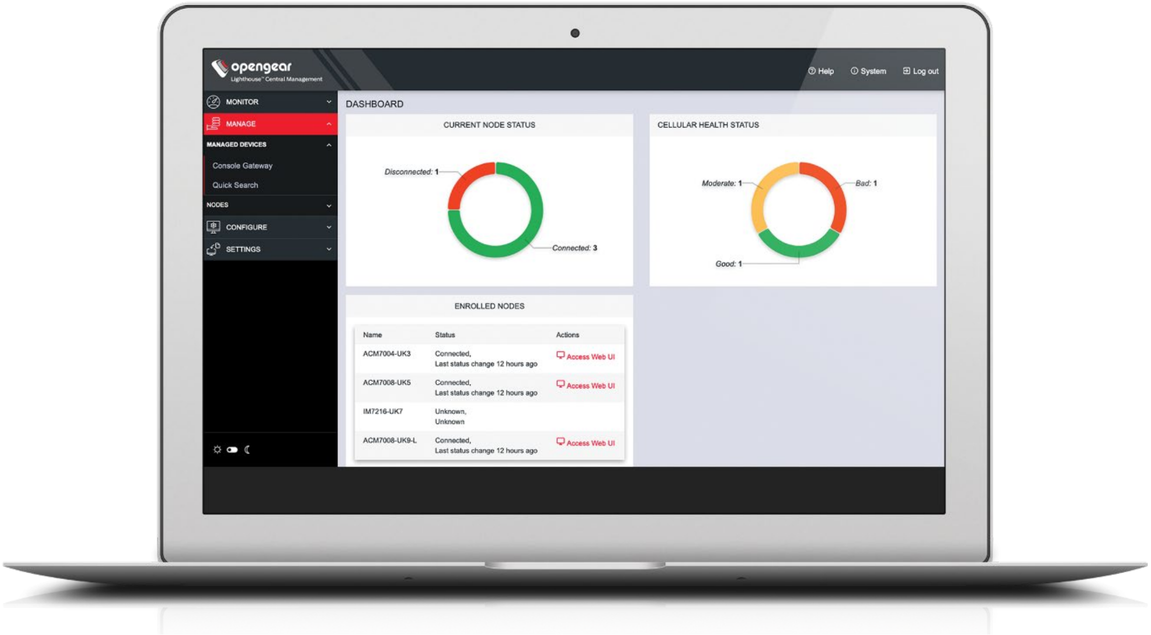
Lighthouse
User Guide

Revision 2021.Q4.0

December 2021



Name	Status	Actions
ACM7004-UK3	Connected, Last status change 12 hours ago	Access Web UI
ACM7008-UK5	Connected, Last status change 12 hours ago	Access Web UI
IM7216-UK7	Unknown, Unknown	
ACM7008-UK9-L	Connected, Last status change 12 hours ago	Access Web UI

TABLE OF CONTENTS

1. About this User Guide

This manual covers Lighthouse and is current as of 2021.Q4.0. When using a minor release, 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: OPERATIONS MANAGER support is partial for this release. Mass node enrollment using ZTP, enrollment via USB drive, and SNMP information are not currently supported for OPERATIONS MANAGER nodes. All template types are supported.

GLOSSARY

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

TERM	DEFINITION
AUTHDOWNLOCAL (RADIUS/LDAP/AAA)	When this authentication option is selected, if remote authentication fails because the user does not exist on the remote AAA server, the user is denied access.
AUTHLOCAL (RADIUS/LDAP/AAA)	When this authentication option is selected, if remote authentication fails because the user does not exist on the remote AAA server, Lighthouse tries to authenticate the user using a local account.
CELLULAR HEALTH	Status of the cellular connection of a node.
DARK MODE	Changes the user interface to display mostly dark colors, reducing the light emitted by device screens.
DOCKER	An open platform for developing, shipping, and running applications. *Docker* enables you to separate your applications from your infrastructure so you can deliver software quickly. Docker powers the NetOps platform within the Lighthouse product.
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.
LIGHT MODE	Changes the user interface to display mostly light colors. This is the default UI setting.

LIGHTHOUSE	System for accessing, managing and monitoring Opengear console servers.
LIGHTHOUSE ENTERPRISE	Offers an elevated centralized management solution with additional functionality. It supports growing trends such as edge computing and SD-WAN with High Availability and Remote IP Access.
LIGHTHOUSE VPN	The OpenVPN based connections that the Lighthouse instance has with the nodes it is managing
LOCALAUTH (RADIUS/LDAP/AAA)	When this authentication option is selected, if local authentication fails, Lighthouse tries to authenticate the user using a remote AAA server.
MANAGED DEVICE	A device that is managed via a node through a serial, USB, or network connection.
MULTIPLE INSTANCE	Access nodes through multiple Lighthouse instances at the same time.
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.
PASSWORD POLICY	Administrative users can define rules for Lighthouse user passwords including length, types of characters, reuse, and expiration period.
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.
REMOTE LOGGING/REMOTE SYSLOG	The ability to send logs to a remote server, for the offsite storage and review of logs.
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

To host Lighthouse, the VM needs to be configured to support a 50GB SCSI disk. As of Lighthouse 20.Q3.0, a second NetOps disk is no longer required and is included as a part of the normal Lighthouse disk. Modules will need to be synchronized with the latest version from Dockerhub, or updated using the offline installer.

- 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, image format. Lighthouse can also be run through cloud hosting services including Amazon's AWS, and Microsoft Azure.
- 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:
    - 50GB SCSI disk. (This can be expanded or reduced after installation and first run)
    - 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 (Up to 500 nodes), allocate:

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

For medium deployments (between 500 and 1000 nodes), allocate:

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

For large deployments (more than 1000), allocate:

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

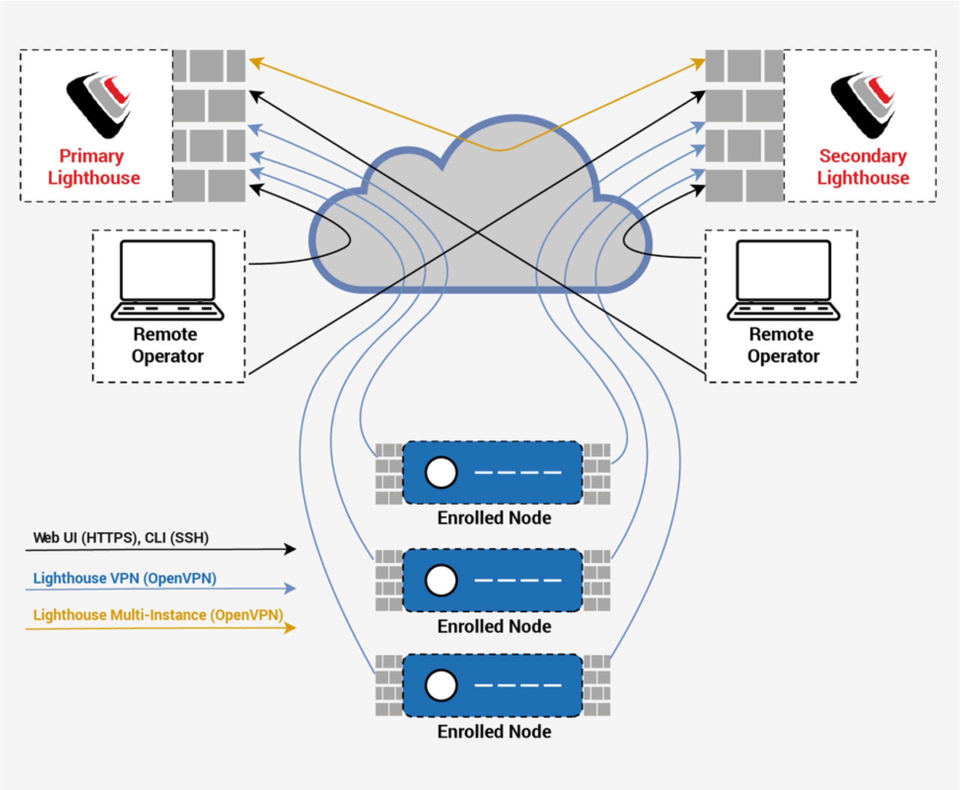
For large deployments, please contact us for guidance on the deployment options, including low and zero-touch enrollment. The performance and limitations are dependent on network deployment.

Additionally, 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.



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:

- Up to ten secondary instances can be enrolled.
- Multiple instance support is available starting with Lighthouse 5.3.
- 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 50GB SCSI disk.

3.1 Lighthouse VM components

Lighthouse VM is available in several formats:

- An Open Volume Format file — **`lighthouse-21.Q4.0-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-21.Q4.0-vmx.zip`** —inside a PKZip archive. This is for use with virtual machine managers from VMware.
- A raw (.hdd) file, **`lighthouse-21.Q4.0-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-21.Q4.0.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-21.Q4.0-hyperv.zip`** — inside a PKZip archive. This is for use in Microsoft Hyper-V deployment.
- A Microsoft Azure file, **`lighthouse-21.Q4.0.azure.zip`** for deploying on Azure.
- An Amazon Web Services bootstrap shell script **`lighthouse-aws-bootstrap.sh`**, and the **`lighthouse-21.Q4.0.aws.raw.tar`** image for deploying on AWS.
- An upgrade file, **`lh_upg.`**

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

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

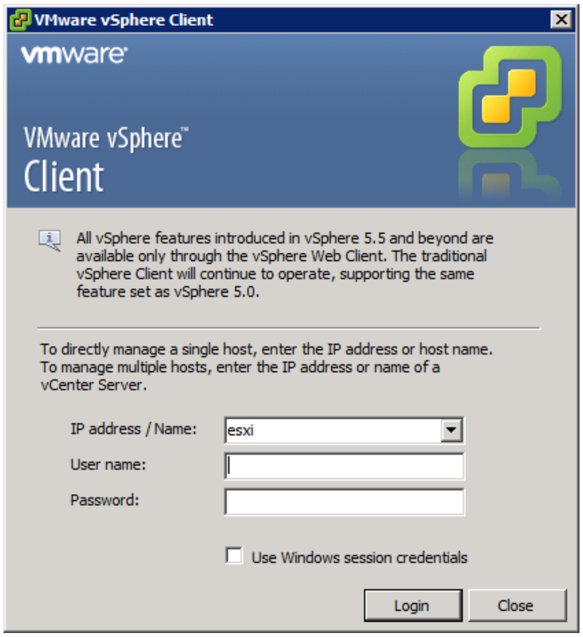
This procedure was tested using the VMware Sphere Client 6.0 running on Windows 7 Enterprise SP 1.

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

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

   Start > All Programs > VMware > VMware vSphere Client

   The VMware vSphere Client opens a login window.



All vSphere features introduced in vSphere 5.5 and beyond are available only through the vSphere Web Client. The traditional vSphere Client will continue to operate, supporting the same feature set as vSphere 5.0.

To directly manage a single host, enter the IP address or host name. To manage multiple hosts, enter the IP address or name of a vCenter Server.

IP address / Name: esxi

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.

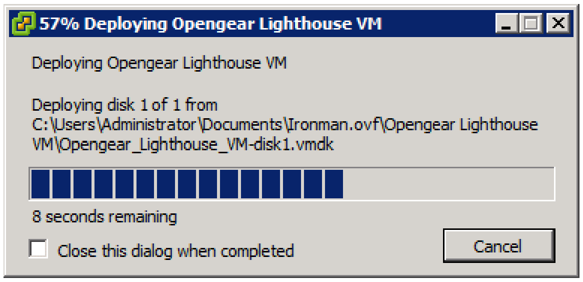
   Navigate to the directory containing the file `Opengear Lighthouse VM.ovf`.

   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.



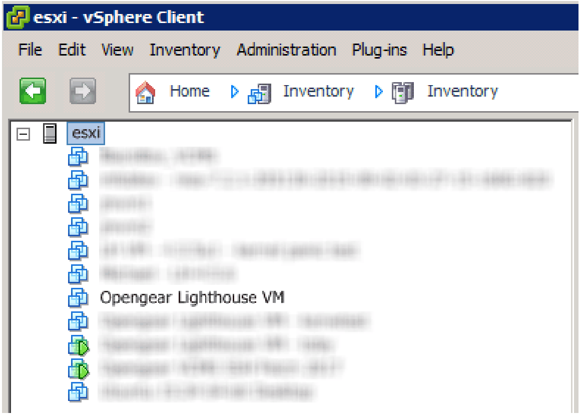
57% Deploying Opengear Lighthouse VM

Deploying Opengear Lighthouse VM

Deploying disk 1 of 1 from C:\Users\Administrator\Documents\Ironman.ovf\Opengear Lighthouse VM\Opengear_Lighthouse_VM-disk1.vmdk

8 seconds remaining

□ Close this dialog when completed



esxi - vSphere Client

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-21.Q4.0-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 `lighthouse` 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\lighthouse\`.

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 `Lighthouse`.

4. If only one file — `Lighthouse` — is visible, double-click it to add the Lighthouse virtual machine to the VMware Workstation 12 Player virtual machines list. If more than one file appears, double-click `Lighthouse.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-21.Q4.0.zip` archive to `C:\Users\%USERNAME%\Virtual Machines\.`
2. Right-click the `lighthouse-21.Q4.0-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 **lighthouse** 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\lighthouse\`.
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 `Lighthouse`.
4. If only one file — `Lighthouse` — appears, double-click it to add the Lighthouse virtual machine to the VMware Workstation Pro virtual machines list. If more than one file appears, double-click `Lighthouse.vmx`.
5. The Lighthouse 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 `lighthouse-hyperv.zip.`
2. Navigate to the extracted folder. Make sure `lighthouse.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 `lighthouse-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 `lighthouse-hyperv.zip`
3. Navigate to the extracted folder. Make sure `lighthouse.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 `lighthouse-hyperv.zip.`
3. Navigate to the extracted folder. Make sure `lighthouse.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-21.Q4.0-ovf.zip` is in `C:\Users\%USERNAME%$\Downloads`.

1. Unzip `lighthouse-ovf.` It may appear as `lighthouse-21.Q4.0-ovf.zip` depending on the Windows Explorer preference settings).
2. Right-click the `lighthouse-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\Lighthouse-ovf`.
4. Uncheck the **Show extracted files when complete** checkbox and edit the destination by removing `Lighthouse-ovf` from the path.
5. Click **Extract**.
6. A folder called `lighthouse-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\Lighthouse.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-21.Q4.0-ovf.zip` is in `~/Downloads`.

1. Unzip **`lighthouse-21.Q4.0-ovf.zip`**.

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

```
Lighthouse-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/Lighthouse.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-21.Q4.0-ovf.zip` is in `~/Downloads`.

1. Unzip `lighthouse-21.Q4.0-ovf.zip`.

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

```
Lighthouse-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/Lighthouse.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-21.Q4.0-ovf.zip` is in `~/Downloads.`

1. Unzip `lighthouse-21.Q4.0-ovf.zip`. This creates a folder — `Lighthouse.ovf` — in `~/Downloads` that contains the following files and folders:

```
Lighthouse.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/Lighthouse.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-21.Q4.0-raw.hdd.tar` is in `~/Downloads`.

1. Expand **`lighthouse-21.Q4.0-raw.hdd.tar`.** This extracts **`lighthouse-21.Q4.0-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-21.Q4.0-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-21.Q4.0-ovf.zip`** is in **`~/Downloads`**.

1. Unzip **`lighthouse-21.Q4.0-ovf.zip`.** This creates a folder — **`Lighthouse.ovf`** — in **`~/Downloads`** that contains the following files and folders:

```
Lighthouse.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/Lighthouse.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-21.Q4.0-ovf.zip`** is in **`~/Downloads`**.

