be matched against.
6. Click **Apply** to see the results of the filter.
7. Click **Save As** and type in a name for the search.

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:

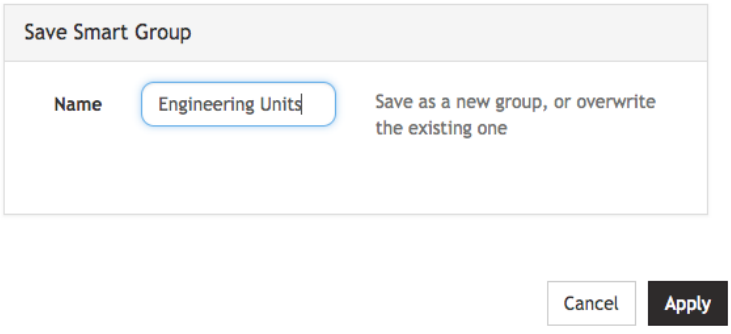
Select **CONFIGURE > Edit Smart Groups**.



- Click the **x** button to delete an existing Smart Group.
- Click the **Edit Group** button  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 > 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**.



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

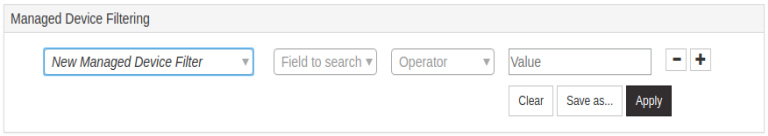
7.7 Creating Managed Device Filters

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.

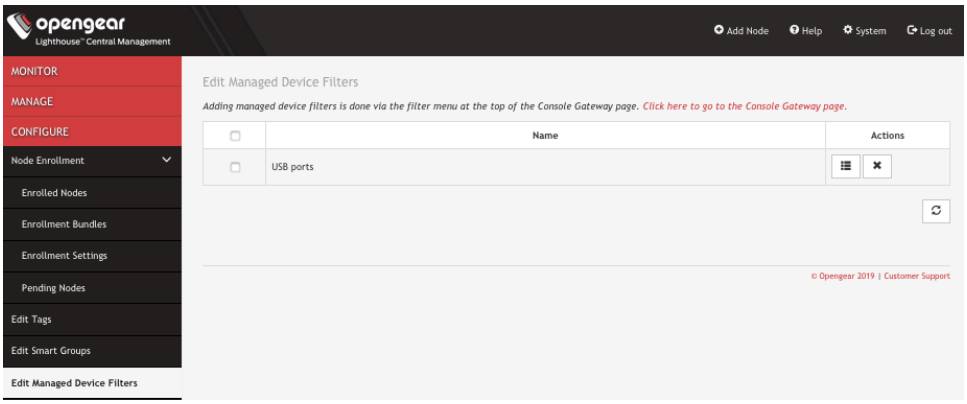


3. Click the **Field to search** drop-down to select a node attribute to filter on.
4. Select **Port Label** configuration.
5. Click the **Operator** drop-down to select the operator to apply to the **Value**. In general, the **Contains** operator is the most useful.
6. Populate the **Value** to be matched 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

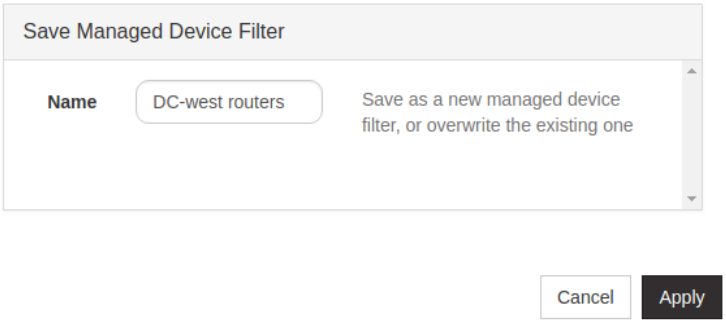
To edit an existing Managed Device Filter, select **CONFIGURE > Edit Managed Device Filters page**.



- Click the **x** button to delete an existing Managed Device Filter.
- Click the **Edit** button to change a Managed Device Filter's name.

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

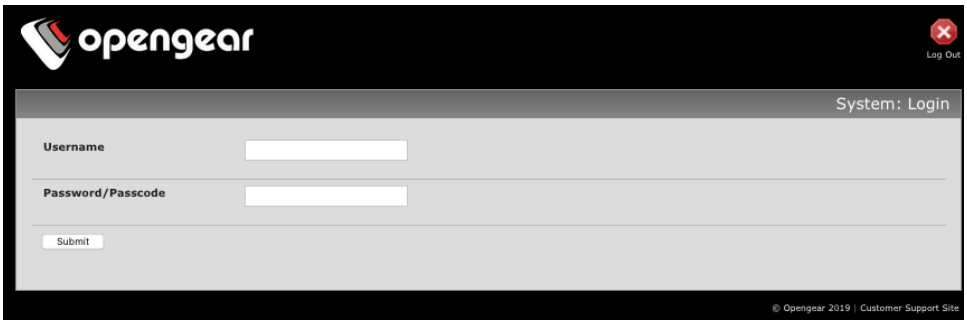
1. Navigate to a page that displays Managed Device Filter, such as **MANAGE > Managed Devices > Console Gateway**.
2. Select the Managed Device Filter to change from the **Select Managed Device Filter** drop-down menu.
3. Change the parameters (e.g. **Operator** values) as required.
4. Click **Save as**.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



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.



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 **+** button 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.

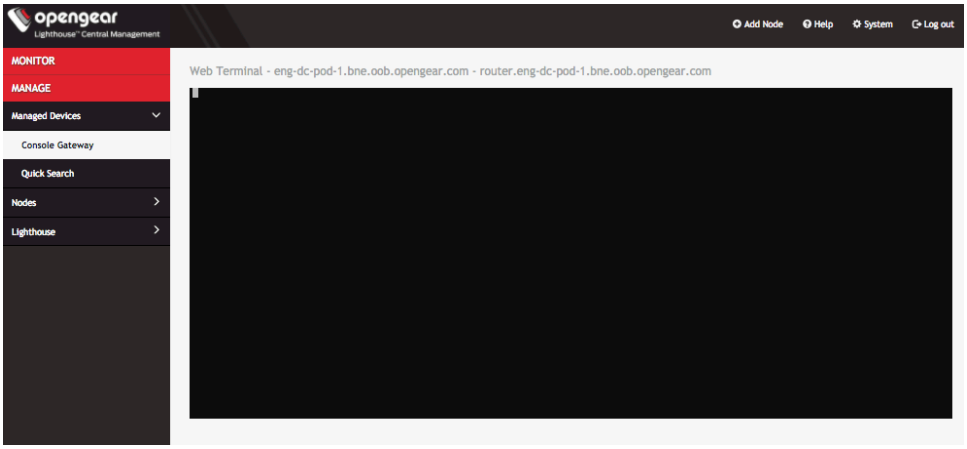
7.10.1 Access via HTML5 Web Terminal

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.

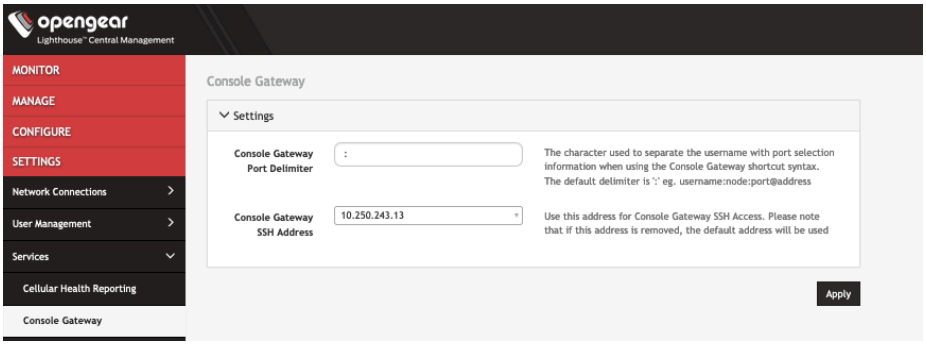
These auto-generated links use the colon (:) as the field-delimiter. The auto-generated SSH link has the following form:

```
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 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:

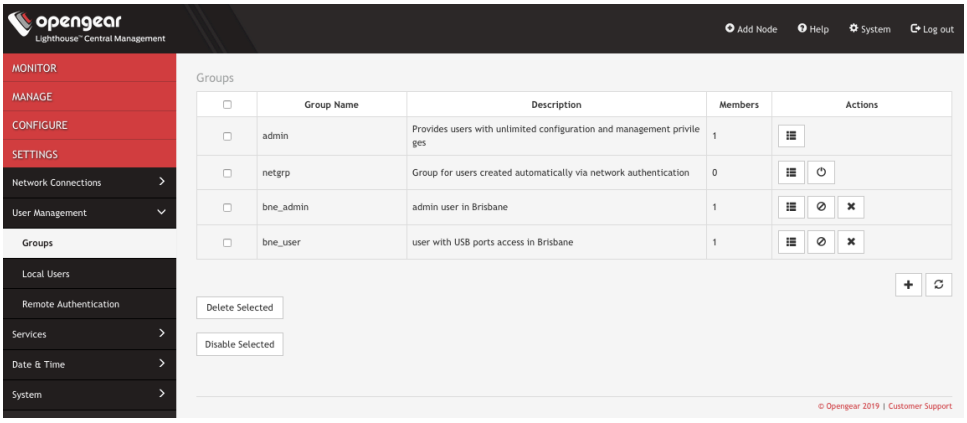
Role	Description
Lighthouse Administrator	The **Lighthouse Administrator** role is assigned to groups whose members need to manage and maintain the Lighthouse appliance. Members have access to all data on the Lighthouse system
Node Administrator	The **Node Administrator** role is assigned to groups that 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

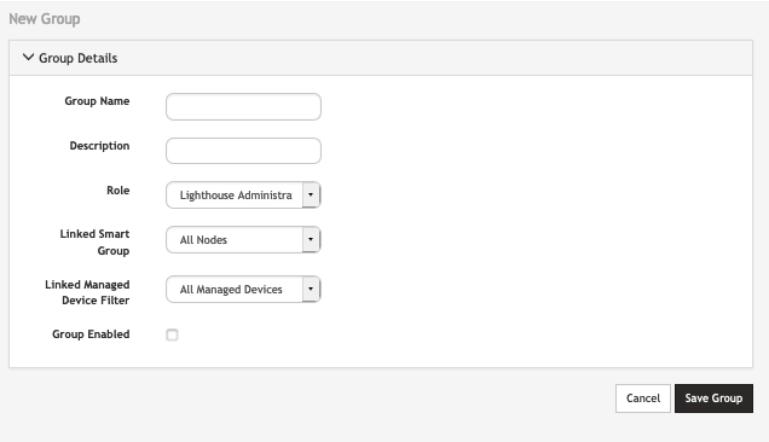
All password fields in Lighthouse are **write-only**. They accept data from the clipboard or pasteboard but do not pass data out.

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.

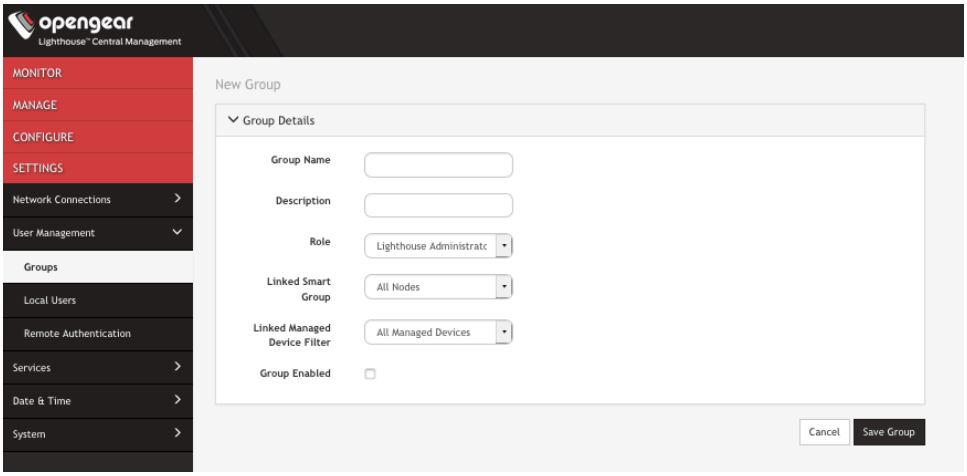
Group Name is case sensitive. It can contain numbers and some alphanumeric characters. When using remote authentication, characters from a user's remote groups that are not allowed on Lighthouse are converted to underscores during authentication. Local groups can be created that take that into account,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

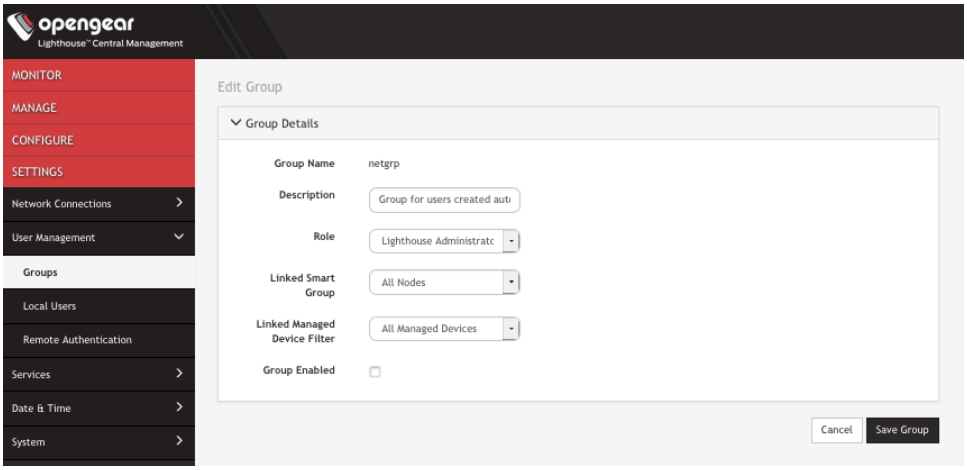


NOTE: When a new group is given the **Lighthouse Administrator** role, members of the group have access to the `sudo` command. Groups or users with the **Lighthouse Administrator** role are added to the **admin** group, which is in the list of allowed sudoers. On first boot of a new Lighthouse instance, 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 the **Edit Group** button ( ) 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.

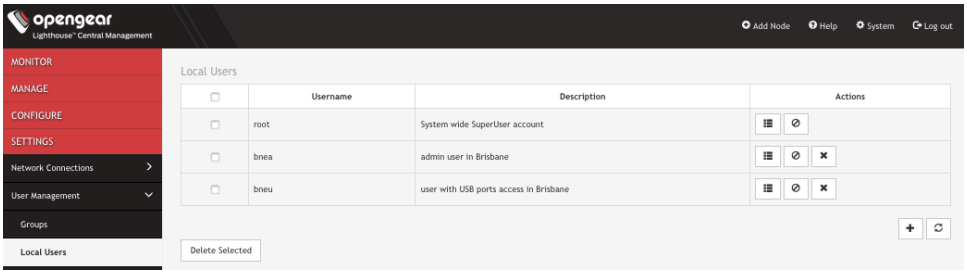
The **Groups** page also allows user to delete a group. All users who were members of the deleted group lose any access and administrative rights inherited from the group.

8.4 A note on default netgrp Lighthouse group

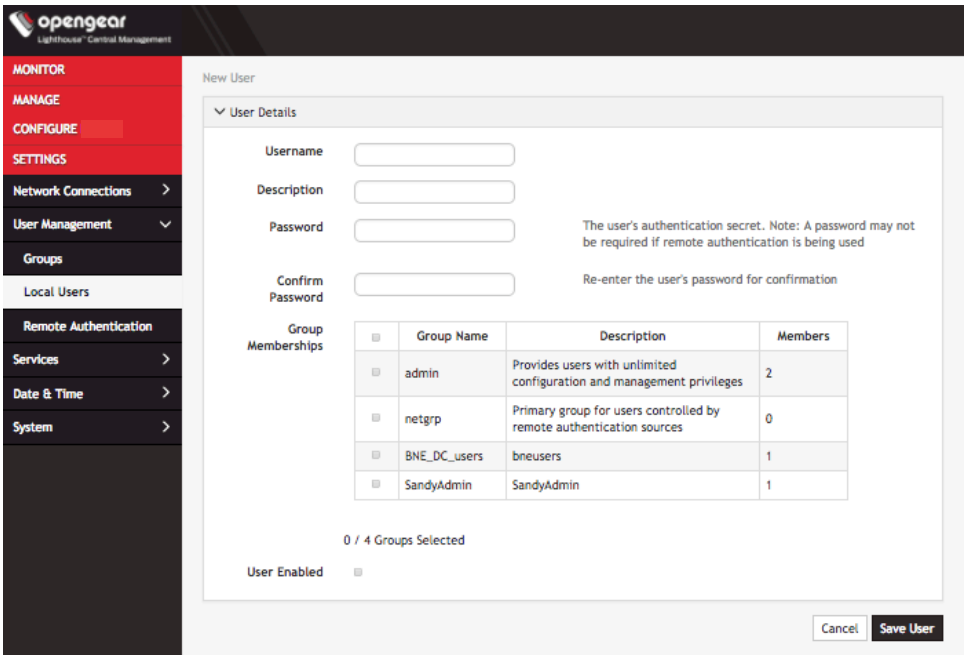
The **netgrp** group is inherited as the primary group for all remote AAA users who are not defined locally on Lighthouse. By default, **netgrp** has the **Lighthouse Administrator** role and is disabled - it must be enabled to take effect for remote AAA users.

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>
```

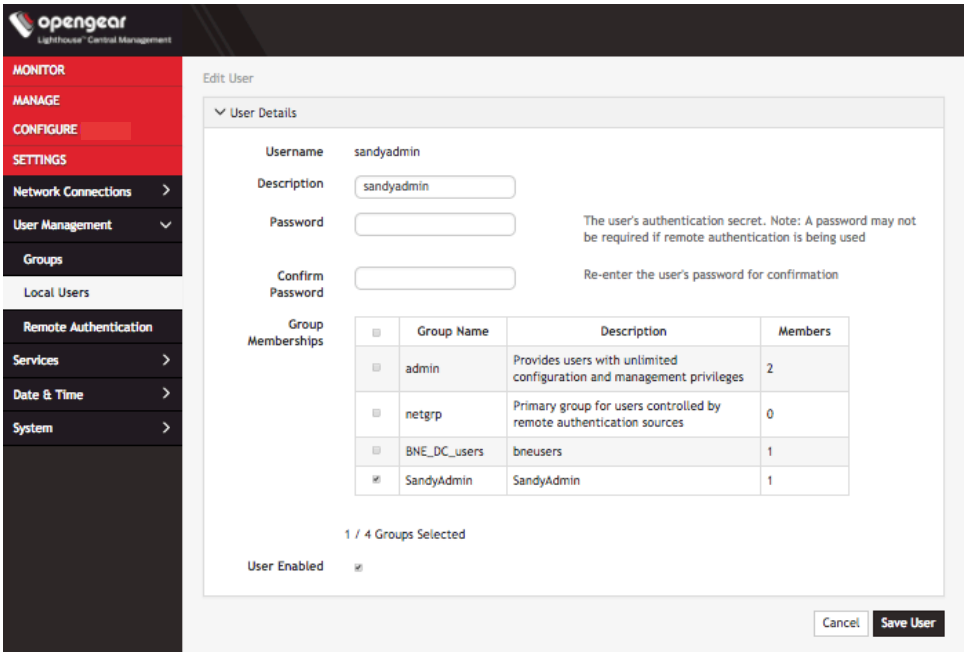
If the created user is set to disabled, the `configurator_users` message does not appear as they have not been added to the passwords file.

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 the **Edit** button ( ) in the **Actions** section of the user to be modified and make desired changes.
3. Click **Save User**.



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 the **Delete** button (**x**) in the **Actions** section of the user to be deleted.
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

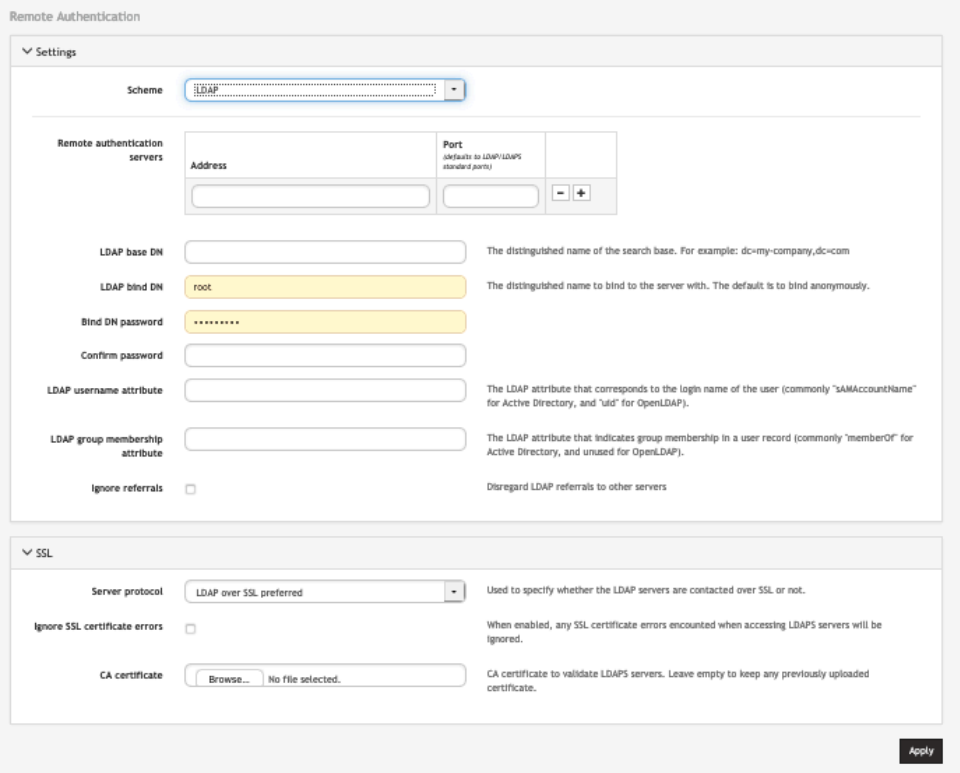
Lighthouse supports three AAA systems:

- 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



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 **LDAP 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 **LDAP Base DN** is **dc=ACME,dc=com**

4. Add the **LDAP 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 and confirm a password for the binding user.
6. Add the **LDAP username attribute**. This depends on the underlying LDAP system. Use **sAMAccountName** for Active Directory systems, and **uid** for OpenLDAP based systems.
7. Add the **LDAP 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.
9. Under the **SSL section**, choose the desired **Server protocol**.
   LDAP over SSL preferred: this will attempt LDAPS before trying LDAP without SSL
   LDAP (no SSL) only: non-SSL LDAP is always used
   LDAP over SSL only: LDAP over SSL is always used
10. If desired, check **Ignore SSL certificate errors** to ignore any SSL certificate errors.
11. **CA Certificate** is used to upload an SSL Certificate which will verify any LDAP servers you specify on the page.