1. Unzip **`lighthouse-21.Q4.0-ovf.zip`**.

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

```
Lighthouse.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/Lighthouse.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 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 `lighthouse.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.**

3.17 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.

To use the AWS environment, you will need to complete several steps. Visit AWS Help for assistance in completing the following steps:

1. Sign Up for AWS
2. Create an IAM User.
3. Create a Key Pair
4. Connect to your instance using your key pair
5. Create a Virtual Private Cloud (VPC)
6. Create a Security Group

Launch a Linux build-box instance

We do not have a public AMI. You will need to launch a Linux build-box instance in order to create a Lighthouse AMI. This is a one-time procedure, as once Lighthouse is running, you can upgrade it from the Lighthouse Web GUI.

We have a compressed binary image and a script. On the first deployment of Lighthouse you do need to spin up a Linux host in EC2 to use as a "build-box", upload the Lighthouse AWS-specific aws.raw.tar image, untar it, then run the Lighthouse aws-bootstrap script, which creates a private AMI. You can then use this to launch one or more Lighthouse instances.

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**.

   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.

8. Click **View Instances** to close the confirmation page and return to the console.

Launch Lighthouse VM Instance

1. Click the acknowledgement check box, and then choose **Launch Instances**.
2. 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.

Upload Lighthouse software to the build-box and run script to create the AMI

After creating the AWS EC2 instance:

1. Locate the `lighthouse-aws-bootstrap.sh` script, provided in the current Lighthouse release folder.
2. Connect via SCP and copy `lighthouse-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 `lighthouse-aws-bootstrap.sh` script with the appropriate parameters to tell it where to find the untarred Lighthouse image on the instance.

   NOTE: *lighthouse-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

NOTE: boot up is headless, so most of the information in section **4. First boot of the Lighthouse VM** does not apply. The root password must be specified in the **Advanced Details**.

4. First boot of the Lighthouse VM

NOTE: This section does not apply to Azure or AWS.

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

   Graphics console boot is preselected and should not be changed. After the first boot has completed a message appears:

   ```
   Welcome to Lighthouse. This is software version:
   2021.Q4.0
   ```

3. The final step in the initial setup appears:

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

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

   ```
   Enter new root password:
   ```

5. 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.

6. The confirm prompt appears:

   ```
   Confirm given password
   ```

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

   ```
   lighthouse login:
   ```

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

```
Password:
```

Enter the newly-set password and press **Enter**. A standard **bash** shell prompt appears with the list of static, DHCP, and IPv6 addresses.

```
net1        192.168.0.1/24
net1:dhcp   192.168.1.186/24
net1        fe80::a00:27ff:fe39:daa3/64
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.

If there is no DHCP and Lighthouse is not reachable on the default address 192.168.0.1 then the static IP address can be changed from the console using the `ogsetnetwork.sh` command.

```
root@lighthouse:~# ogsetnetwork.sh --help

Usage: ogsetnetwork.sh [Use options below to configure a static IP]
        -a, --address           Static IP address to set
        -n, --netmask           Netmask for IP address
        -g, --gateway           Network gateway address
        -d, --dns1              Chosen DNS server #1
        -D, --dns2              Chosen DNS #2
```

Example:

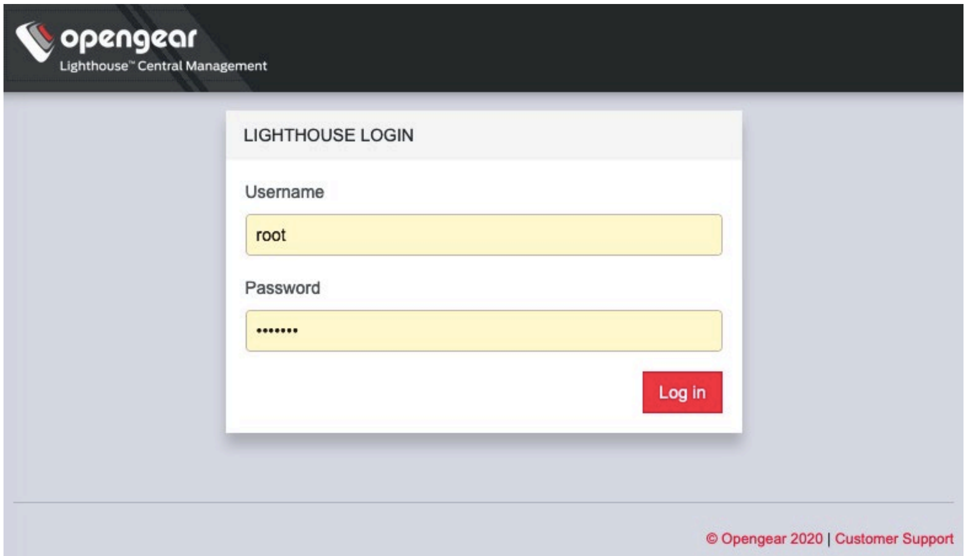
```
ogsetnetwork.sh -a 192.168.1.24 -n 255.255.255.0 -g 192.168.1.1
```

Tip: Type ogset<tab> and tab completion will give you the full command.

5.2 Loading Lighthouse

Open a new browser window or tab and enter:

1. **`https://192.168.0.1/`**, **`https://[DHCP-supplied address]/`**, or an IPv6 address, for example: **`https://[fe80::a00:27ff:fe39:daa3/64]`.**
2. Press **Return**. The Lighthouse Login page loads.


opengear
Lighthouse™ Central Management

LIGHTHOUSE LOGIN

Username

root

Password

•••••••

Log in
© Opengear 2020 | Customer Support

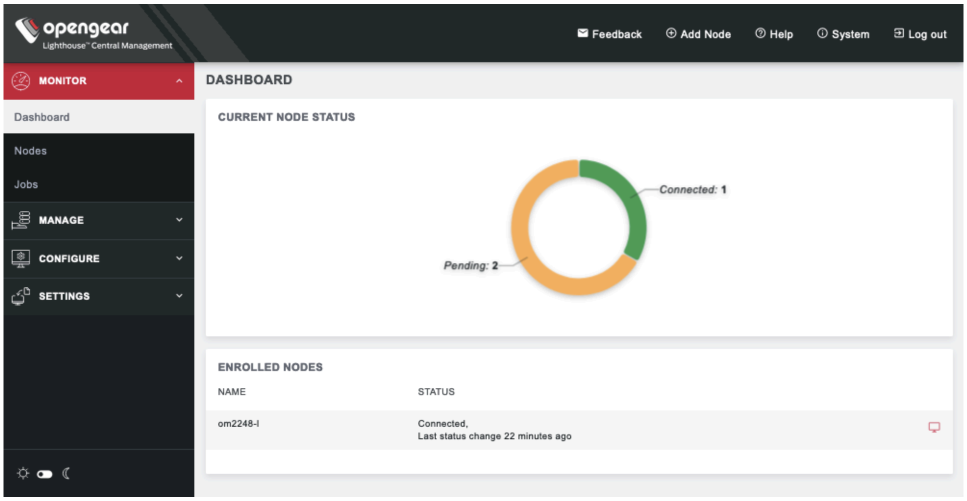
To login to Lighthouse:

1. Enter a **username** in the ***Username*** field. The first time you log in, the **Username** will be *root*.
2. Enter the password in the ***Password*** field.
3. Click **Log In** or press **Enter**. The **Dashboard** loads.

When you log in, you will see a few standard Lighthouse interface elements including:

- The primary menu options **MONITOR**, **MANAGE**, **CONFIGURE**, and **SETTINGS** on the left.
- A light/dark mode toggle switch on the bottom left of the interface. This control allows you to modify the appearance of Lighthouse for low lighting situations.
- System menu options **Add Node**, **Help**, **System**, and **Log Out** on the top right.
- Some column headings have Arrow Icons next to them. Click to toggle between ascending and descending order.

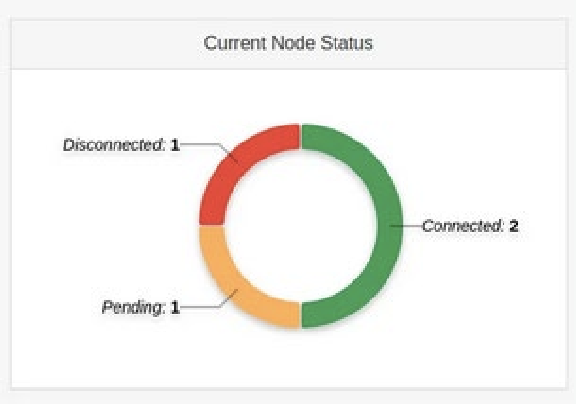
The elements that appear on the **Dashboard** page depend on the privileges granted to the currently logged in user.



NAME	STATUS
om2248-l	Connected, Last status change 22 minutes ago

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**.

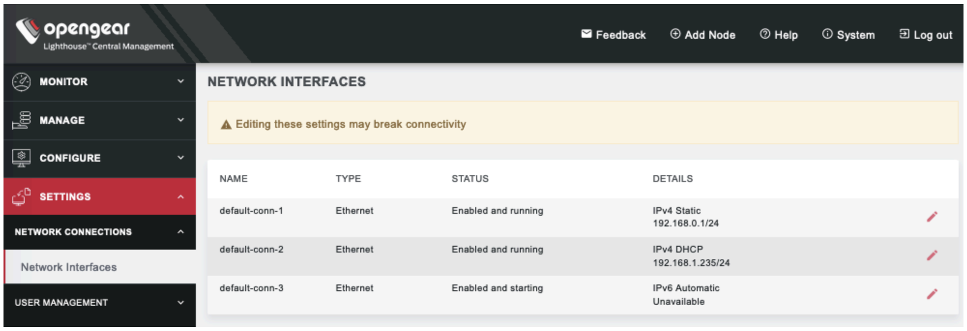


- **Pending** opens **CONFIGURE > NODE ENROLLMENT > Pending Nodes**
- **Connected** opens **CONFIGURE > NODE ENROLLMENT > Enrolled Nodes**
- **Disconnected** opens **MANAGE > NODES > Node Web UI** filtered to show only disconnected nodes

NOTE: The appearance of the Dashboard, the sidebar, and other Lighthouse pages depends on the privileges assigned to the currently 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 with regards to creating and changing settings within Lighthouse.

5.4 Network connections

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



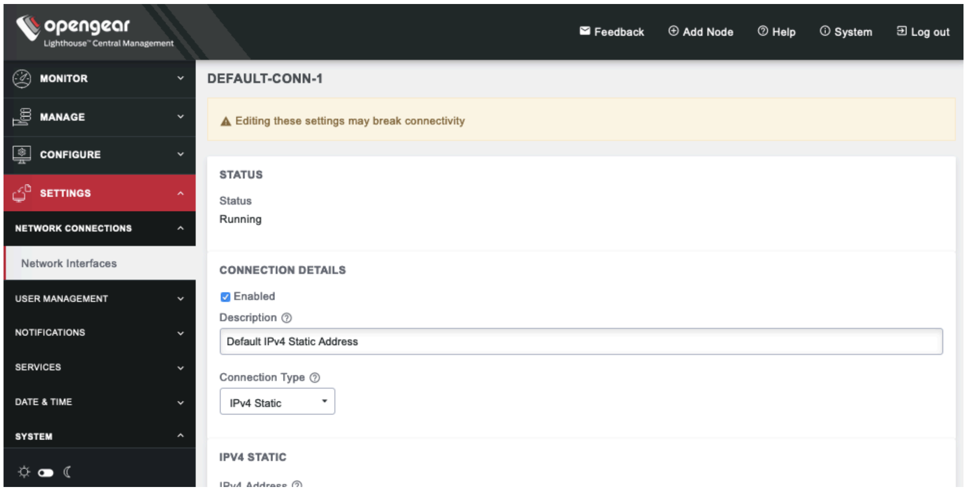
NETWORK INTERFACES

Editing these settings may break connectivity

NAME	TYPE	STATUS	DETAILS
default-conn-1	Ethernet	Enabled and running	IPv4 Static 192.168.0.1/24
default-conn-2	Ethernet	Enabled and running	IPv4 DHCP 192.168.1.235/24
default-conn-3	Ethernet	Enabled and starting	IPv6 Automatic Unavailable

To edit a given network interface:

1. Select **SETTINGS > NETWORK CONNECTIONS > Network Interfaces**
2. Click the ***Edit*** icon to the right of the network interface to be modified.
3. Make the desired changes.
4. Click **Apply**.



DEFAULT-CONN-1

Editing these settings may break connectivity

STATUS

Status
Running

CONNECTION DETAILS

☑ Enabled

Description: Default IPv4 Static Address

Connection Type: IPv4 Static

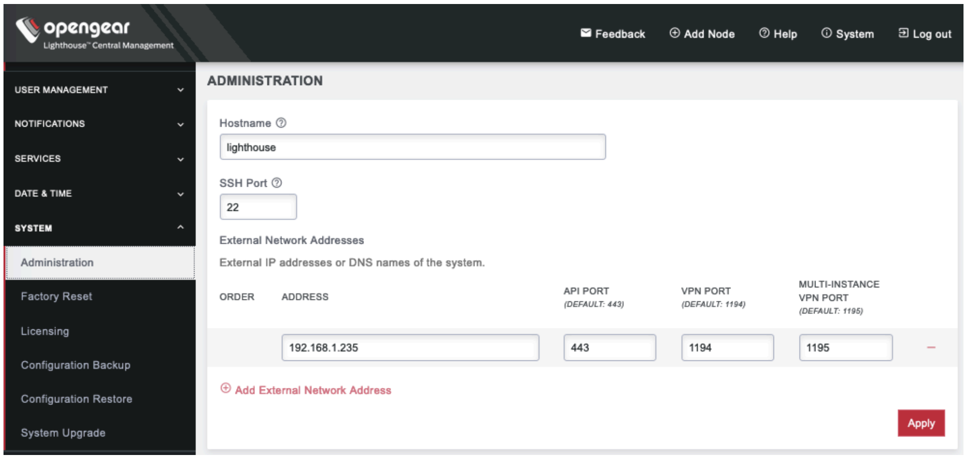
IPV4 STATIC

IPv4 Address

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 desired.



ADMINISTRATION

Hostname: lighthouse

SSH Port: 22

External Network Addresses

External IP addresses or DNS names of the system.

ORDER	ADDRESS	API PORT (DEFAULT: 443)	VPN PORT (DEFAULT: 1194)	MULTI-INSTANCE VPN PORT (DEFAULT: 1195)
	192.168.1.235	443	1194	1195

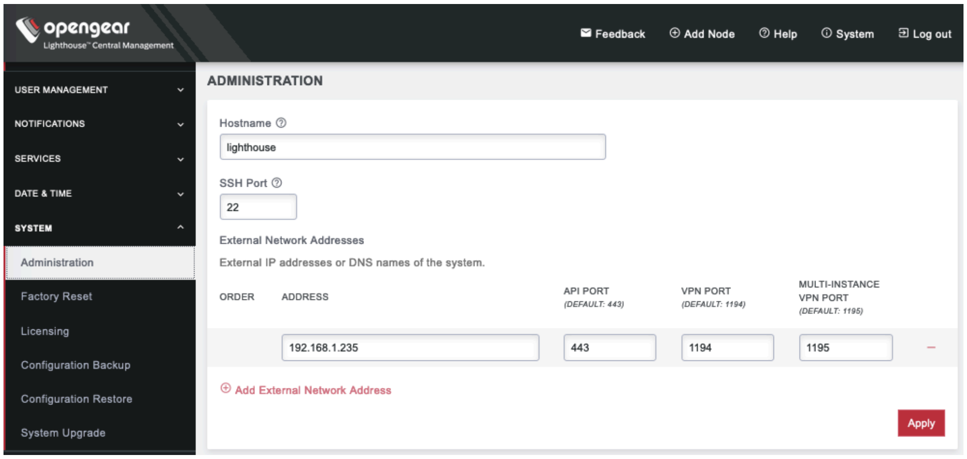
Apply.

5.6 Adding external IP addresses manually

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**.



ADMINISTRATION

Hostname: lighthouse

SSH Port: 22

External Network Addresses

External IP addresses or DNS names of the system.