NOTE: The certificate will be uploaded but will not be used if you've chosen to ignore certificate errors.

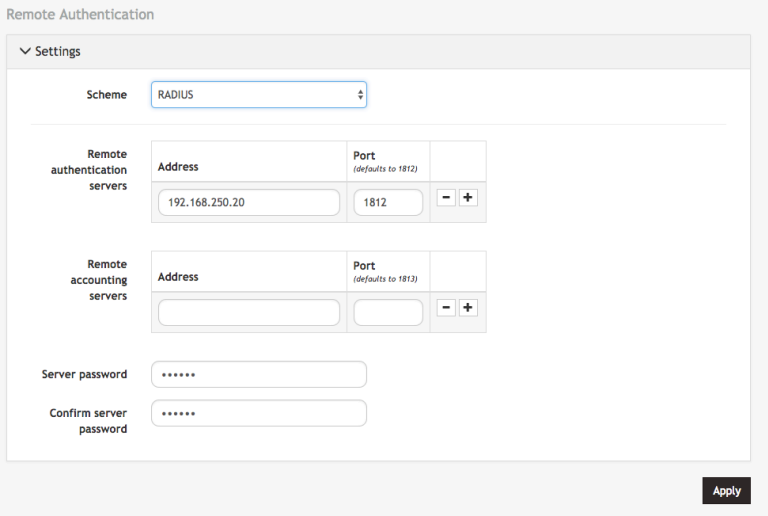
12. Install the **CA certificate** by clicking the **Browse…** button and locating the appropriate file.
13. Click **Apply**.

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**.



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:

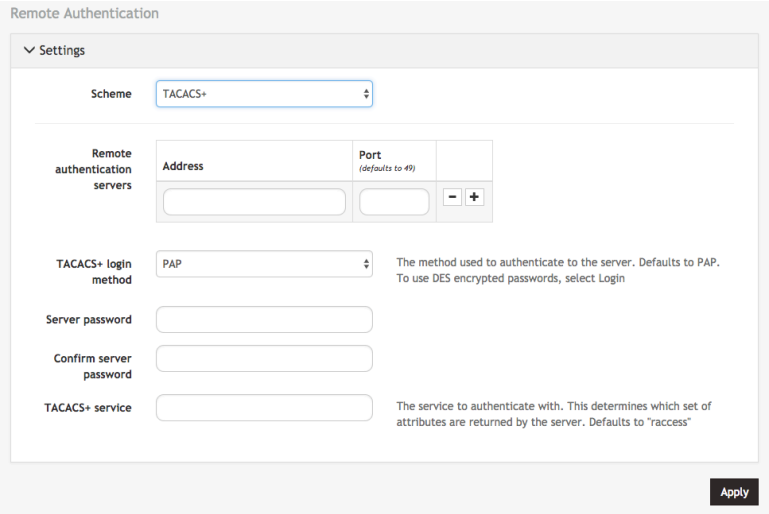
```
operator1 Auth-Type := System
      Framed-Filter-ID = ":group_name=west_coast_admin,east_coast_user:"
```

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**.



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 Opengear Help Desk.

9. Lighthouse central configuration

Templates are a centralized way of changing the configuration for enrolled Opengear 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 > Configuration Templating > Users and Groups Templates** to create, edit, and delete users and groups templates. Each template must contain at least one group.

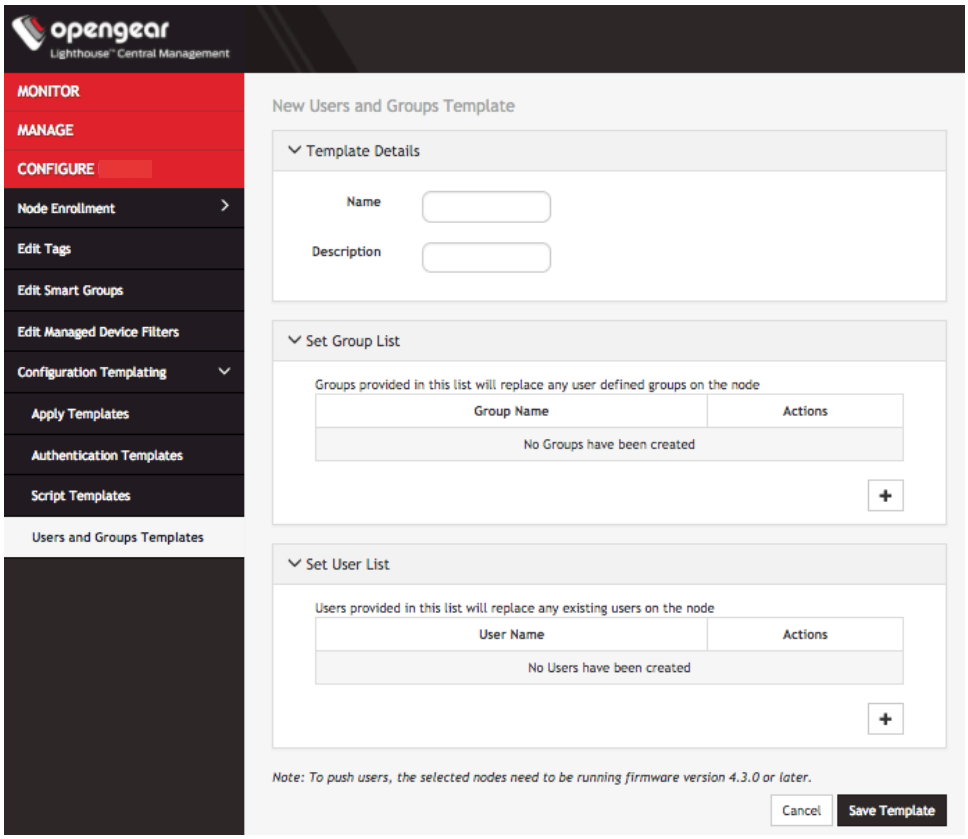
Each template contains a list of user-defined groups and/or individual users. Each group has a defined role which determines what privileges group members have. User roles are defined by the groups they are a member of.

The available group roles are:

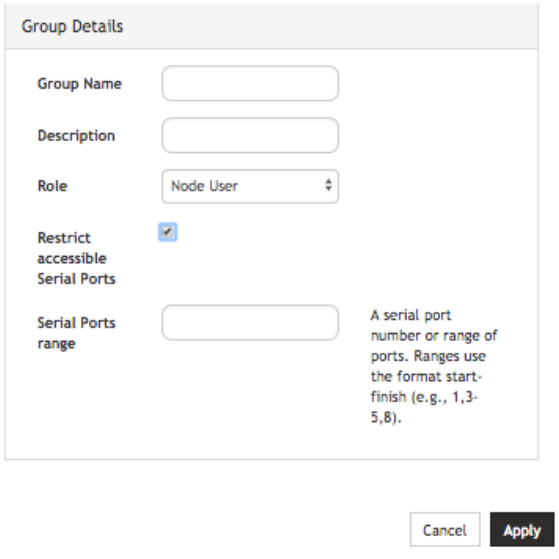
- **Node Administrator** — maps to the administrator role on the nodes.
- **Node User** — maps to the ports user role and the pmshell role on the nodes. Ports access can be restricted if required.

To create a new users and groups template:

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

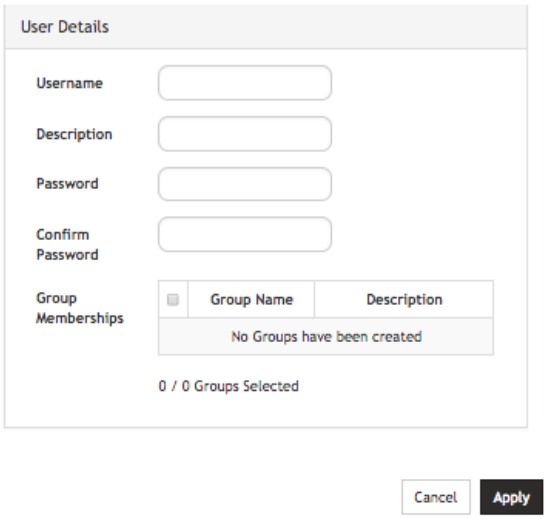


3. Enter a **Name** and **Description** for a template in the **Template Details** section.
4. Click the **+** button in the **Set Group List** section to add a new group. The **Group Details** dialog loads.



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.



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**.

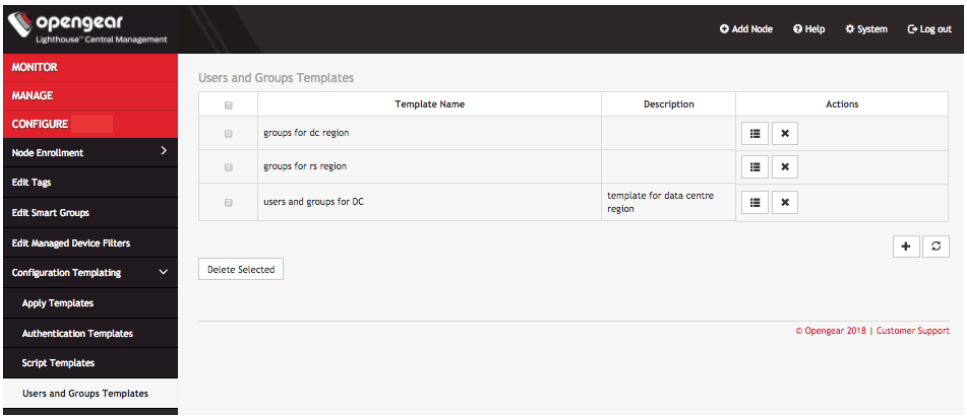
NOTE: When a **users and groups template** is pushed to a node, all custom groups on that node are replaced by groups defined in the template. If no users are in the new template, existing users will remain on the node. To push users, the selected nodes need to be running firmware version 4.3.0 or later.