ORDER	ADDRESS	API PORT (DEFAULT: 443)	VPN PORT (DEFAULT: 1194)	MULTI-INSTANCE VPN PORT (DEFAULT: 1195)
	192.168.1.235	443	1194	1195

3. Change the **API Port**, **VPN Port,** or **Multi-Instance VPN Port** if the ports used on the entered IP address are different from the default settings.
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. Change the **API Port**, **VPN Port,** or **Multi-Instance VPN Port** if the ports used on the entered IP address are different from the default settings.
4. 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.
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.

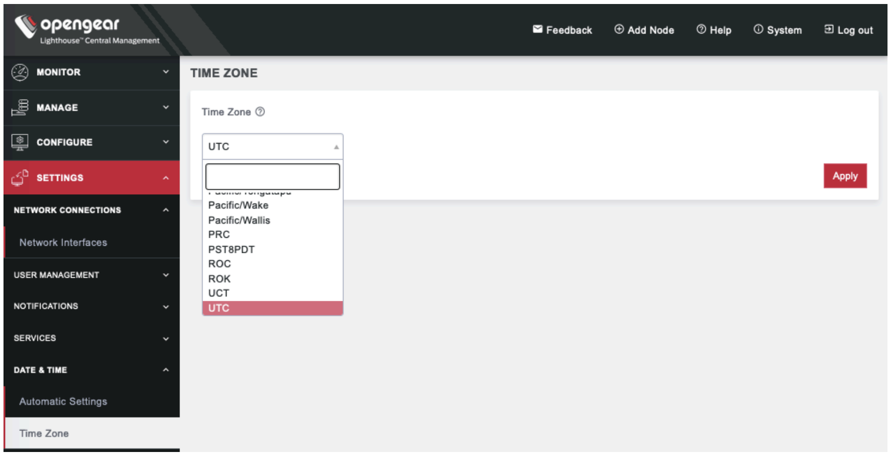
The external IP addresses are sent to a remote node during enrollment in the following order:

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:

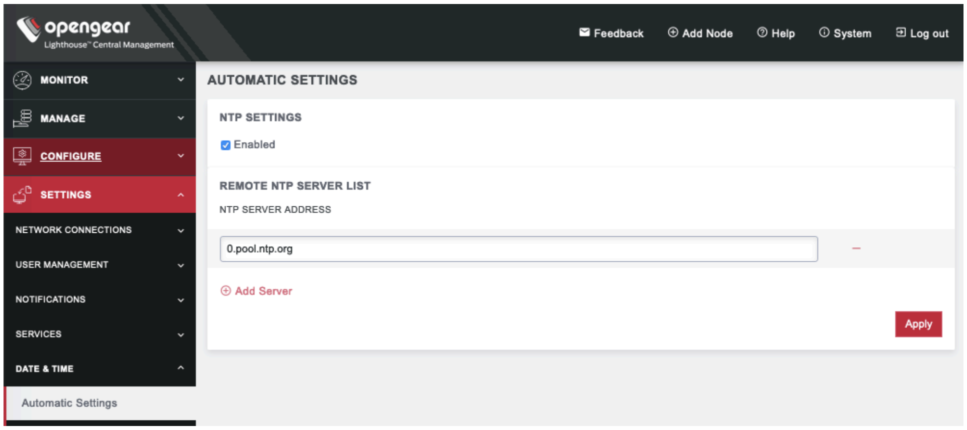
1. Select **SETTINGS > DATE & TIME > Time Zone**.
2. Select the *Lighthouse* instance's time-zone from the ***Time Zone*** drop-down list.
3. Click **Apply**.



TIME ZONE

Time Zone: UTC

1. Select **SETTINGS > DATE & TIME > Automatic Settings**.
2. Click the ***Enabled*** checkbox.
3. Click on **+ Add Server**.
4. Enter a working NTP Server address in the ***NTP Server Address*** field.
5. Click **Apply**.



AUTOMATIC SETTINGS

NTP SETTINGS

☑ Enabled

REMOTE NTP SERVER LIST

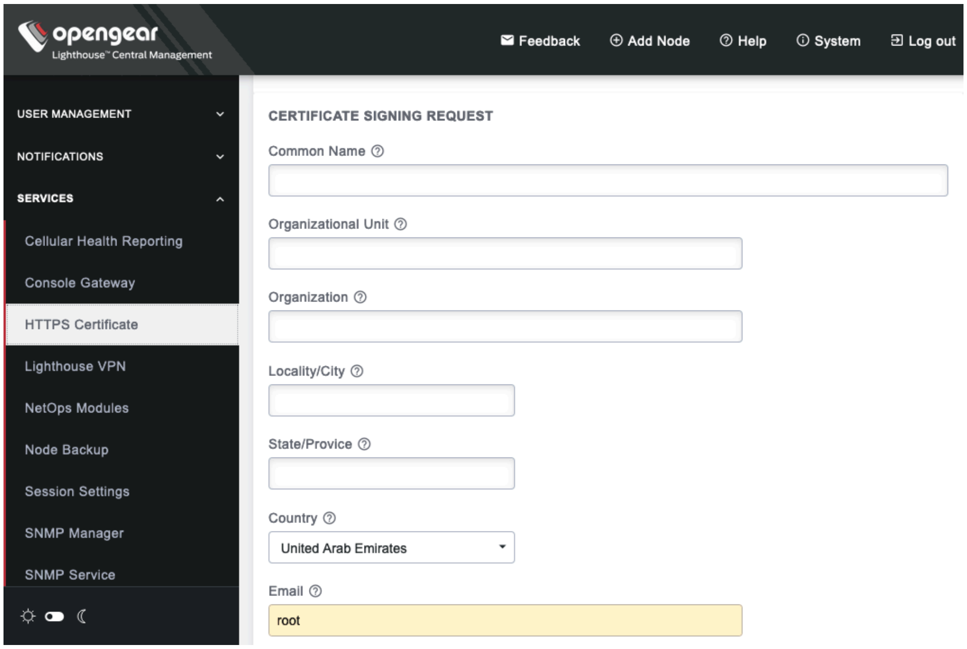
NTP SERVER ADDRESS: 0.pool.ntp.org

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

NOTE: If you plan to use the Lighthouse multiple instance feature, the certificate will be used on all instances. In this case, we recommend using a wildcard certificate.

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.



CERTIFICATE SIGNING REQUEST

Common Name

Organizational Unit

Organization

Locality/City

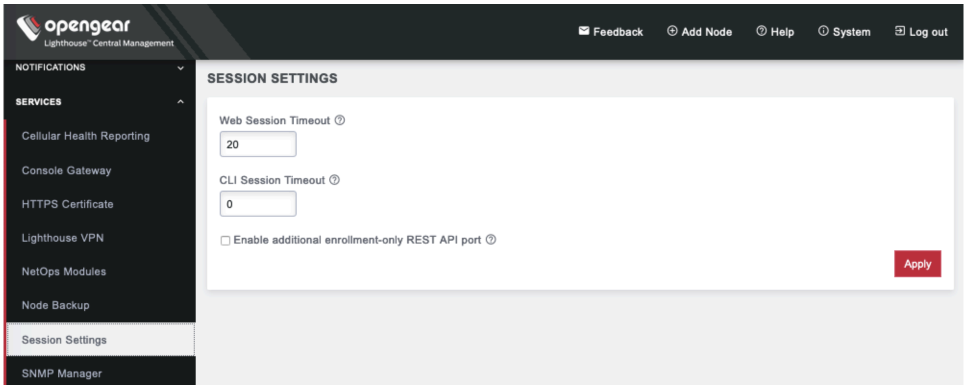
State/Provice

Country: United Arab Emirates

Email: root

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.



SESSION SETTINGS

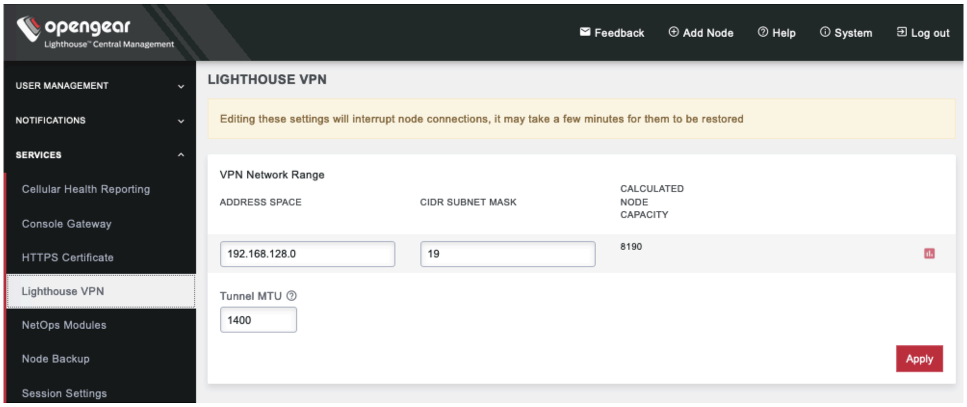
Web Session Timeout: 20

CLI Session Timeout: 0

☐ Enable additional enrollment-only REST API port

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 modify the MTU of the Lighthouse VPN tunnel select **SETTINGS > SERVICES > Lighthouse VPN**. Allowed values are between 1280 and 1500.



LIGHTHOUSE VPN

Editing these settings will interrupt node connections, it may take a few minutes for them to be restored

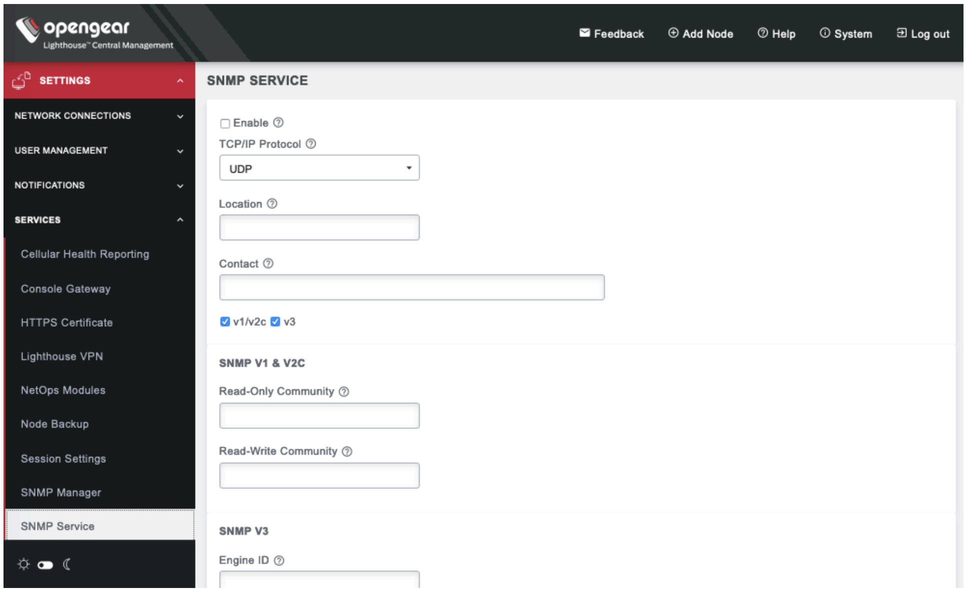
VPN Network Range

ADDRESS SPACE	CIDR SUBNET MASK	CALCULATED NODE CAPACITY
192.168.128.0	19	8190

Tunnel MTU: 1400

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 **Engine ID**, the auto-generated ID coming out of snmpd is displayed. Only standard enterprise MIBs can be used. Lighthouse Health statistics (load/uptime/memory usage, etc.) can be retrieved.



SNMP SERVICE

☐ Enable

TCP/IP Protocol: UDP

Location:

Contact:

☑ v1/v2c ☑ v3

SNMP V1 & V2C

Read-Only Community:

Read-Write Community:

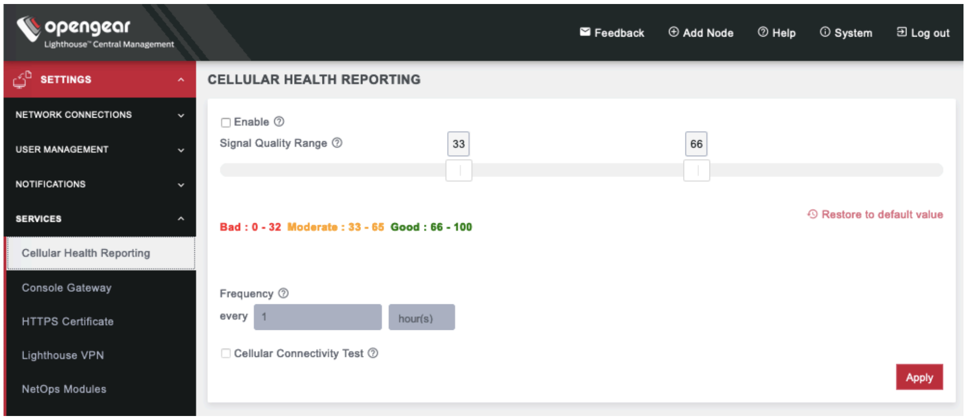
SNMP V3

Engine ID:

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**.



CELLULAR HEALTH REPORTING

□ Enable

Signal Quality Range 33 66

Restore to default value

Bad : 0 - 32 Moderate : 33 - 65 Good : 66 - 100

Frequency

every 1 hour(s)

□ Cellular Connectivity Test

When cell health checks are enabled, the network carrier, IMEI, IMSI and ICCID of the downstream SIM being utilized are part of the information that is displayed in Lighthouse for managed nodes

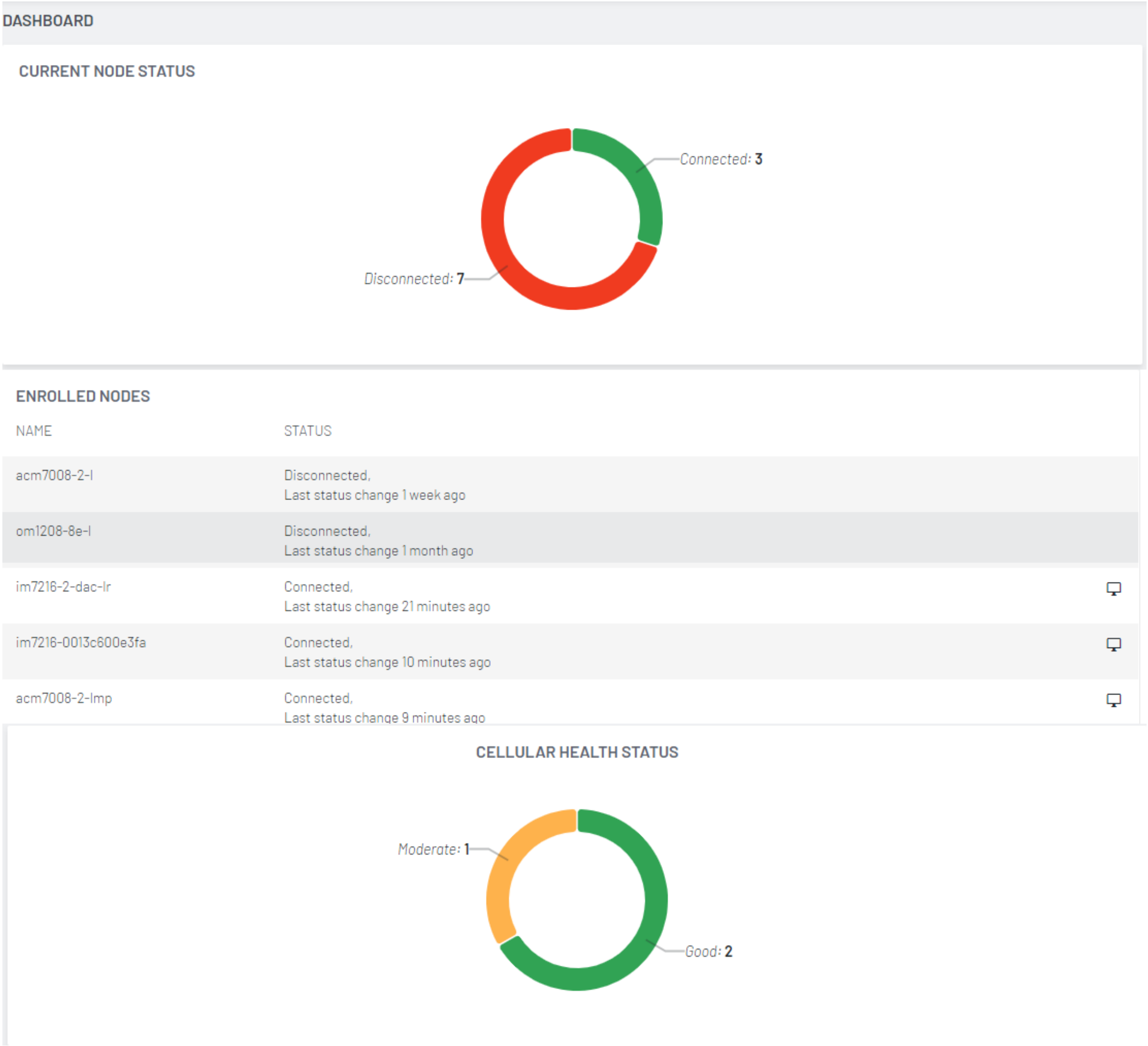
If a managed node has the modem disabled/off, an appropriate status is shown in Lighthouse for the node.

- Check the **Enable** box to enable Cellular Health monitoring.
- Adjust the sliders on this screen to control 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**.
- Adjust how frequently Lighthouse will check the signal quality.
- Finally, you can run a periodic **Cellular Connectivity Test** which will make sure the cellular can actually connect. This will use cellular data.

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

5.12.1 Cellular Health Dashboard

The current health status of enrolled nodes can be viewed from the Cellular Health Dashboard: **MONITOR > Dashboard**. Click on a segment of interest to open the Node Web UI page which displays Node health information:



DASHBOARD

CURRENT NODE STATUS

ENROLLED NODES

NAME	STATUS	
acm7008-2-l	Disconnected, Last status change 1 week ago	
om1208-8e-l	Disconnected, Last status change 1 month ago	
im7216-2-dac-lr	Connected, Last status change 21 minutes ago	🖥
im7216-0013c600e3fa	Connected, Last status change 10 minutes ago	🖥
acm7008-2-lmp	Connected, Last status change 9 minutes ago	🖥

CELLULAR HEALTH STATUS

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:

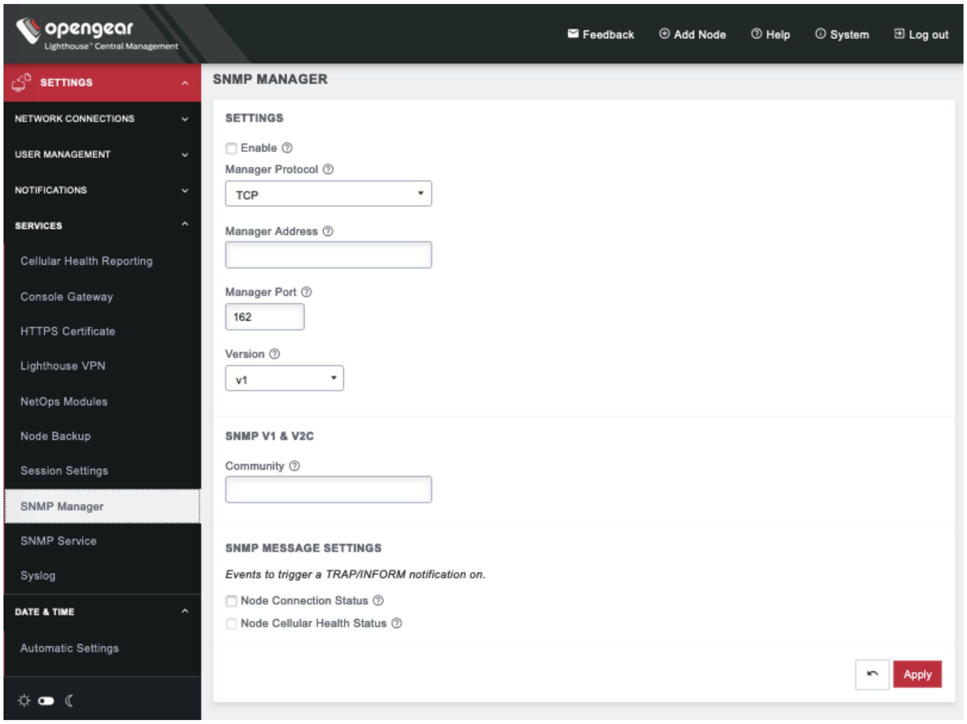
```
snmpwalk -m ALL -v1 -c public 192.168.1.1 ogLhNodeSerialNumber.192.168.128.2
snmpget -m ALL -v1 -c public 192.168.1.1 ogLhNodeSerialNumber.192.168.128.2
```

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.



SNMP MANAGER

SETTINGS

□ Enable

Manager Protocol: TCP

Manager Address:

Manager Port: 162

Version: v1

SNMP V1 & V2C

Community:

SNMP MESSAGE SETTINGS

Events to trigger a TRAP/INFORM notification on.

□ Node Connection Status

□ Node Cellular Health Status

1. Under the **Settings** section, select the **Enable** checkbox.
2. Choose **UDP, UDP over IPv6, TCP,** or **TCP over IPv6** in the **Manager Protocol** drop-down.
3. Enter the **Manager Address** to receive SNMP messages.
4. Enter the **Manager Port**.
5. Check the SNMP protocol **Version** from the **v1, v2c, v3** drop-down.

Depending on the selected SNMP version, complete the following steps.

For **v1**, enter the SNMP **Community** to use for messages.

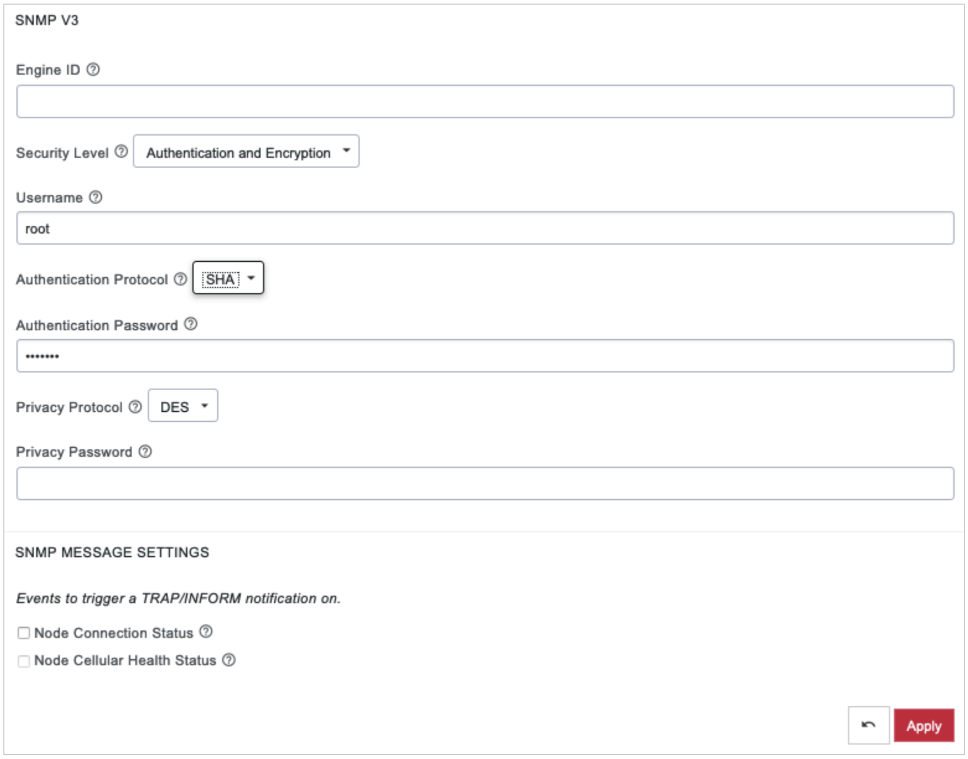
For **v2c**:

1. Choose **TRAP** or **INFORM** as the **SNMP Message Type**.
2. Enter the SNMP **Community** to use for messages.

For **v3**:

1. Choose **TRAP** or **INFORM** as the **SNMP Message Type**.

2. 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 Engine ID is not needed for an SNMP INFORM message.
3. Enter the SNMP v3 **Engine ID** and desired **Security Level**.
4. Enter the **Username** to send the messages as, select the **Authentication Protocol**, either **MD5** or **SHA**, and enter the SNMP user's **Authentication Password**.
5. Choose a **Privacy Protocol,** either **DES** or **AES,** and enter a **Privacy Password**.



SNMP V3

Engine ID ⓘ

Security Level ⓘ Authentication and Encryption

Username ⓘ root

Authentication Protocol ⓘ SHA

Authentication Password ⓘ •••••••

Privacy Protocol ⓘ DES

Privacy Password ⓘ

SNMP MESSAGE SETTINGS

Events to trigger a TRAP/INFORM notification on.

□ Node Connection Status ⓘ
□ Node Cellular Health Status ⓘ

For all three SNMP versions, trigger TRAP/INFORM notifications by checking either or both of the **Node Connection Status** and the **Node Cellular Heath Status** checkboxes. Click **Apply**.

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:

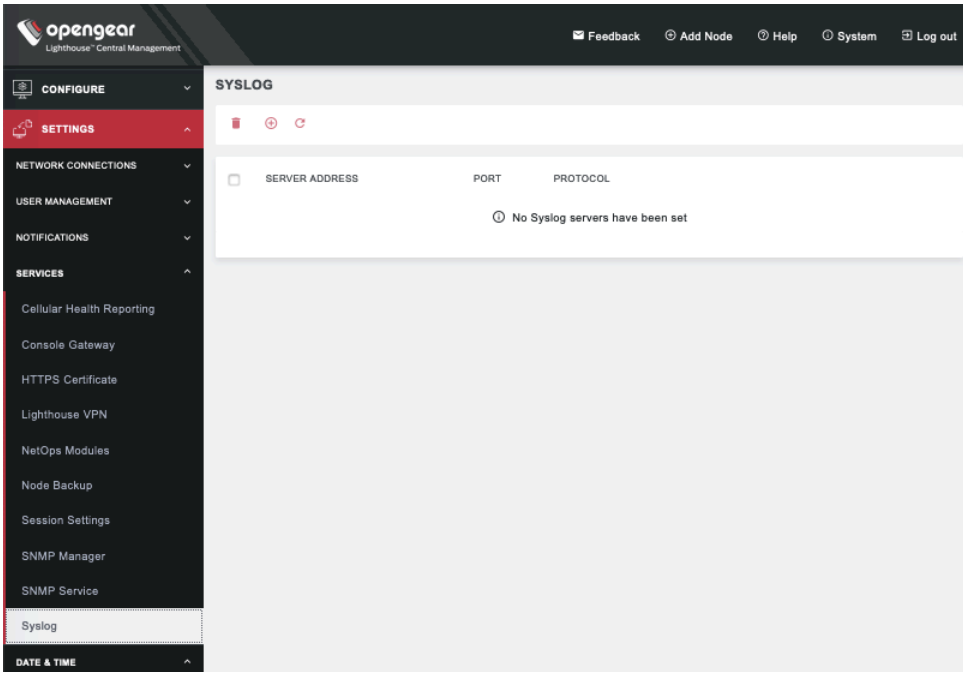
```
nodeStatusNotif
        ogLhNodeName
        ogLhNodeIndex
        ogLhNodeConnStatus
```

Structure of notifications for third-party nodes:

thirdPartyNodeStatusNotif
- ogLhThirdPartyNodeIndex
- ogLhThirdPartyNodeName
- ogLhThirdPartyNodeAddress
- ogLhThirdPartyNodeConnStatus

5.15 Syslog export

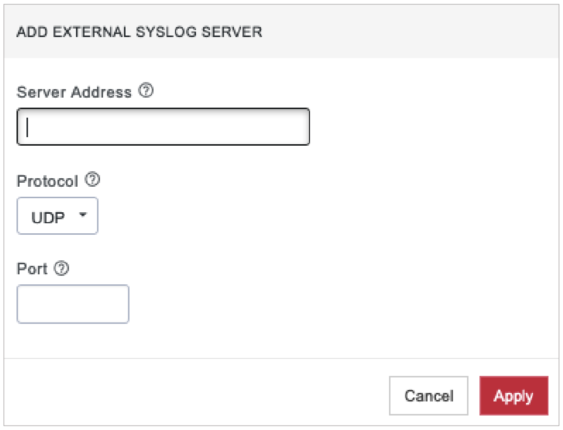
Administrative users can specify multiple external servers to export the syslog to via TCP or UDP. Select **SETTINGS > SERVICES > Syslog**.



No Syslog servers have been set

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

1. Click the **+** symbol. The **Add External Syslog Server** dialog opens.



Protocol: UDP

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 ✎. Delete a server by clicking the **x** button ⊗.

Edit the user "fred" and make him a member of the admin group

```
2020-05-22T16:22:40.646116+01:00 LH20Q4-pre rest_api_log[2513]: GET 200 (root |
192.168.1.230) - /api/v3.5/groups?page=1&per_page=99
2020-05-22T16:22:46.403616+01:00 localhost rest_api_log[62]: PUT 200 (root |
192.168.1.230) - /api/v3.5/users/users-2 REQUEST={'user': {'username': 'fred',
'password': '********', 'description': 'fred', 'enabled': True, 'groups': ['groups-
2'], 'rights': {'delete': True, 'modify': True}, 'no_password': False, 'expired':
False, 'locked_out': False}} RESPONSE={'user': {'username': 'fred', 'description':
'fred', 'enabled': True, 'id': 'users-2', 'no_password': False, 'expired': False,
'locked_out': False, 'rights': {'delete': True, 'modify': True}, 'groups': ['groups-
2']}}
2020-05-22T16:22:46.490627+01:00 localhost rest_api_log[62]: GET 200 (root |
192.168.1.230) - /api/v3.5/users?page=1&per_page=10 RESPONSE={'users': [{'username':
'root', 'description': 'System wide SuperUser account', 'enabled': True, 'id': 'users-
1', 'no_password': False, 'expired': False, 'locked_out': False, 'rights': {'delete':
True, 'modify': True}, 'groups': ['groups-2']}, {'username': 'fred', 'description':
'fred', 'enabled': True, 'id': 'users-2', 'no_password': False, 'expired': False,
'locked_out': False, 'rights': {'delete': True, 'modify': True}, 'groups': ['groups-
2']}], 'meta': {'total_pages': 1}}
```

Disable the user "fred"

```
2020-05-22T16:24:11.626425+01:00 localhost rest_api_log[62]: PUT 200 (root |
192.168.1.230) - /api/v3.5/users/users-2 REQUEST={'user': {'username': 'fred',
'password': '********', 'description': 'fred', 'enabled': False, 'groups': ['groups-
2'], 'rights': {'delete': True, 'modify': True}, 'no_password': False, 'expired':
False, 'locked_out': False}} RESPONSE={'user': {'username': 'fred', 'description':
'fred', 'enabled': False, 'id': 'users-2', 'no_password': False, 'expired': False,
'locked_out': False, 'rights': {'delete': True, 'modify': True}, 'groups': ['groups-
2']}}
```