9.2 Modifying existing users and groups templates

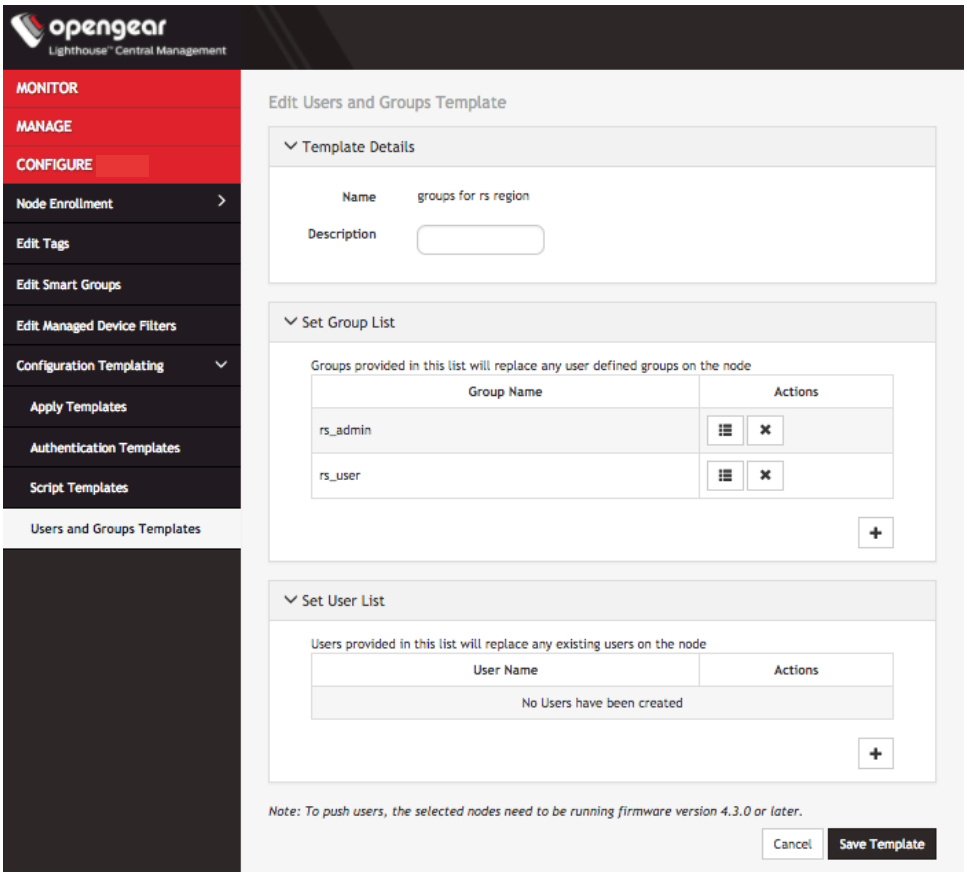
The **Edit Users and Groups Template** dialog allows a template's **Description, Group List,** and **User List** to be set and changed.

To modify a template:

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



2. Click **Edit button** 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 > 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 > 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.

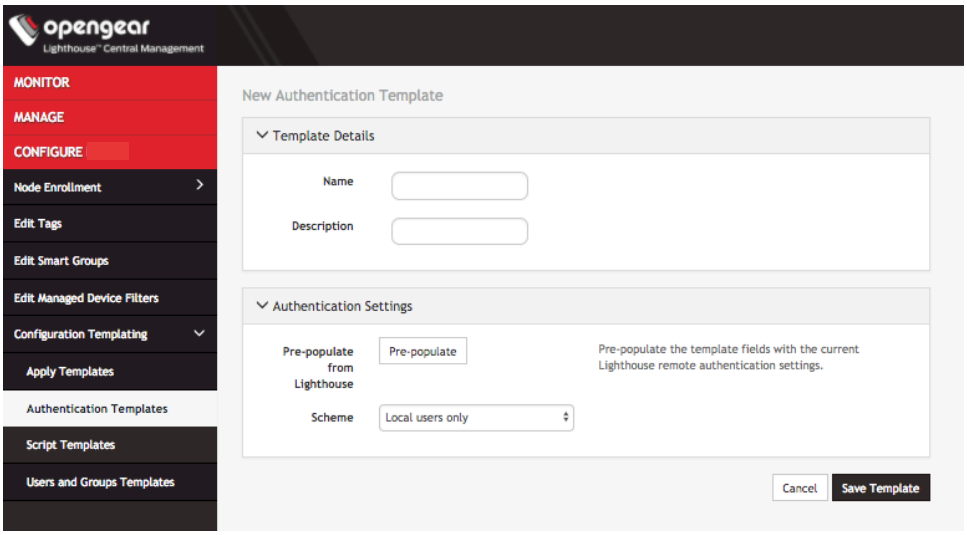
9.5 Creating new authentication templates

Only users assigned to the **Lighthouse Administrator** role can access **CONFIGURE > 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 > Configuration Templating > Authentication Templates**.
2. Click the **+** button. The **New Authentication Template** dialog loads.



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.

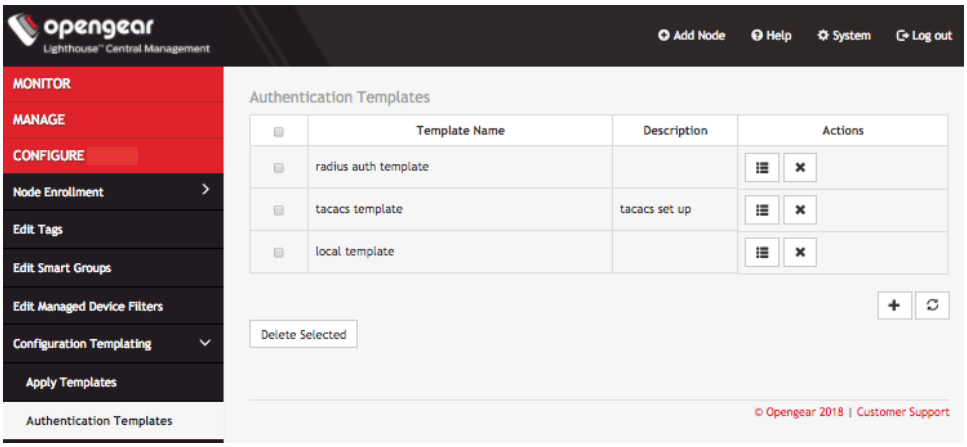
NOTE: The authentication templates do not support the full list of settings that the Opengear console servers support. However, templates can be applied, and then additional settings configured manually.

9.6 Modifying existing authentication templates

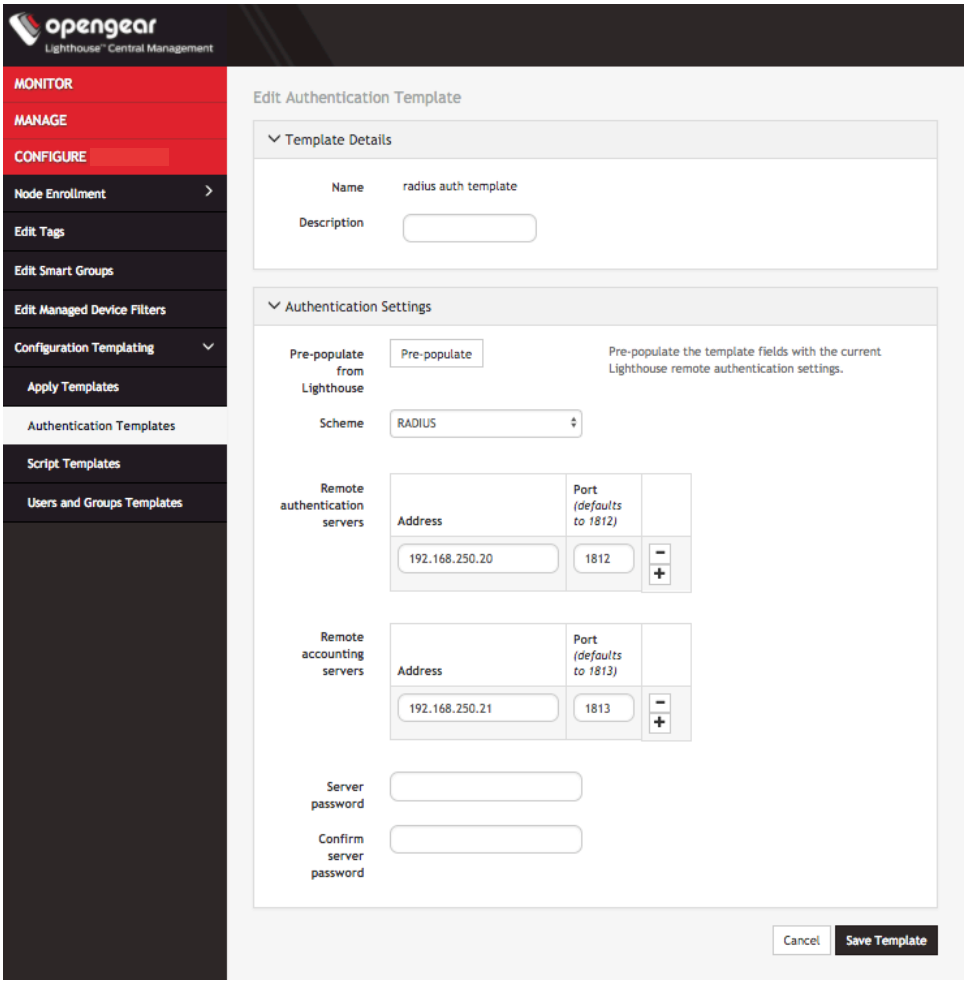
The **Edit Authentication Template** dialog allows the template's **Description** and **Authentication Settings** to be set and changed.

To modify an existing authentication template:

1. Select **CONFIGURE > 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 > 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.

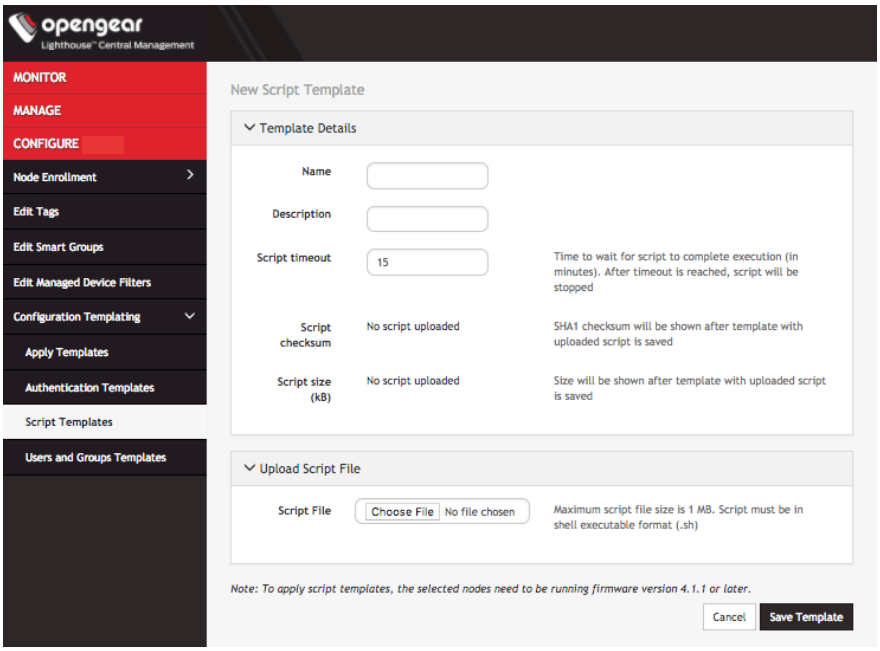
9.8 Creating new script templates

Users assigned to the **Lighthouse Administrator** role can access **CONFIGURE > 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 > Configuration Templating > Script Templates**.
2. Click the **+** button. The **New Script Template** dialog loads.