Enable the user “fred”

```
2020-05-22T16:25:08.013266+01:00 localhost rest_api_log[62]: PUT 200 (root |
192.168.1.230) - /api/v3.5/users/users-2 REQUEST={'user': {'username': 'fred',
'password': '********', 'description': 'fred', 'enabled': True, 'groups': ['groups-
2'], 'rights': {'delete': True, 'modify': True}, 'no_password': False, 'expired':
False, 'locked_out': False}} RESPONSE={'user': {'username': 'fred', 'description':
'fred', 'enabled': True, 'id': 'users-2', 'no_password': False, 'expired': False,
'locked_out': False, 'rights': {'delete': True, 'modify': True}, 'groups': ['groups-
2']}}
2020-05-22T16:25:08.236626+01:00 LH20Q4-pre configurator_users[21644]: User <fred>
added to passwords file
```

Change the user “fred's” password

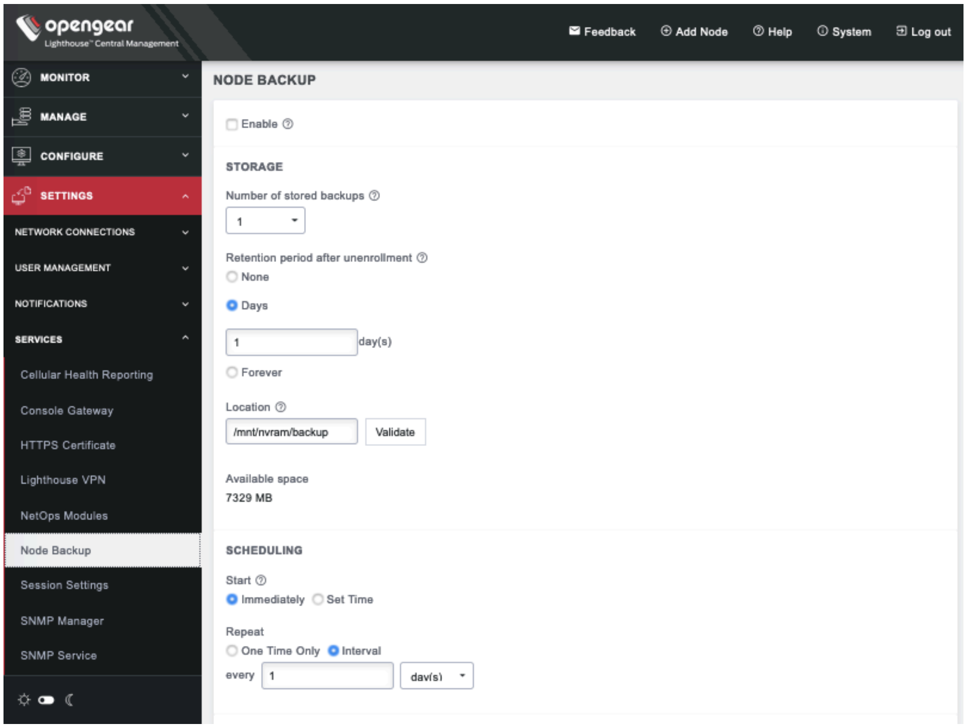
```
2020-05-22T16:29:41.698685+01:00 localhost rest_api_log[62]: PUT 200 (root |
192.168.1.230) - /api/v3.5/users/users-2 REQUEST={'user': {'username': 'fred',
'password': '********', 'description': 'fred', 'enabled': True, 'groups': ['groups-
2'], 'rights': {'delete': True, 'modify': True}, 'no_password': False, 'expired':
False, 'locked_out': False}} RESPONSE={'user': {'username': 'fred', 'description':
'fred', 'enabled': True, 'id': 'users-2', 'no_password': False, 'expired': False,
'locked_out': False, 'rights': {'delete': True, 'modify': True}, 'groups': ['groups-
2']}}
2020-05-22T16:29:41.866376+01:00 localhost rest_api_log[62]: GET 200 (root |
192.168.1.230) - /api/v3.5/users?page=1&per_page=10 RESPONSE={'users': [{'username':
'root', 'description': 'System wide SuperUser account', 'enabled': True, 'id': 'users-
1', 'no_password': False, 'expired': False, 'locked_out': False, 'rights': {'delete':
True, 'modify': True}, 'groups': ['groups-2']}, {'username': 'fred', 'description':
'fred', 'enabled': True, 'id': 'users-2', 'no_password': False, 'expired': False,
'locked_out': False, 'rights': {'delete': True, 'modify': True}, 'groups': ['groups-
2']}], 'meta': {'total_pages': 1}}
```

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**.



NODE BACKUP

☐ Enable

STORAGE

Number of stored backups: 1

Retention period after unenrollment
○ None
● Days
1 day(s)
○ Forever

Location: /mnt/nvram/backup Validate

Available space
7329 MB

SCHEDULING

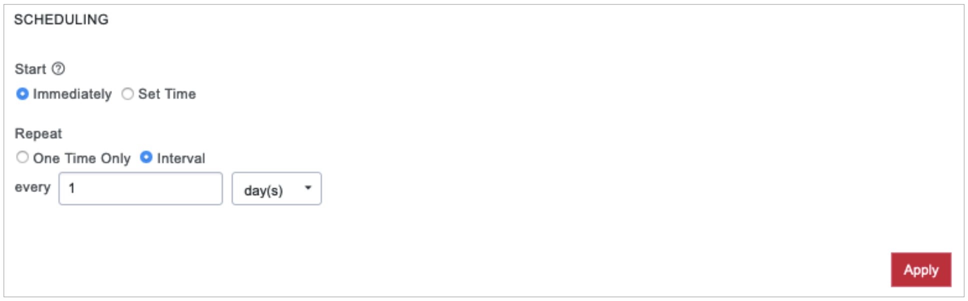
Start
● Immediately ○ Set Time

Repeat
○ One Time Only ● Interval
every 1 day(s)

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. We suggest you store backups at `/mnt/nvram/`.
6. Click **Validate** to make sure the location exists and has enough space to store them.
7. Click **Apply** or set a schedule.

To set an automated schedule for performing node backups:

1. Scroll down to the **Scheduling** section.



SCHEDULING

Start
● Immediately ○ Set Time

Repeat
○ One Time Only ● Interval
every 1 day(s)

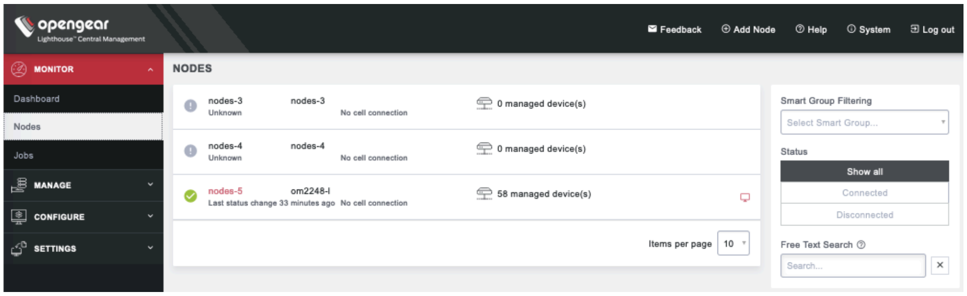
Apply

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.

5.17 Monitor Nodes

Lighthouse allows you to view all nodes including connected, not connected, enrolled, and pending, as well as nodes using cellular. Go to **MONITOR > Nodes**.



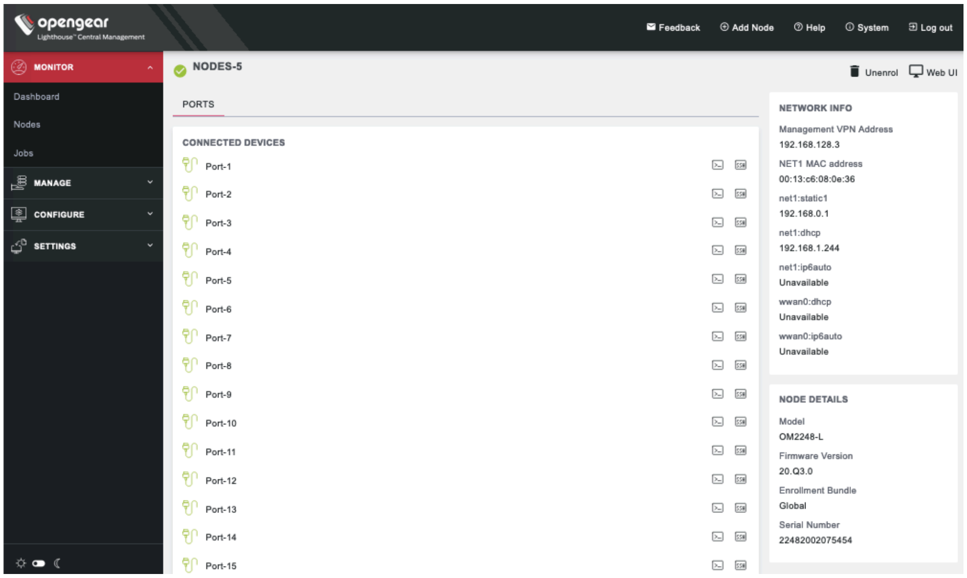
NODES

nodes-3 Unknown	nodes-3	No cell connection	0 managed device(s)
nodes-4 Unknown	nodes-4	No cell connection	0 managed device(s)
nodes-5 Last status change 33 minutes ago	om2248-l	No cell connection	58 managed device(s)

This page displays the cellular health and number of ports. It also offers a button on the right of connected nodes to access the Web UI.

To the right of the table, you can filter by Smart Group, connection status, and perform a text search. By default, the **Show all** filter is selected.

You can click on any connected node to view the node details, list of connected devices, and a list of unconfigured ports. From this detail page, you can access the web terminal and SSH of each connected device. You can also unenroll a connected node or visit the Web UI using the buttons on the top right of the page.



NODES-5

CONNECTED DEVICES

Port-1
Port-2
Port-3
Port-4
Port-5
Port-6
Port-7
Port-8
Port-9
Port-10
Port-11
Port-12
Port-13
Port-14
Port-15

NETWORK INFO

Management VPN Address
192.168.128.3

NET1 MAC address
00:13:c6:08:0e:36

net1:static1
192.168.0.1

net1:dhcp
192.168.1.244

net1:ip6auto
Unavailable

wwan0:dhcp
Unavailable

wwan0:ip6auto
Unavailable

NODE DETAILS

Model
OM2248-L

Firmware Version
20.Q3.0

Enrollment Bundle
Global

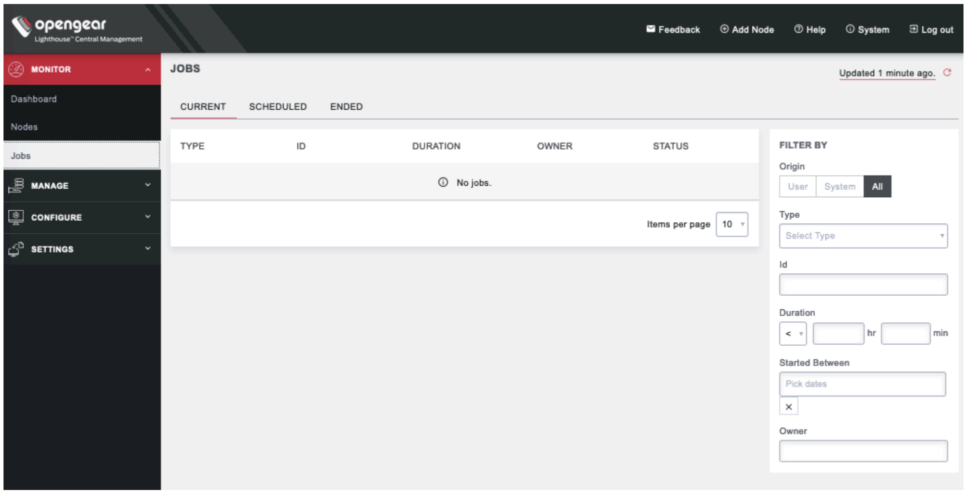
Serial Number
22482002075454

5.18 Monitor Jobs

You can keep track of jobs in Lighthouse which are scheduled, running, or have run.

NOTE: For the 21.Q3.0 release (onwards until notified otherwise), this can only monitor system-created jobs. Future releases will add increased functionality.

Go to **MONITOR > Jobs**.



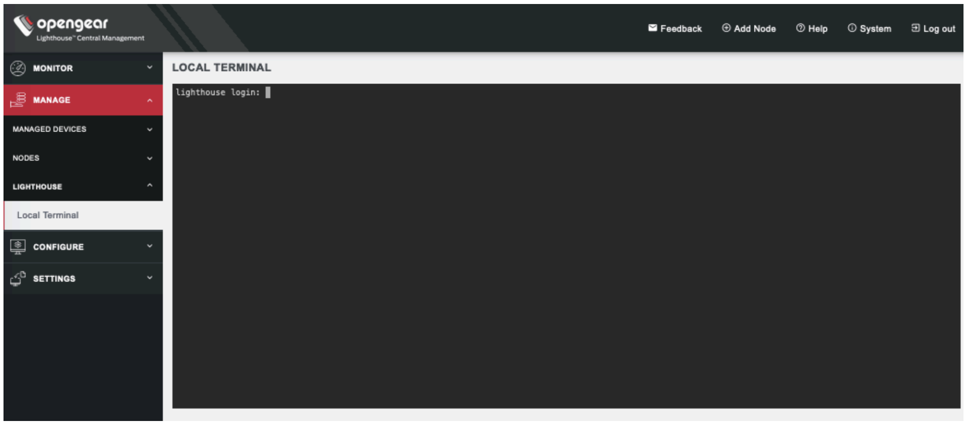
At the top of the page, you can click on **CURRENT**, **SCHEDULED**, or **ENDED** links to see those jobs. You can also use the **FILTER BY** form on the right to further refine your results by type, ID, job duration, dates the job began, and owner.

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**



```
lighthouse login: 
```

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:

- Licensing third-party nodes before enrollment
- Enrolling nodes
- The **Enrolled Nodes** page
- Filtering pages displaying nodes
- Creating and editing **Smart Groups**
- Creating and editing **Managed Device Filters**
- Connecting to a node's web management interface
- Connecting to a node's serial ports via **Console Gateway**

7.1 Licensing third-party nodes before enrollment

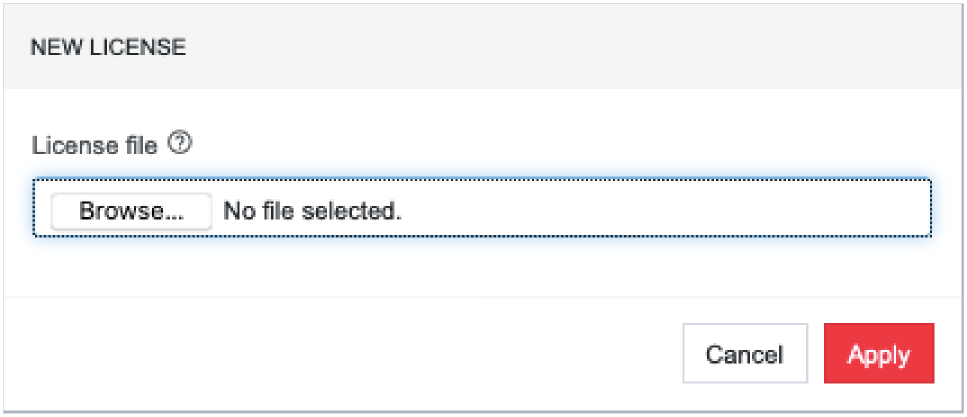
Lighthouse includes support for managing third-party remote nodes at no cost. Support for third-party remote nodes is not built-in to a new Lighthouse instance, however: it is added via a *license*.

A *license* is an encrypted, RFC 7519-compliant, JSON web token that contains key-values. Licenses are distributed by Opengear and are available as `.zip f`iles 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 at the top of the page. A **New License** dialog opens.