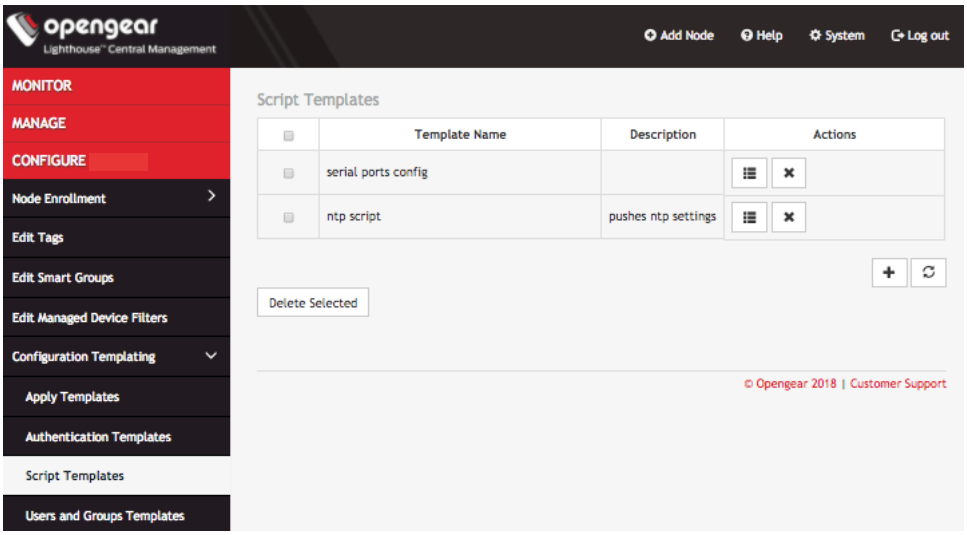


3. Enter a **Name** and **Description** for a template in the **Template Details** section.
4. To select a script to upload, click **Choose file**.
5. Click **Save Template**. **Script checksum** and **Script size** are shown after template with uploaded script is saved.

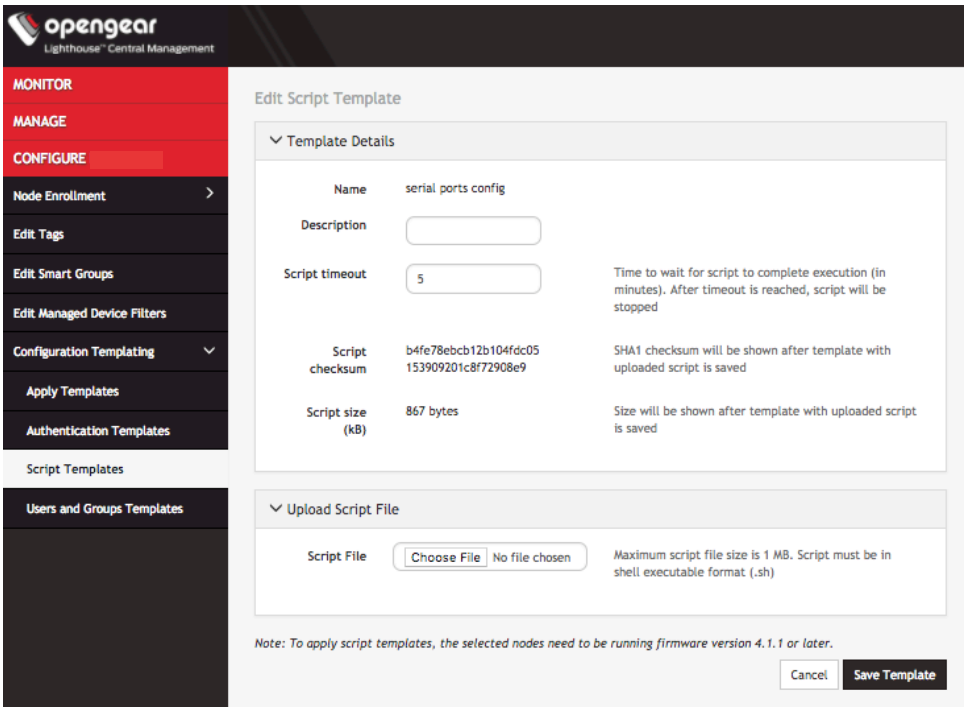
9.9 Modifying existing script templates

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 > Configuration Templating > Script Templates**.



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

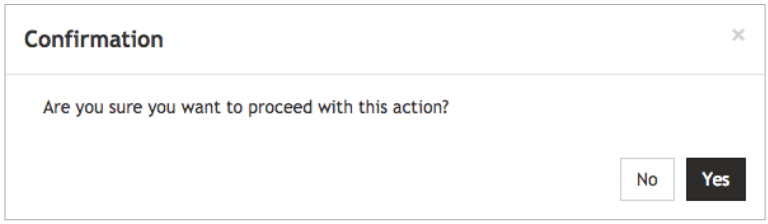


3. Make required changes.
4. Click **Save Template**.

9.10 Deleting script templates

To delete a script template completely:

1. Select **CONFIGURE > Configuration Templating > Script 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 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 > Configuration Templating > Apply Templates** and execute templates affecting any node.

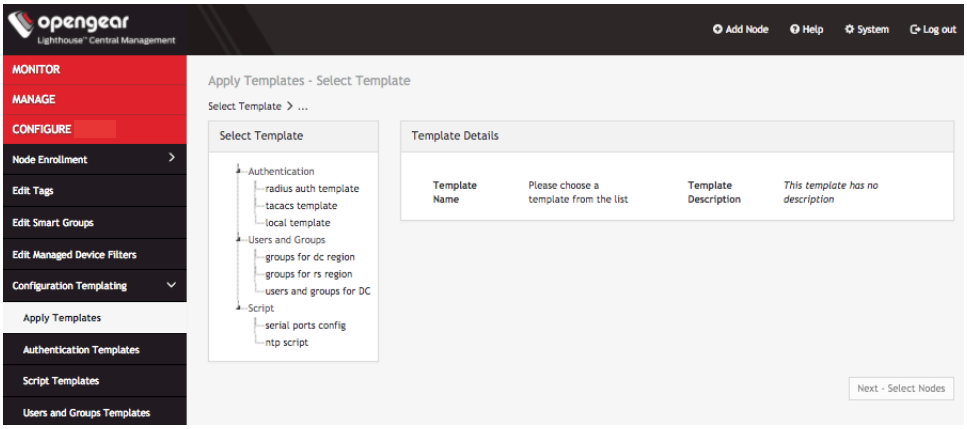
Users with Node Administrator privileges (i.e. users with the Node Administrator role or users who are members of groups with the Node Administrator role) can access **CONFIGURE > Configuration Templating > Apply Templates** and execute templates affecting nodes in Smart Groups linked to their role.

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

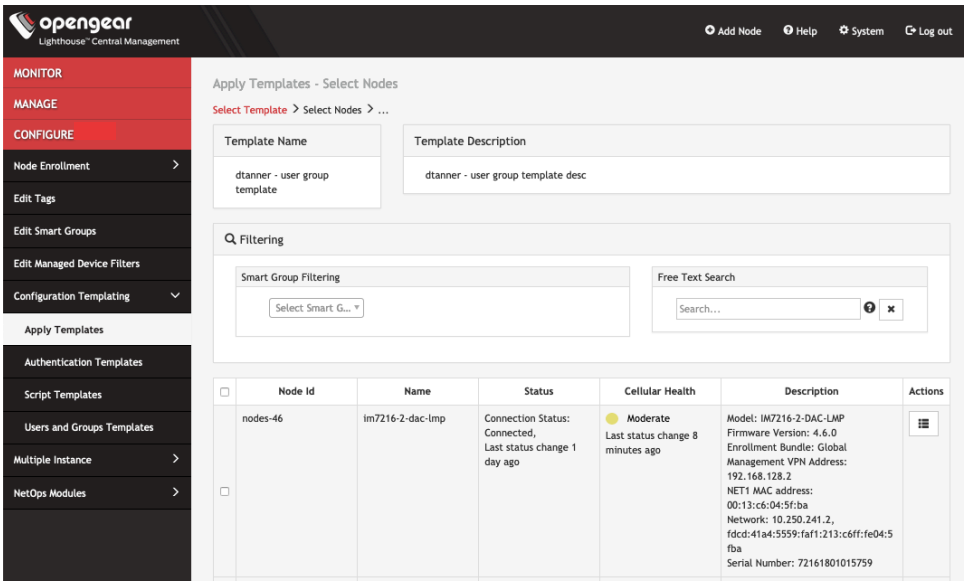
1. Select Template.
2. Select Nodes.
3. Preflight. This test run simulates what happens if the template is pushed to the selected nodes.
4. Execution.

To apply a template:

1. Select **CONFIGURE > 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.



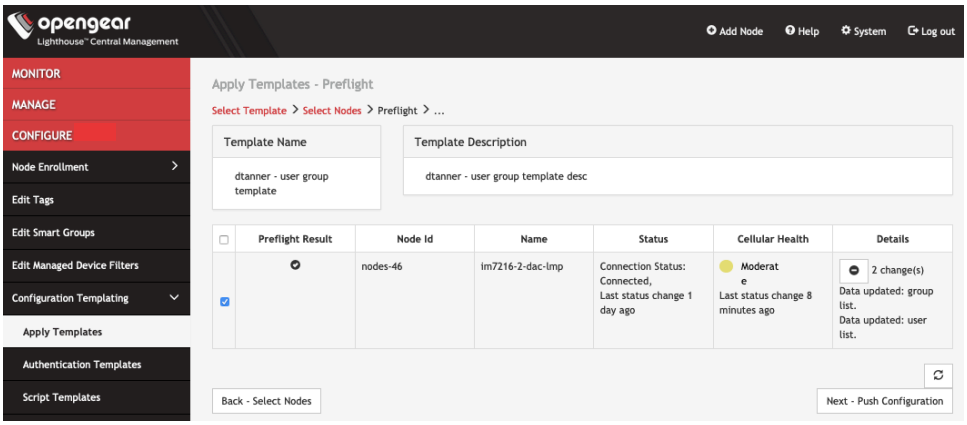
4. Select nodes from the list of enrolled nodes. **Smart Group Filtering** and **Free Text Search Filtering** can be used to narrow down the results.

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.

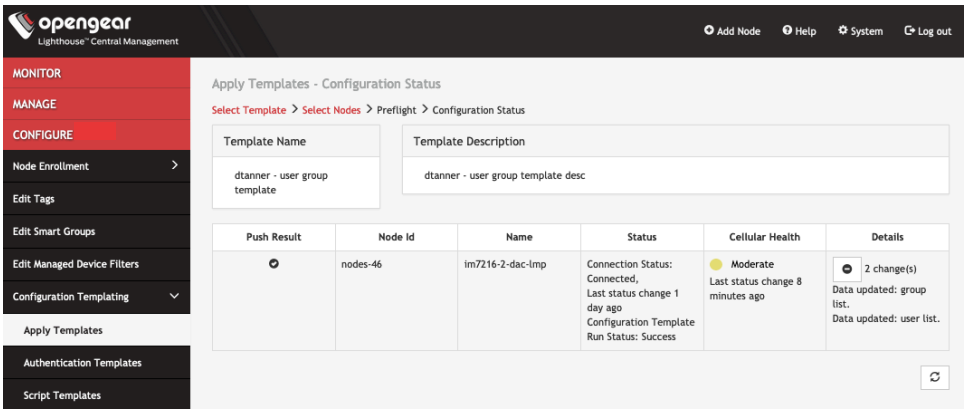
5. Click **Next — Preflight**. The **Preflight** stage loads. This stage requires manual refresh to retrieve updated **Preflight Result** and **Details**.

   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**.

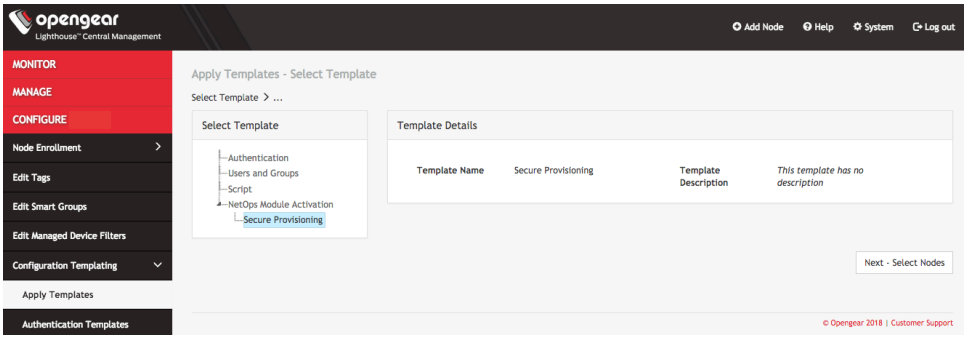
After all nodes finish the template push, a success message appears.



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 > 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. Multiple Instance

This chapter discusses licensing, setup, configuration, promoting and disconnecting secondary instances, and upgrading a multiple instance Lighthouse.

10.1 Licensing

Multiple instance functionality requires the installation of a valid license with the multiple instance feature. This license must only be installed on the primary Lighthouse instance.

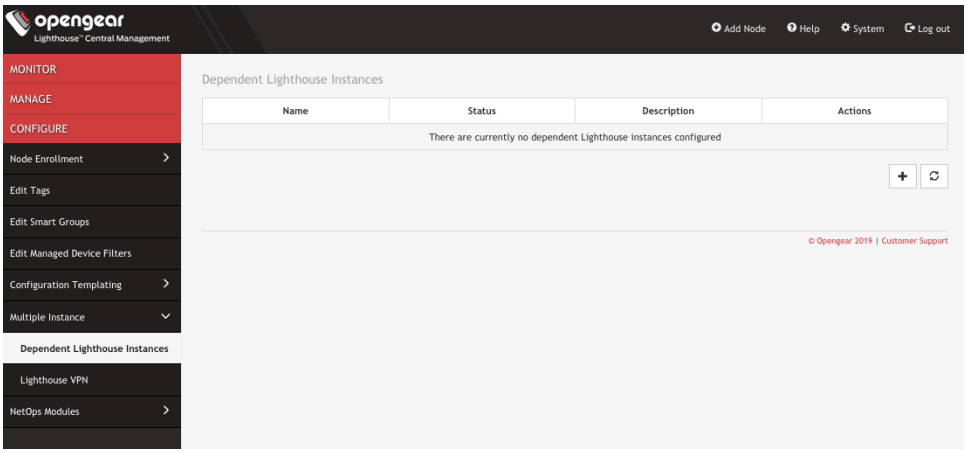
If a multiple instance license is not installed:

- The dependent Lighthouse instances page will display a banner on an empty page
- The multiple instance Lighthouse VPN page will display a banner but will allow the user to modify the default VPN settings in case it conflicts with their network.

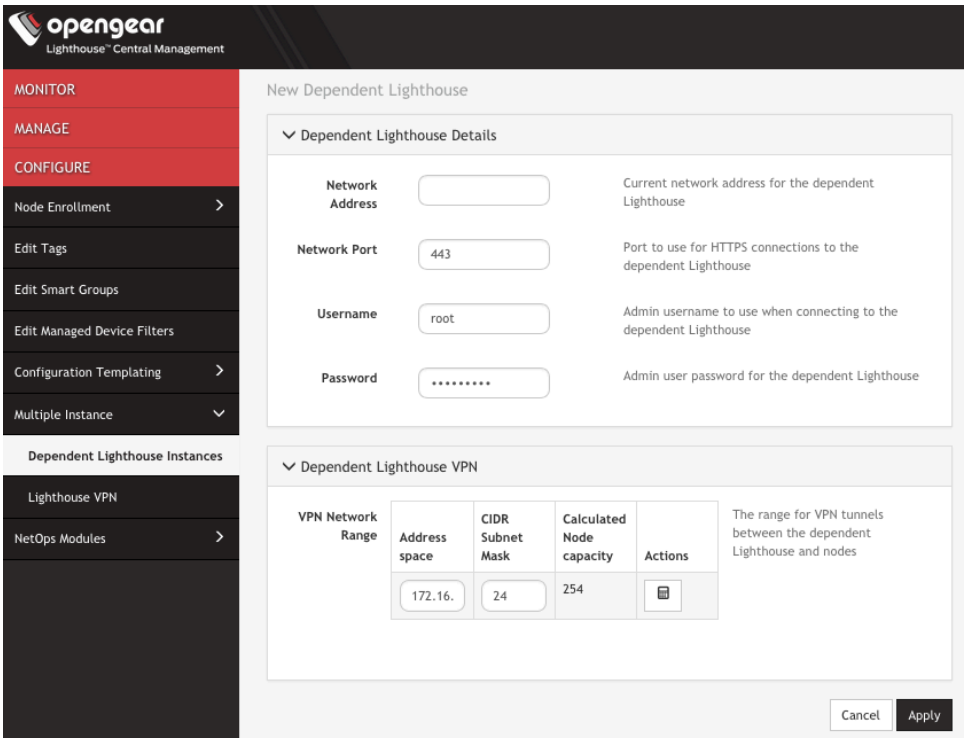
10.2 Setting up a multiple instance

Lighthouse supports up to 10 secondary instances for each primary.

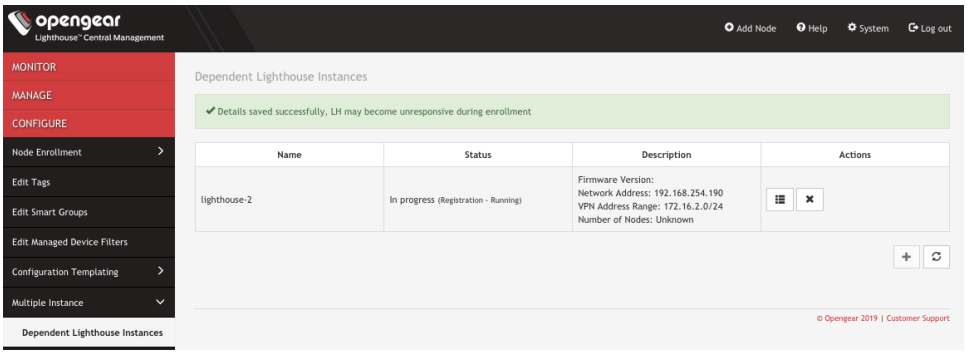
1. Start with what will be the primary instance and one or more Lighthouse instances to act as secondary. All instances must have the same version of Lighthouse. To support more than one instance, you must use 19.Q3 or later.
2. Configure the networking information for each instance (hostname, external endpoints, network addresses).
3. Configure time settings of each instance.
4. Install a license with the multiple instance feature on the primary Lighthouse.
    a. On the primary Lighthouse, click **Configure > Multiple Instance > Dependent Lighthouse Instances.**



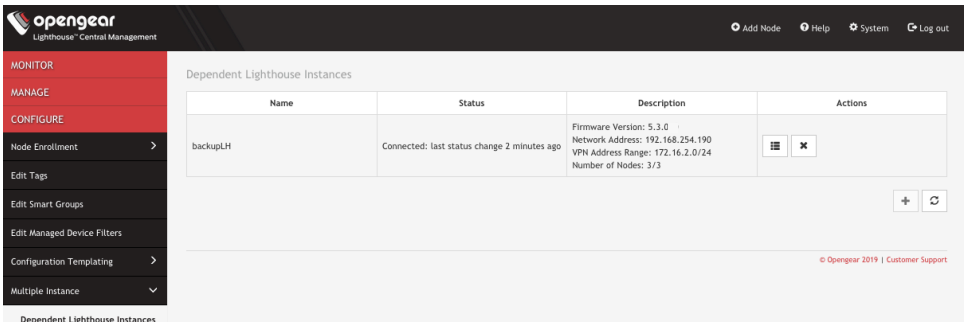
    b. Click **Add**. Enter the network address, username and password of a Lighthouse instance to enroll as secondary. Optionally enter a valid, unused network subnet to use as the dependent lhvpn address range. If none is entered, a default will be assigned.



5. Dependent Lighthouse enrollment will show status as is moves from Pending > Registered > Enrolled.



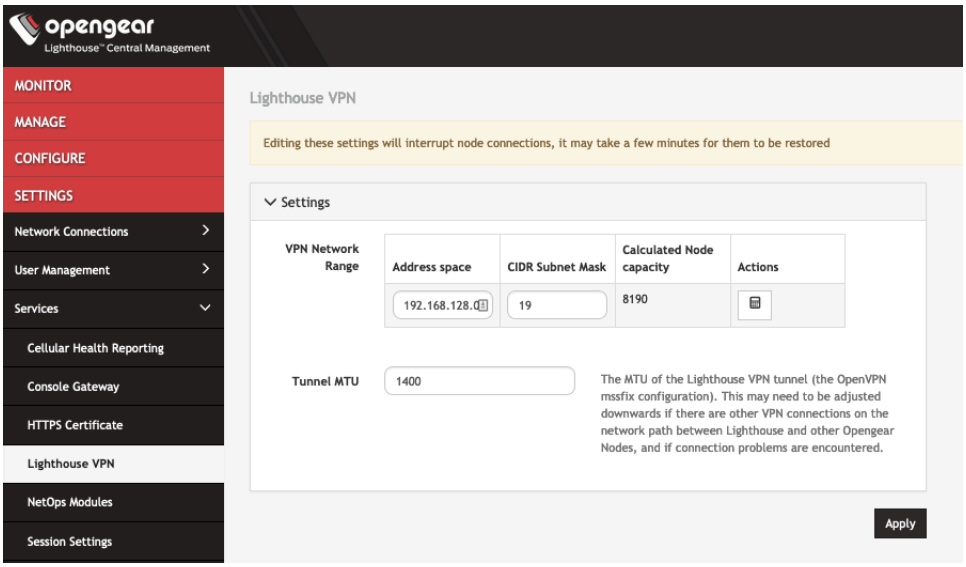
6. When the VPN connection is established between primary and secondary Lighthouse, this page will display **Connected** with the time since the last status change and **Disconnected** when the connection is lost. Any errors in the enrollment process will be displayed in the status column.