NEW LICENSE

License file

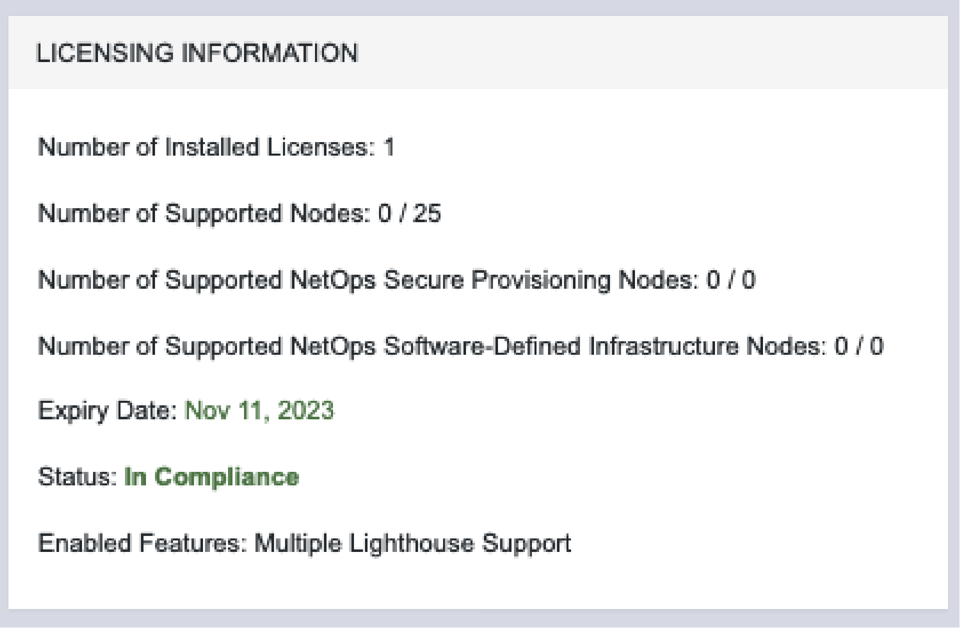
No file selected.

3. Click **Browse…** to 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 under the **LICENSING INFORMATION** section. Select **MONITOR > Dashboard** to open the **Dashboard**.



LICENSING INFORMATION

Number of Installed Licenses: 1

Number of Supported Nodes: 0 / 25

Number of Supported NetOps Secure Provisioning Nodes: 0 / 0

Number of Supported NetOps Software-Defined Infrastructure Nodes: 0 / 0

Expiry Date: Nov 11, 2023

Status: **In Compliance**

Enabled Features: Multiple Lighthouse Support

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.

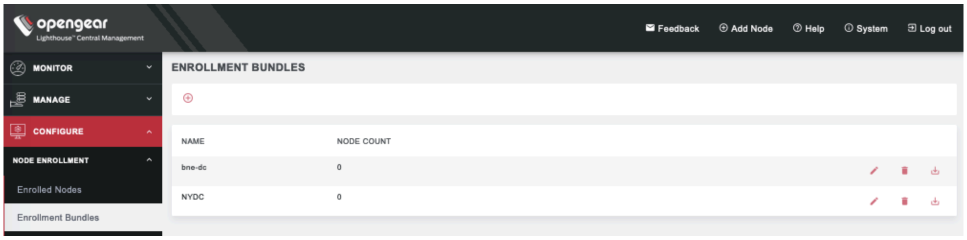
Associating templates with an enrollment bundle allows to run a set of templates on a node, after it has been enrolled. Any template defined on the Lighthouse can be added to an enrollment bundle, and each bundle supports any number of templates.

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:

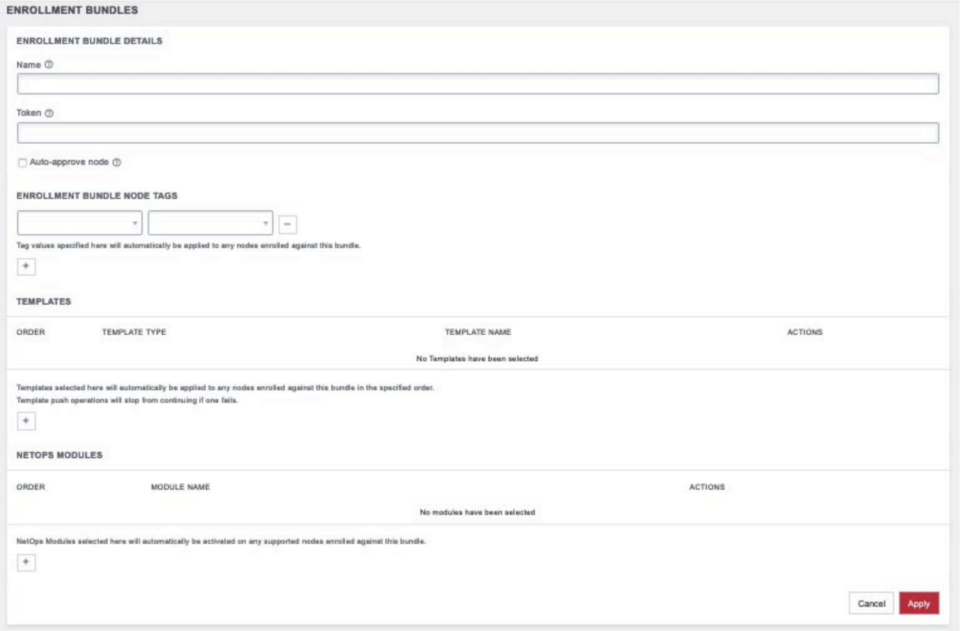
1. Select **CONFIGURE > NODE ENROLLMENT > Enrollment Bundles**



ENROLLMENT BUNDLES

NAME	NODE COUNT
bne-dc	0
NYDC	0

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



ENROLLMENT BUNDLES

ENROLLMENT BUNDLE DETAILS

Name

Token

□ Auto-approve node

ENROLLMENT BUNDLE NODE TAGS

Tag values specified here will automatically be applied to any nodes enrolled against this bundle.

TEMPLATES

ORDER	TEMPLATE TYPE	TEMPLATE NAME	ACTIONS
		No Templates have been selected	

Templates selected here will automatically be applied to any nodes enrolled against this bundle in the specified order.
Template push operations will stop from continuing if one fails.

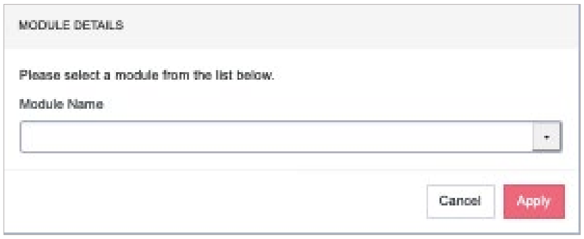
NETOPS MODULES

ORDER	MODULE NAME	ACTIONS
	No modules have been selected	

NetOps Modules selected here will automatically be activated on any supported nodes enrolled against this bundle.

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.



MODULE DETAILS

Please select a module from the list below.