10.3 Multiple instance configuration

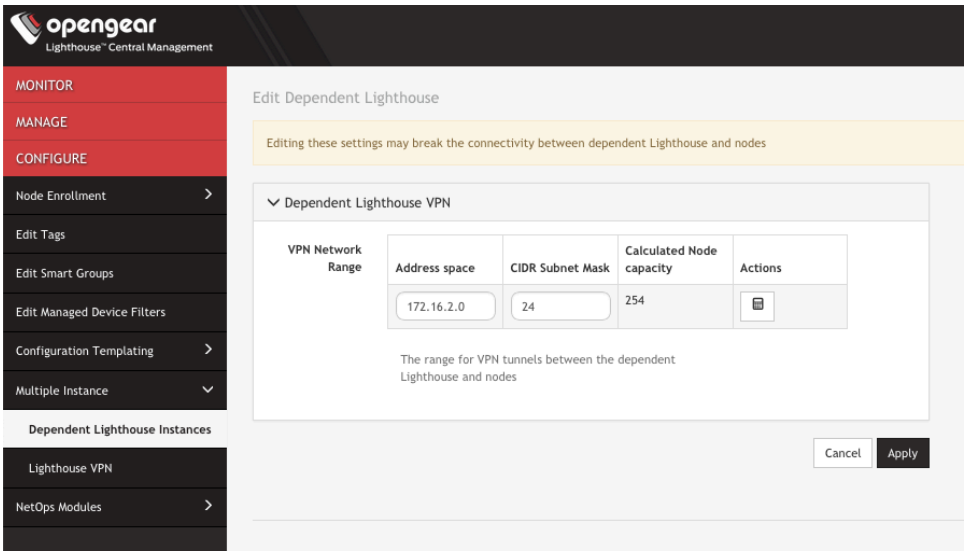
Lighthouse with multiple instance support requires multiple separate subnets for Lighthouse VPN connections: between each instance and its nodes, and between the primary and dependent Lighthouses. Each subnet must not overlap any subnet in use by another Lighthouse instance.

The subnet between the primary Lighthouse and its nodes is modified under **Settings > Services > Lighthouse VPN** on the primary Lighthouse. Click the button under **Actions** to calculate the addressable nodes based on the network address and CIDR mask.

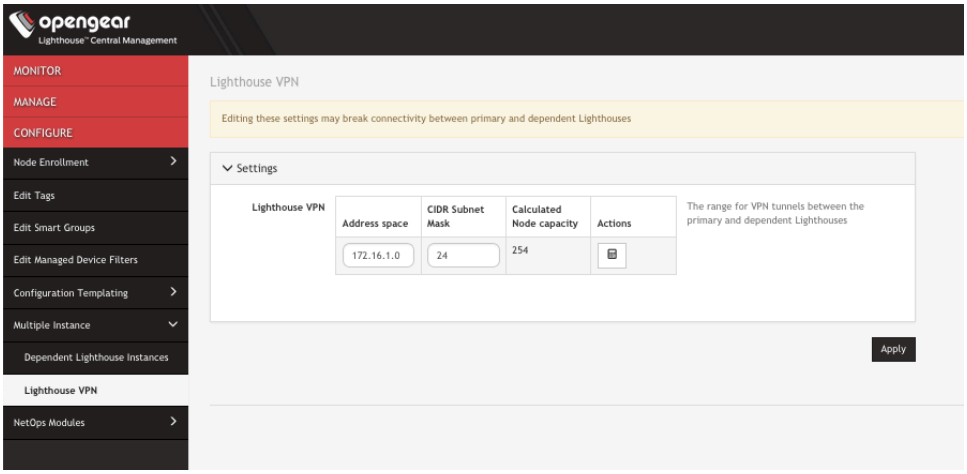


A secondary Lighthouse is read-only and cannot be modified. The **Settings > Services > Lighthouse VPN** page displays the subnet used by this Lighthouse instance, but it cannot be modified directly.

The subnet between each secondary Lighthouse and its nodes can be modified on the primary Lighthouse under **Configure > Multiple Instance > Dependent Lighthouse Instances > Edit**.



The subnet between the primary Lighthouse and dependent Lighthouse instance can be modified on the primary Lighthouse under **Configure > Multiple Instance > Lighthouse VPN**



Other information that is specific to dependent Lighthouse should be configured before enrolling but can be modified on the primary Lighthouse via `ogconfig-cli`.
Instance specific information includes:

- hostname
- time zone
- networking
- external interfaces

The instance specific information is present on both Lighthouses but read-only on the secondary Lighthouse. Both configurations can be viewed via `ogconfig-cli`.

Primary Lighthouse configuration is stored in lighthouse_configurations[0]
Secondary Lighthouse configuration is stored in lighthouse_configurations[1]

View all secondary Lighthouse instance specific configuration (can be run on either Lighthouse instance):

```
ogconfig-cli
print lighthouse_configurations[1]
```

You can modify secondary configuration from primary Lighthouse. For example, to update the hostname of the secondary Lighthouse, run the following commands on the Primary Lighthouse:

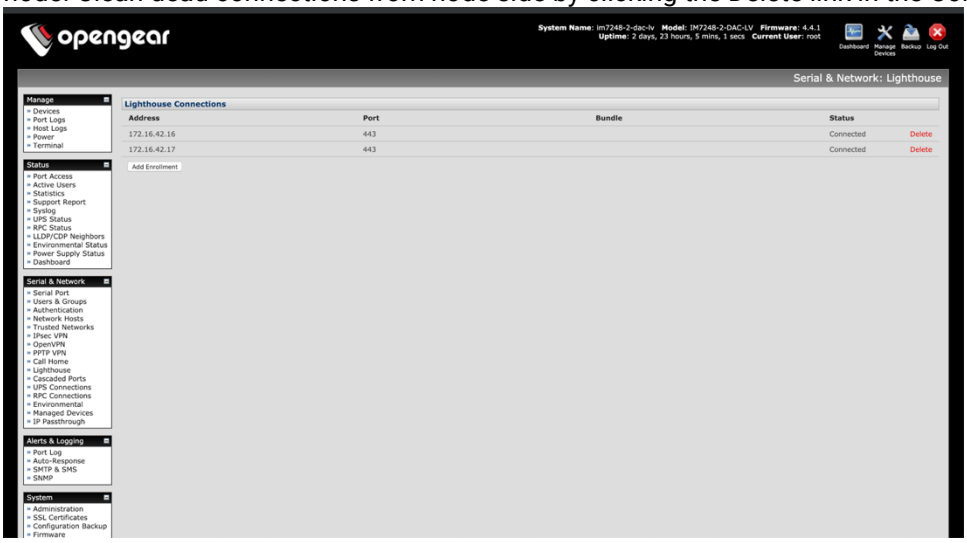
```
ogconfig-cli
set lighthouse_configurations[1].hostname new_name
push
```

10.4 Disconnecting a secondary instance

Dependent Lighthouse instances can be removed from the primary Lighthouse. To do so, click **Configure > Multiple Instance > Dependent Lighthouse Instances**, and click the **x** button under **Actions** next to the instance.

The secondary Lighthouse will begin unenrollment, which will factory reset the secondary Lighthouse. A user will be required to enter a new root password via console when it reboots.

You will need to manually remove the connection to the secondary Lighthouse from each connected node. Clean dead connections from node side by clicking the Delete link in the Console Server.



10.5 Promoting a secondary instance

When a primary Lighthouse is no longer reachable, a secondary Lighthouse instance can be promoted to primary. The new primary can then be used to enroll a secondary Lighthouse if required.

NOTE: This should only be performed if the primary Lighthouse has no chance of returning, the procedure is not reversible and will break all node connections with the previous primary instance. The previous primary instance must be factory reset before it can be used again.

To promote a secondary instance to primary, login as root on the secondary instance via console or ssh and run

```
promote-secondary-lighthouse
```

You will need to remove all dead connections from node side from the Console Server. The Promotion tool deletes connection between primary and secondary instance but does not touch node connections.

The new primary can then be used to enroll a secondary Lighthouse if required.

NOTE: If the previous primary becomes accessible again, it will not be able to connect to its enrolled nodes or the previous secondary Lighthouses.

10.6 Upgrading a multiple instance Lighthouse

To upgrade a Multiple Instance Lighthouse:

When the primary Lighthouse is updated, any secondary Lighthouses will be updated in a rolling fashion after the primary has successfully booted. If any Lighthouse fails to successfully update along the way, the update will stop.

If the auto update fails, a manual update can be performed.

To upgrade a Lighthouse running version 5.3.0 with enrolled dependent instances, a script called 'sysflash_mi_upgrade_bootstrap.sh' which is provided with the Lighthouse 2019.Q2 release must be used.

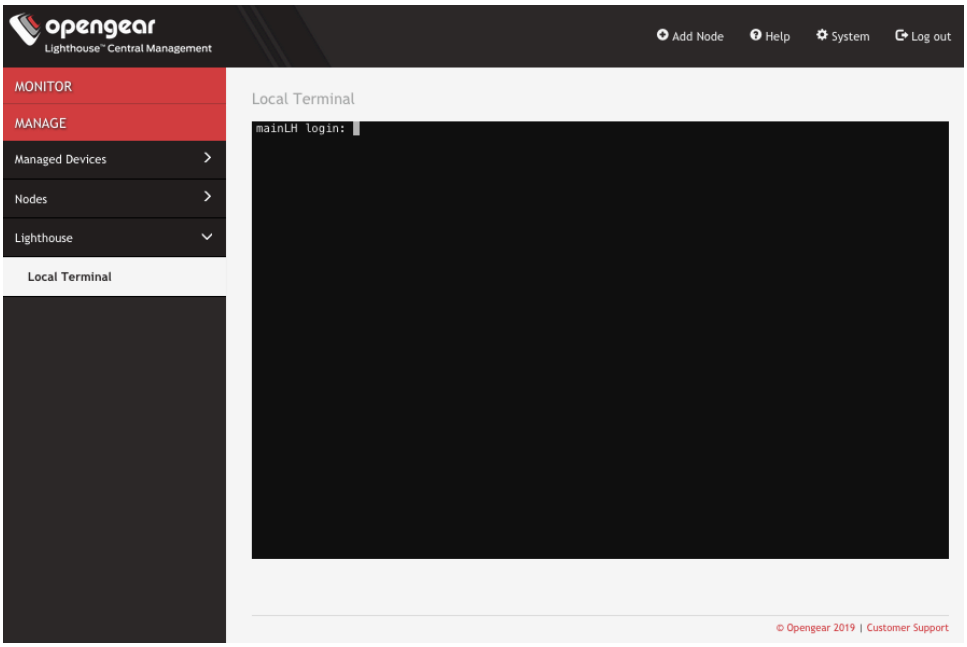
To upgrade the secondary independently of the primary:

1. Upload sysflash_mi_upgrade_bootstrap.sh script to the secondary Lighthouse instance.
2. Ensure that the script is executable
3. Run the command as you would with sysflash. It can be passed a URL to the upgrade file or a path to the file on the box.

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:

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

11.4.7 Support for multiple instance Lighthouse with `ogconfig-cli`

Configuration system information can be displayed, searched, and set from both the primary and secondary Lighthouse instances. To reference the primary instance, use `lighthouse_configurations[0]`. The secondary instance is reachable with `lighthouse_configurations[1]`.

For example, to display nodes all network connections to the primary Lighthouse, use:

```
ogcfg> print lighthouse_configurations[0].system.net.conns
```

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:

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

11.6 cron

The cron service can be used to schedule file execution at specific times. Daemon can be managed via the /etc/init.d/crond interface, and cron tables managed via crontab.

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

Options:

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

To perform start/stop/restart on crond service:

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

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 contain one command to run. The following format is used:

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

For example, the following entry will run a the specified backup.sh script every day at 3am:

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

When finished, save and close the crontab 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:

`-b, --board-name <name>`	Override board name (currently `lighthouse-vm`)
`-B, --board-revision <version>`	Override board revision (currently 1.0)
`-V, --vendor <vendor>`	Override vendor (currently `opengear`)
`-I, --no-version-check`	Do not check software version for upgradability
`-m, --no-migration`	Do not migrate current config. Start fresh.
`-v, --verbose`	Increase verbosity (may repeat)
`-o, --no-boot-once`	Do not modify bootloader (implies `--no-reboot`)
`-r, --no-reboot`	Do not reboot after upgrading
`-h, --help`	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.

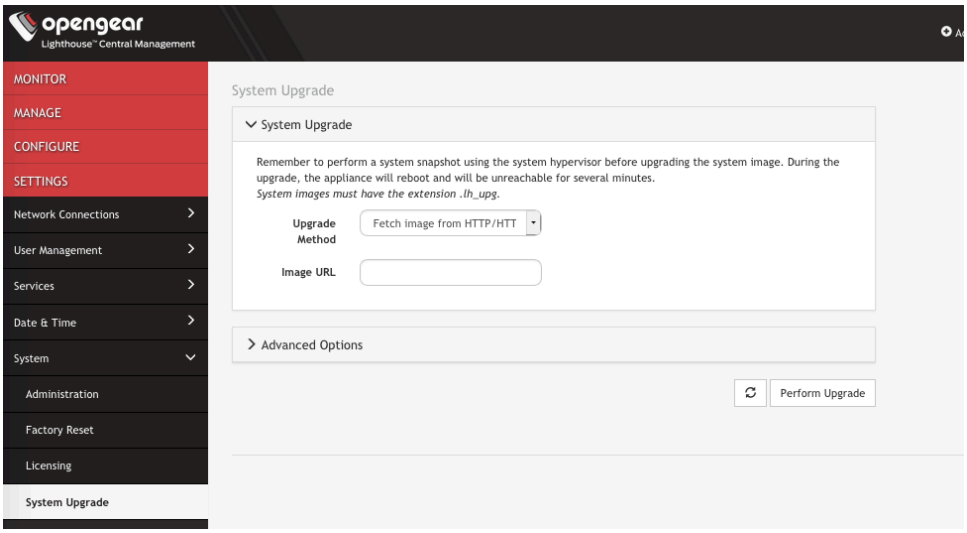
NOTE: Although upgrades do not overwrite existing configurations or user files, you should perform a Configuration Backup prior to upgrading.

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 HTTP 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 Opengear Support.

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

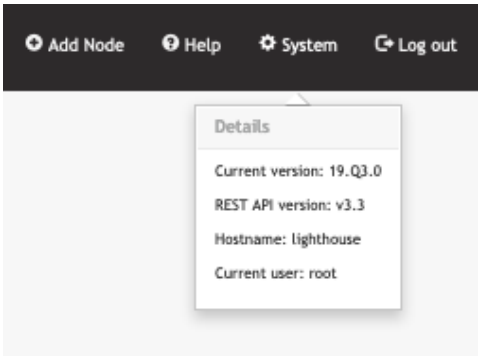
- `sysflash -h` or
- `sysflash --help`
- The same listing is presented in the `sysflash` entry of the Command line tools chapter above.

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
2019.Q3.0
```

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
2019.Q3.0
```

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 2019.Q3.0 \n \l
```

13.2 Technical support reports

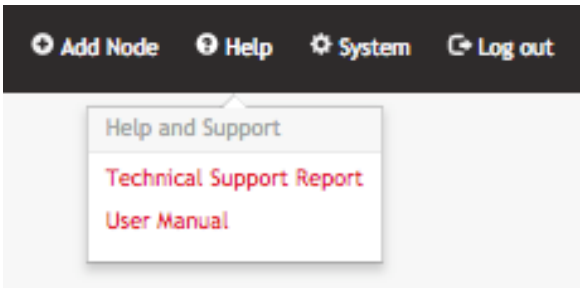
Lighthouse can generate a technical support report that includes Lighthouse configuration information and the current system log for the Lighthouse VM.

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

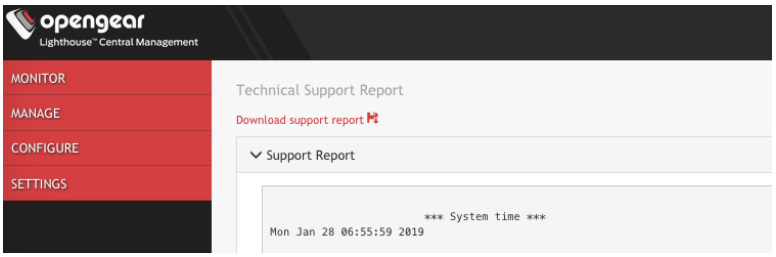
To generate a complete configuration and status report regarding a given Lighthouse VM:

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.

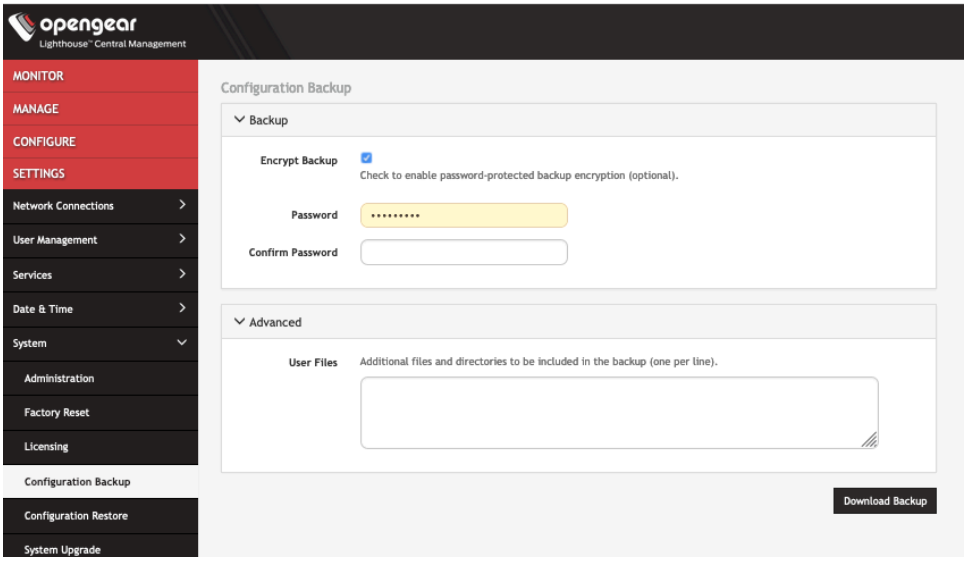
- Use the FTP client on Lighthouse to copy the file to an FTP server. Passive mode must be used for this to work. Example:

```
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- *** Opengear 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 Configuration Backup

Before performing a factory reset or system upgrade, you may want to backup the current Lighthouse configuration. To do so:

1. Select **Settings > System > Configuration Backup**.

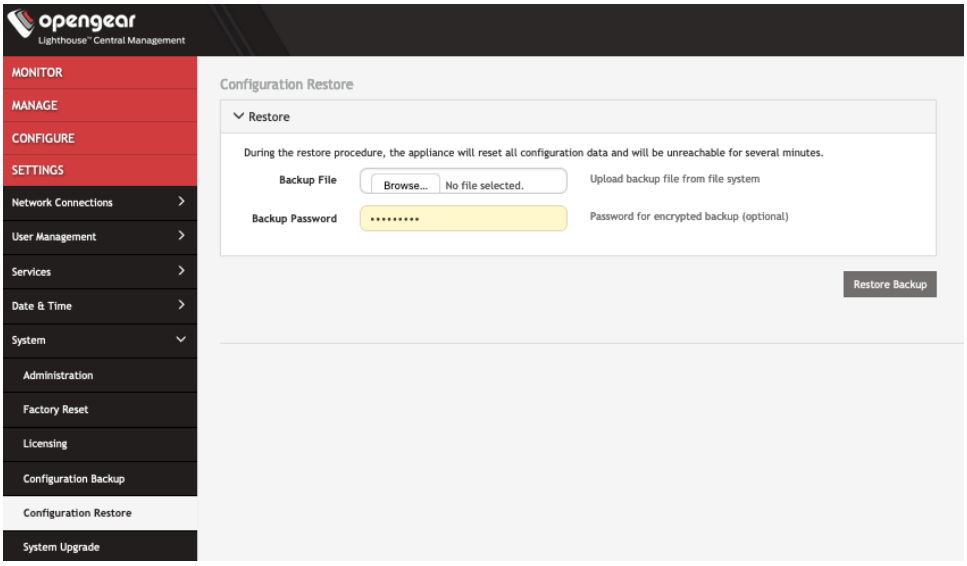


2. If desired, check **Encrypt backup**. Enter and confirm a password.
3. Under the **Advanced** section, specify the paths to any **User Files** you also wish to include in the backup.
4. Click **Download Backup** and save this file. The filename consists of a timestamp and `lh_bak` extension, for example: `lighthouse-20190710100325.lh_bak`

13.4 Configuration Restore

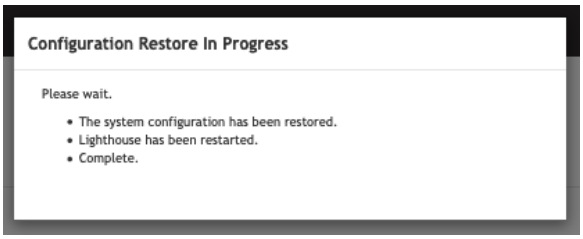
To restore the configuration and user files you backed up using **Configuration Restore**:

1. Select **Settings > System > Configuration Restore**.



2. Locate the file you downloaded when you performed the **Configuration Backup**.

3. If you chose **Encrypt backup** when creating the backup, enter the **Backup Password**.
4. Click **Restore Backup**.



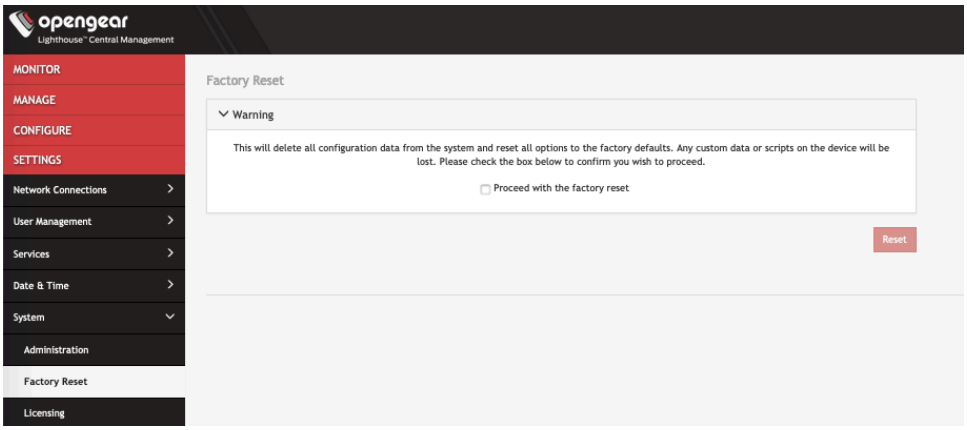
5. Lighthouse will restore the backup and any included user files and restart.

13.5 Returning a Lighthouse instance to factory settings

NOTE: During this process, the current Lighthouse configuration will be overwrittten and user files will be deleted. If you wish, you can create a backup of the configuration and any desired user files.

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 page at opengear.com/support/documentation. It can be accessed by **Help > User Manual** link in the top bar menu.

14. EULA and GPL

The current Opengear end-user license agreement and the GPL can be found at http://opengear.com/eula.