Module Name

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
```

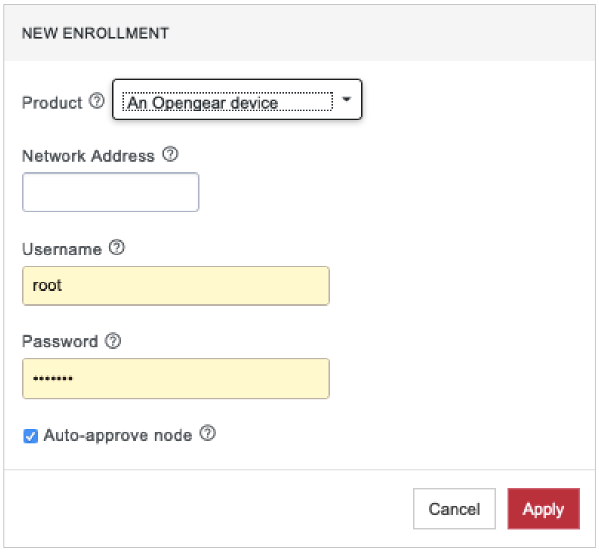
7.2.5 Enrollment via Lighthouse Web UI

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 open a **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
- A Digi Passport



NEW ENROLLMENT

Product: An Opengear device

Network Address:

Username: root

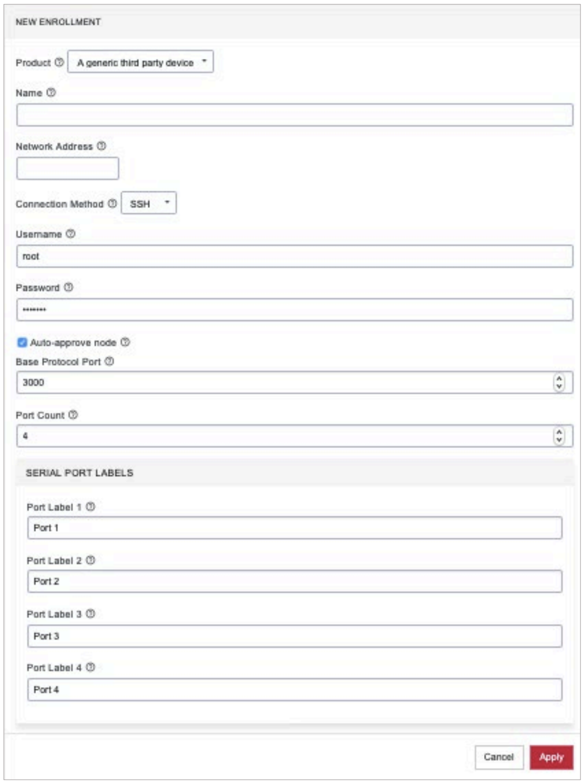
Password: •••••••

☑ Auto-approve node

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***.



NEW ENROLLMENT

Product: A generic third party device

Name:

Network Address:

Connection Method: SSH

Username: root

Password: •••••••

☑ Auto-approve node

Base Protocol Port: 3000

Port Count: 4

SERIAL PORT LABELS

Port Label 1: Port 1

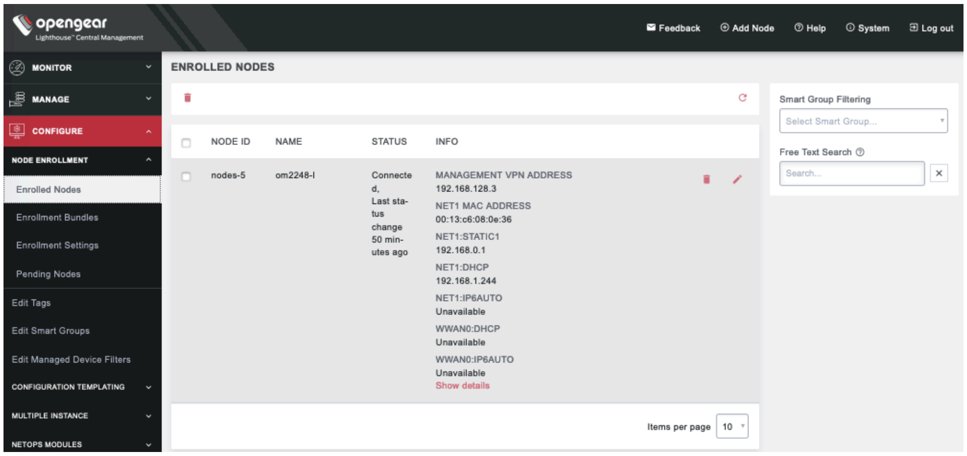
Port Label 2: Port 2

Port Label 3: Port 3

Port Label 4: Port 4

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

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



ENROLLED NODES

NODE ID	NAME	STATUS	INFO
nodes-5	om2248-l	Connected, Last status change 50 minutes ago	MANAGEMENT VPN ADDRESS 192.168.128.3 NET1 MAC ADDRESS 00:13:c6:08:0e:36 NET1:STATIC1 192.168.0.1 NET1:DHCP 192.168.1.244 NET1:IP6AUTO Unavailable WWAN0:DHCP Unavailable WWAN0:IP6AUTO Unavailable

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** of Lighthouse (which can be hostname, FQDN, or IP 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 OM, ACM, CM, and IM Web UI

- **OM:** Nodes can be enrolled into a Lighthouse instance on OPERATIONS MANAGER Web UI using the **CONFIGURE > LIGHTHOUSE Enrollment** menu item and the `lhvpn-callhome` command. See the OPERATIONS MANAGER User Guide for more details.
- **ACM, CM and IM**: On the Web UI, select **Serial & Network > Lighthouse** to open the **Request Enrollment with Lighthouse Server** page.

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:

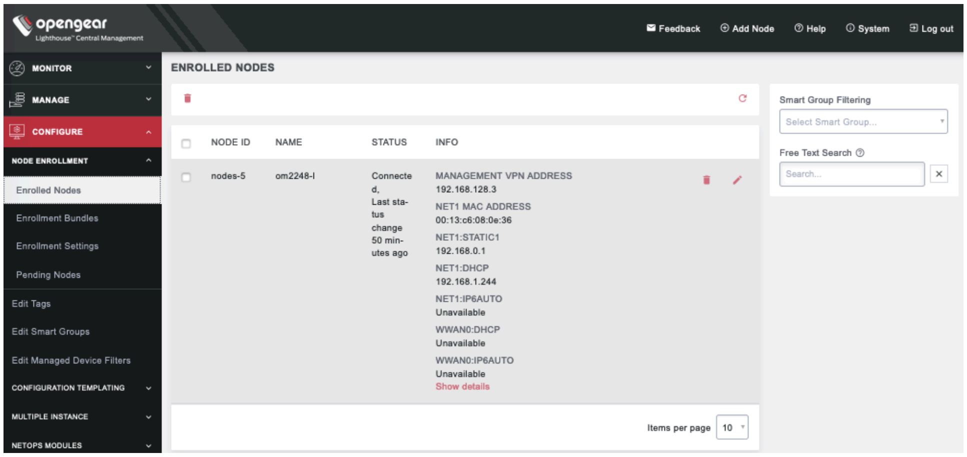
1. Select **CONFIGURE > NODE ENROLLMENT > Enrollment Bundles**. A list of existing **Enrollment Bundles** appears.
2. In the **Actions** column of the particular bundle, click the **download** button, a downward arrow in a circle.
3. Depending on the browser's configuration, a **`manifest.og`** file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



ENROLLED NODES

□	NODE ID	NAME	STATUS	INFO
□	nodes-5	om2248-l	Connected, Last status change 50 minutes ago	MANAGEMENT VPN ADDRESS 192.168.128.3 NET1 MAC ADDRESS 00:13:c6:08:0e:36 NET1:STATIC1 192.168.0.1 NET1:DHCP 192.168.1.244 NET1:IP6AUTO Unavailable WWAN0:DHCP Unavailable WWAN0:IP6AUTO Unavailable

Items per page 10

CONFIGURE > NODE ENROLLMENT > Enrolled Nodes lists all enrolled nodes in the order they are enrolled to *Lighthouse*.

On the bottom right of the page is an **Items per page** drop-down that allows you 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.

Status is the current connection 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:**

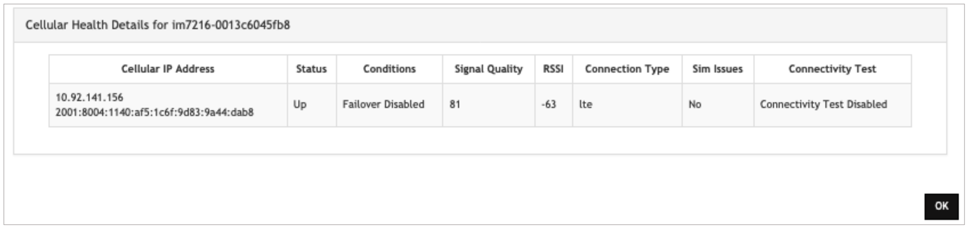
- **Success**: all templates were successfully executed on the node.

- **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.

If **SETTINGS > SERVICES > Cellular Health Reporting** is Enabled, the **Cellular Health** column appears and 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.



Cellular Health Details for im7216-0013c6045fb8

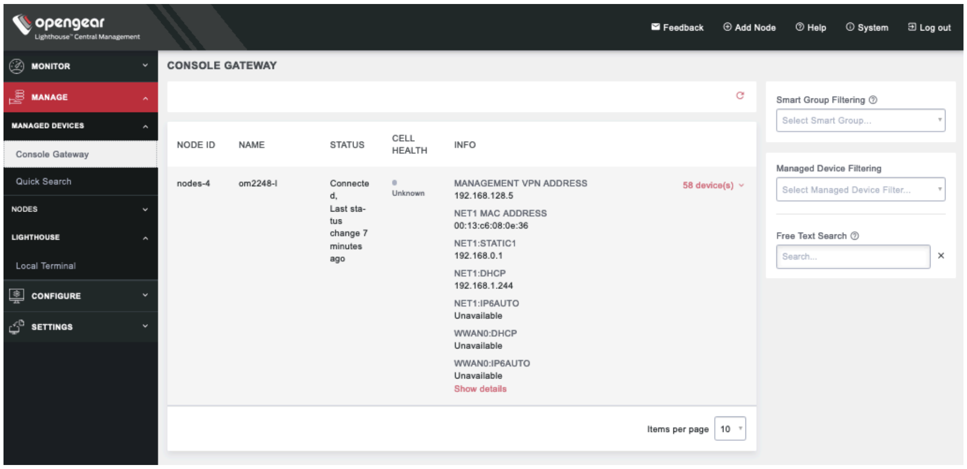
Cellular IP Address	Status	Conditions	Signal Quality	RSSI	Connection Type	Sim Issues	Connectivity Test
10.92.141.156 2001:8004:1140:af5:1c6f:9d83:9a44:dab8	Up	Failover Disabled	81	-63	lte	No	Connectivity Test Disabled

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.



NODE ID	NAME	STATUS	CELL HEALTH	INFO	
nodes-4	om2248-l	Connected, Last status change 7 minutes ago	Unknown	MANAGEMENT VPN ADDRESS 192.168.128.5 NET1 MAC ADDRESS 00:13:c6:08:0e:36 NET1:STATIC1 192.168.0.1 NET1:DHCP 192.168.1.244 NET1:IP6AUTO Unavailable WWAN0:DHCP Unavailable WWAN0:IP6AUTO Unavailable Show details	58 device(s)

7.4.1 Filtering using the Free Text Search field

The Free Text Search input field allows near real-time filtering of nodes and managed devices. It searches over node name, firmware version, management VPN address, MAC address, serial number and port label.

To use the Free Text Search, enter your search term and press the **Enter** key. The Free Text Search field treats multiple search terms separated by the space character as being combined with the logical **AND** operator, returning results only if *all* terms are present in the item.

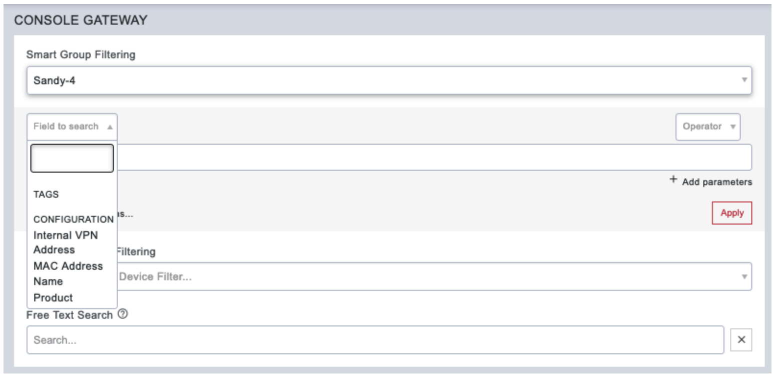
For example, the search phrase `production switch` returns only nodes that contain *both* `production` **AND** `switch` anywhere in searchable fields.

To search for a multi-word term, enclose the search term in double quote characters. For example, `“production switch”` will return results only if the entire search term is matched in the item.

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* for how to create such groups.

Once a particular Smart Group has been selected, further filtering options become available under **Fields to search**:



In the example above, the **CONFIGURE > NODE ENROLLMENT > Enrolled Nodes** page is being filtered on the ***Sandy-4*** Smart Group.

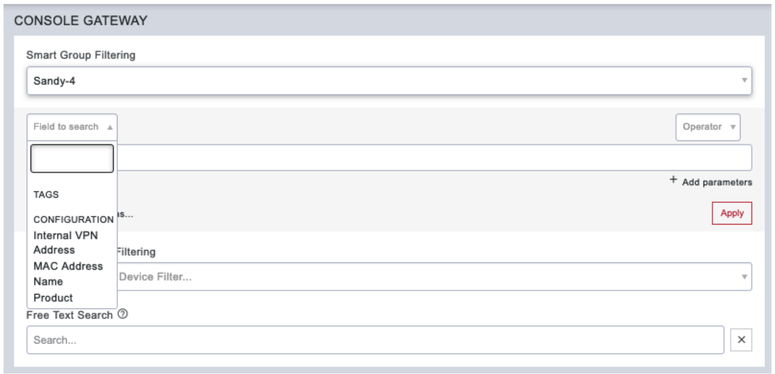
To add more filtering options:

1. Click **Field to search.**
2. Select a field and enter a value in the text box.
3. Select an **Operator** from the drop-down box on the right.
4. Click the **+ Add Parameters button**. Select a parameter.
5. Click **Apply.**

7.4.3 Filtering using the Managed Device Filtering drop-down menu

Selecting from the ***Managed Device Filter*** 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 **Managed Device Filter** has been selected, further filtering options become available under **Fields to search**:



To add more filtering options:

1. Click **Field to search.**
2. Select a field and enter a value in the text box.
3. Select an **Operator** from the drop-down box on the right.
4. Click the **+ Add Parameters button**. Select a parameter.

5. Click **Apply.**

7.5 Creating Smart Groups

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

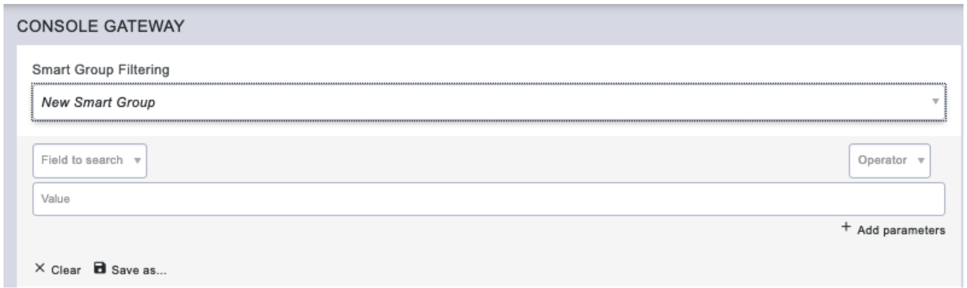
A user group can be linked to a particular **Smart Group**. When a group is linked in this fashion, members of the group inherit rights over all nodes in the group based on the group's role. See *8.3 Modifying existing groups* for how to set a group's role and linked Smart Group.

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***...


CONSOLE GATEWAY

Smart Group Filtering

New Smart Group

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

   These attributes include details about the device (**Internal VPN Address**, ***MAC Address, Name, Product, SSH Port, Firmware Version, Model, Serial Number, Node ID, Connection Status, Cell Health***), 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 ***Field to search*** text box contains tag values.

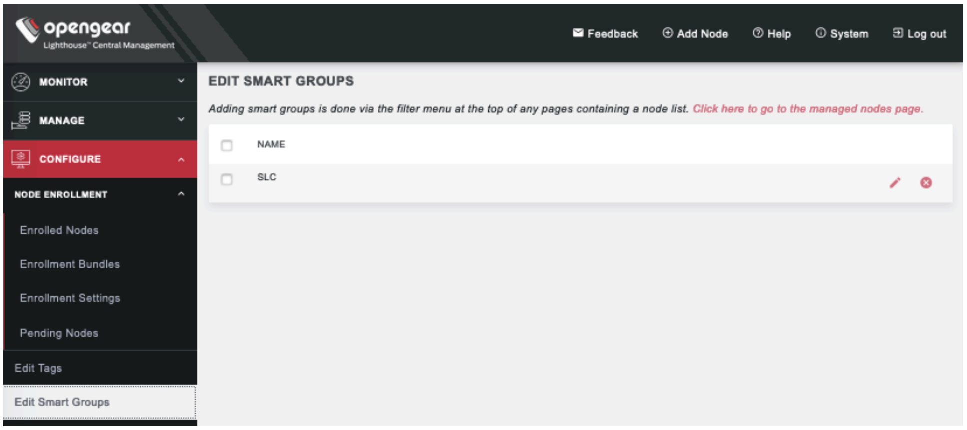
4. Click the ***Operator*** drop-down to select the operator to apply to the ***Value***.
5. Select the ***Value*** to be matched against.
6. Click **Apply** to see the results of the filter.
7. Click **Save As and** type in a name for your new Smart Group.

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**.



EDIT SMART GROUPS

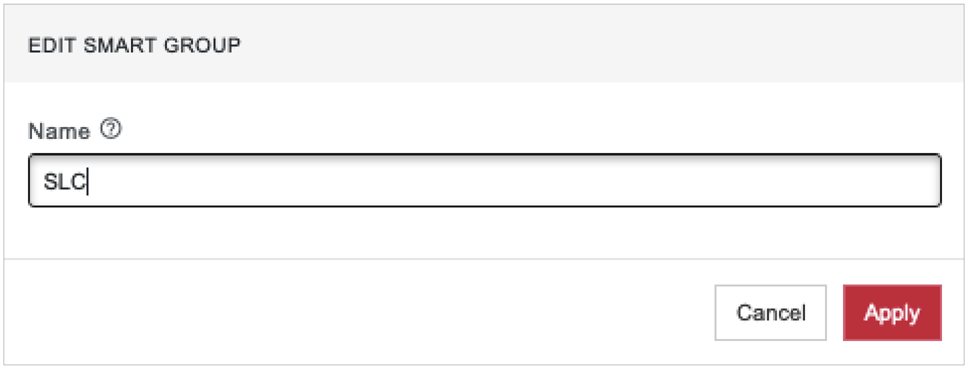
Adding smart groups is done via the filter menu at the top of any pages containing a node list. Click here to go to the managed nodes page.

□	NAME
□	SLC

- Click the **Delete** 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**.



Name: SLC

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

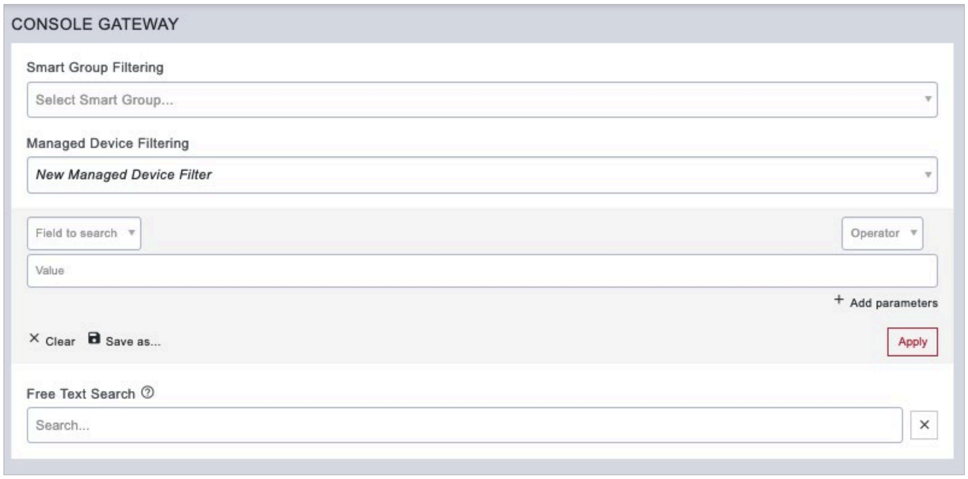
7.7 Creating Managed Device Filters

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***.



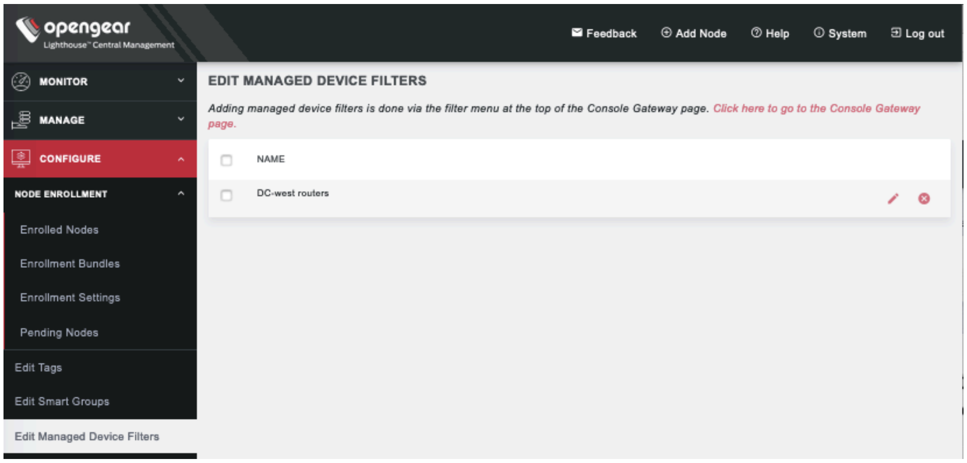
Managed Device Filtering: *New Managed Device Filter*

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***.
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**.



EDIT MANAGED DEVICE FILTERS

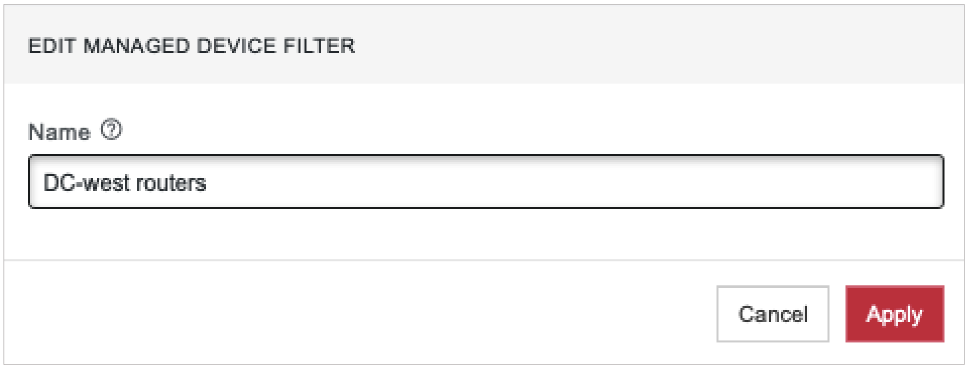
Adding managed device filters is done via the filter menu at the top of the Console Gateway page. Click here to go to the Console Gateway page.

□	NAME
□	DC-west routers

- Click the **Delete** 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.



EDIT MANAGED DEVICE FILTER

Name ⓘ

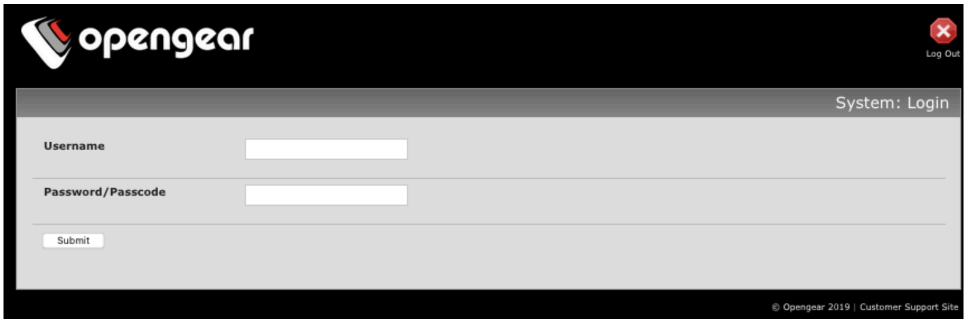
DC-west routers

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**.

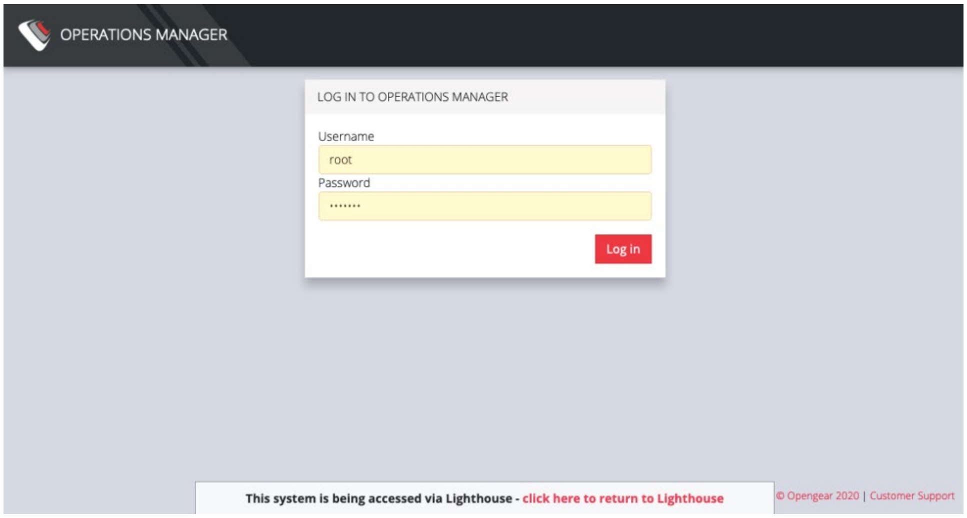
2. In the ***Actions*** column, click the **Access Web UI** link for the particular node.
   The web-based login for that node loads.
3. Authenticate using the username and password required by that node. The appearance of the Web UI depends on which device you have added. Below is the ACM/CM/IM Web UI, followed by the OM Web UI.



Username

Password/Passcode

© Opengear 2019 | Customer Support Site



Username: root

Password: •••••••

This system is being accessed via Lighthouse - click here to return to Lighthouse

© Opengear 2020 | Customer Support

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.

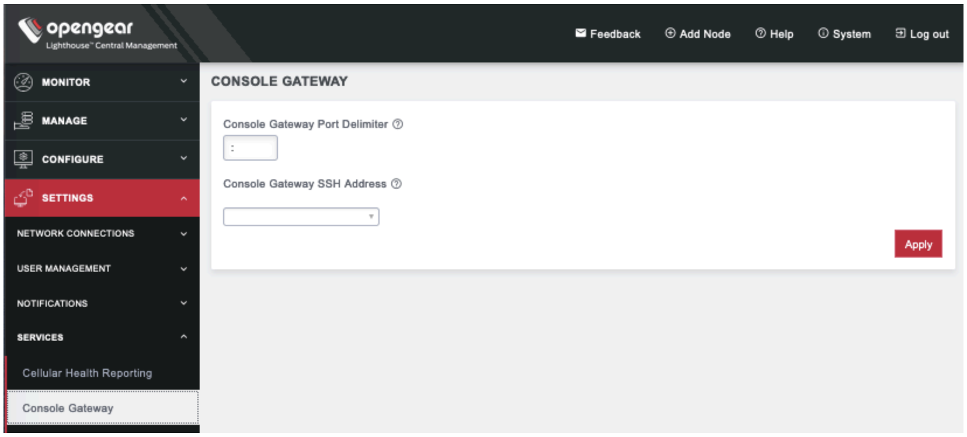
These auto-generated links use the colon (:) as the field-delimiter. The auto-generated SSH link has the following form:

```
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**.



CONSOLE GATEWAY

Console Gateway Port Delimiter

:

Console Gateway SSH Address

- 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.

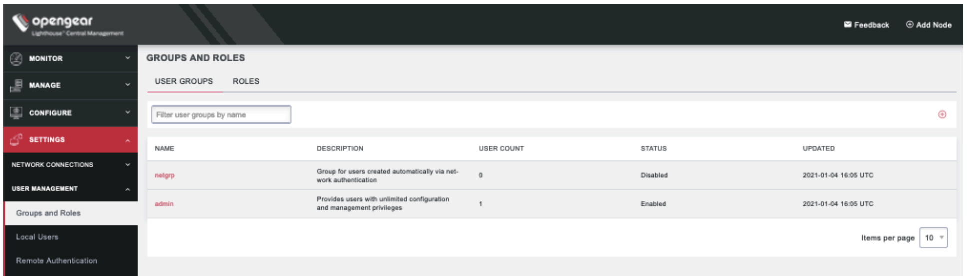
8.1 Password fields in Lighthouse

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 and roles

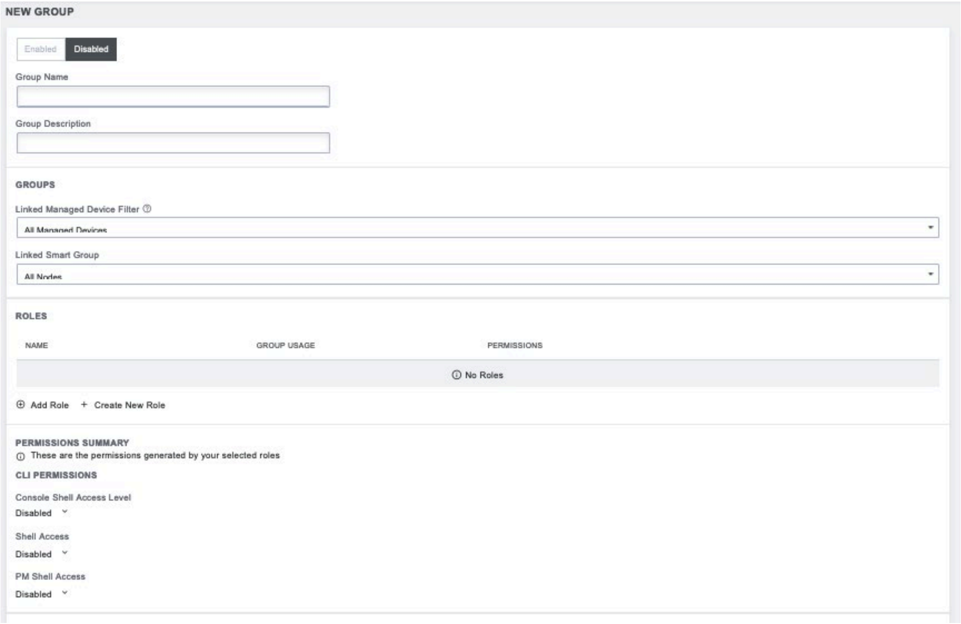
8.2.1 CREATE A NEW GROUP

1. Select **SETTINGS > USER MANAGEMENT > Groups and Roles.** The User Groups tab should be selected.



NAME	DESCRIPTION	USER COUNT	STATUS	UPDATED
netgrp	Group for users created automatically via network authentication	0	Disabled	2021-01-04 16:05 UTC
admin	Provides users with unlimited configuration and management privileges	1	Enabled	2021-01-04 16:05 UTC

2. Click **+** on the upper right. The **New Group** page opens.



NEW GROUP

Enabled | Disabled

Group Name

Group Description

GROUPS

Linked Managed Device Filter: All Managed Devices

Linked Smart Group: All Nodes

ROLES

NAME	GROUP USAGE	PERMISSIONS
No Roles		

PERMISSIONS SUMMARY

These are the permissions generated by your selected roles

CLI PERMISSIONS

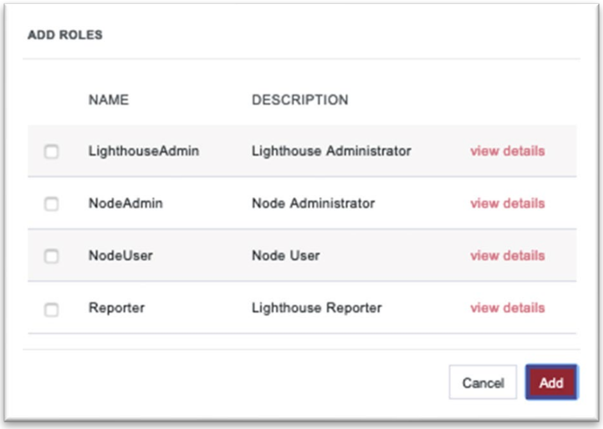
Console Shell Access Level: Disabled

Shell Access: Disabled

PM Shell Access: Disabled

3. Enter a ***Group Name*** and ***Group Description***.

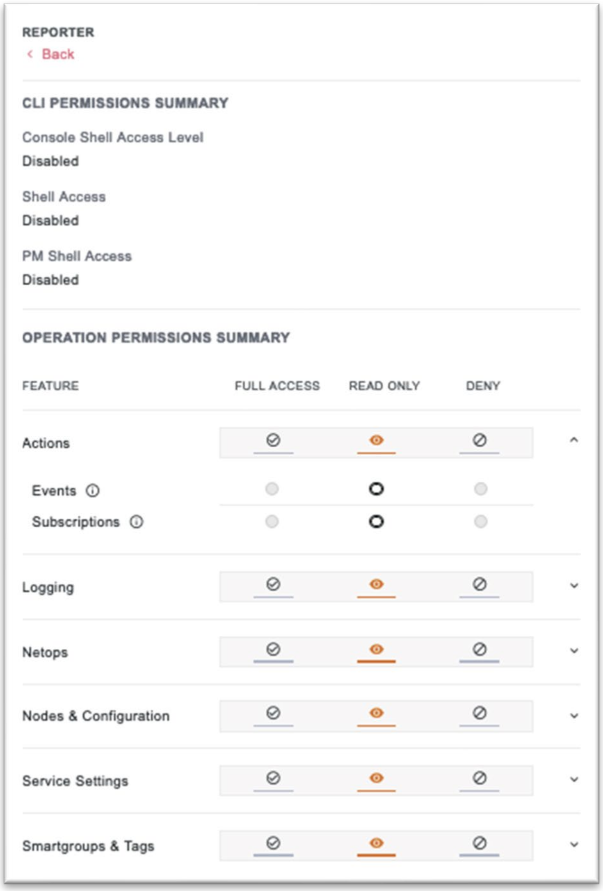
   Group Name is case sensitive. It can contain numbers and some alphanumeric characters. When using remote authentication, characters from a user's remote groups that are not allowed on Lighthouse are converted to underscores during authentication. Local groups can be created that take that into account, allowing the authentication to continue.
4. The **CLI Permissions** section displays permissions based on the roles you have assigned to this group. To change the permissions, you can edit or add new roles with the desired CLI Permissions. See **CREATE A NEW ROLL** below.
5. Click **Enabled** to enable group.
6. If desired, you can select a **Linked Managed Device Filter** and **Linked Smart Grou**p to associate with this group.
7. Add one or more roles by clicking **Add Role** and checking the desired roles.



ADD ROLES

	NAME	DESCRIPTION	
□	LighthouseAdmin	Lighthouse Administrator	view details
□	NodeAdmin	Node Administrator	view details
□	NodeUser	Node User	view details
□	Reporter	Lighthouse Reporter	view details

Each role has specific operation permissions associated with it and **CLI** (**Command Line Interface**) access levels for **console shell**, **shell**, and **PM shell**. Click **view details** to see the information for each group.



REPORTER

‹ Back

CLI PERMISSIONS SUMMARY

Console Shell Access Level
Disabled

Shell Access
Disabled

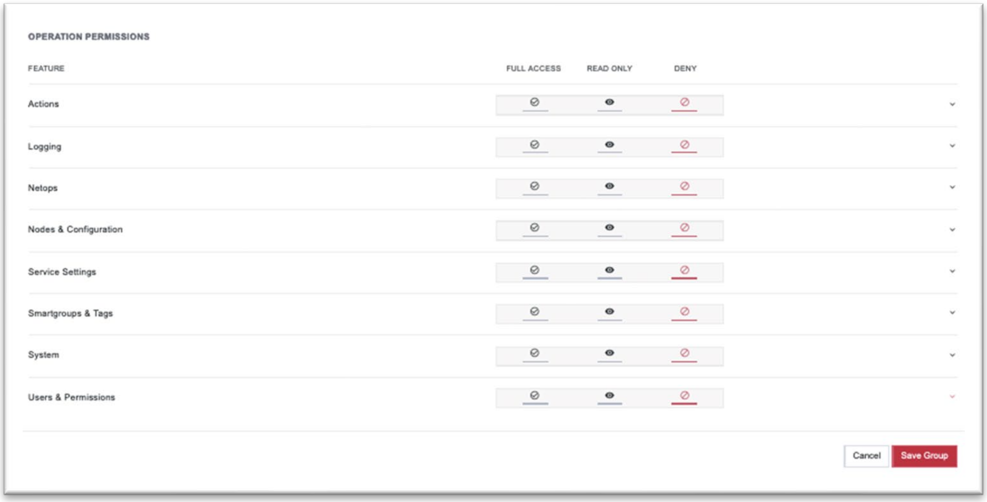
PM Shell Access
Disabled

OPERATION PERMISSIONS SUMMARY

FEATURE	FULL ACCESS	READ ONLY	DENY
Actions	○	●	○
Events ⓘ	○	●	○
Subscriptions ⓘ	○	●	○
Logging	○	●	○
Netops	○	●	○
Nodes & Configuration	○	●	○
Service Settings	○	●	○
Smartgroups & Tags	○	●	○

8. You can also control the new group's permissions independently of the roles you add to your group. Scroll to the bottom of the page to specify **Full Access, Read Only,** or **Deny**. Click to the right of each Operation row to see all options.

   NOTE: See ***Available Operations Permissions*** below for a list of all options.



OPERATION PERMISSIONS

FEATURE	FULL ACCESS	READ ONLY	DENY
Actions	○	○	●
Logging	○	○	●
Netops	○	○	●
Nodes & Configuration	○	○	●
Service Settings	○	○	●
Smartgroups & Tags	○	○	●
System	○	○	●
Users & Permissions	○	○	●

9. Click **Save Group**.

Available Roles:

Lighthouse Administrator: Members of groups with this role have **Full** access to all nodes and managed devices.

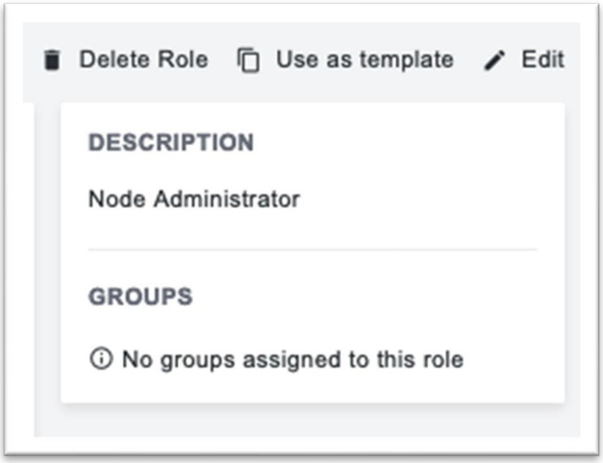
NodeAdmin: Has no shell access. Has **Read Only** access to **Netops Modules**, all **Nodes & Congfiguration Operations, Cell Health, Smart Groups, Tags,** and **Jobs**.

NodeUser: Has PM Shell access. Has **Read Only** access to **Nodes & Devices (Base)** and **Tags**.

Lighthouse Reporter: Has no shell access. Has **Read Only** access to all **Operations**.

You can also create a custom role that allows you to modify **CLI Permissions** and **Operations Permissions** by clicking **Create New Role** on the **New Group** page.

A new role can also be based on an existing role with the **Use as template** link on the upper right of a role's detail page.



DESCRIPTION

Node Administrator

GROUPS

ⓘ No groups assigned to this role

Available Operations Permissions:

Actions
- Events – Enable or disable if events are used to generate notifications.
- Subscriptions – Manage third-party access to events.

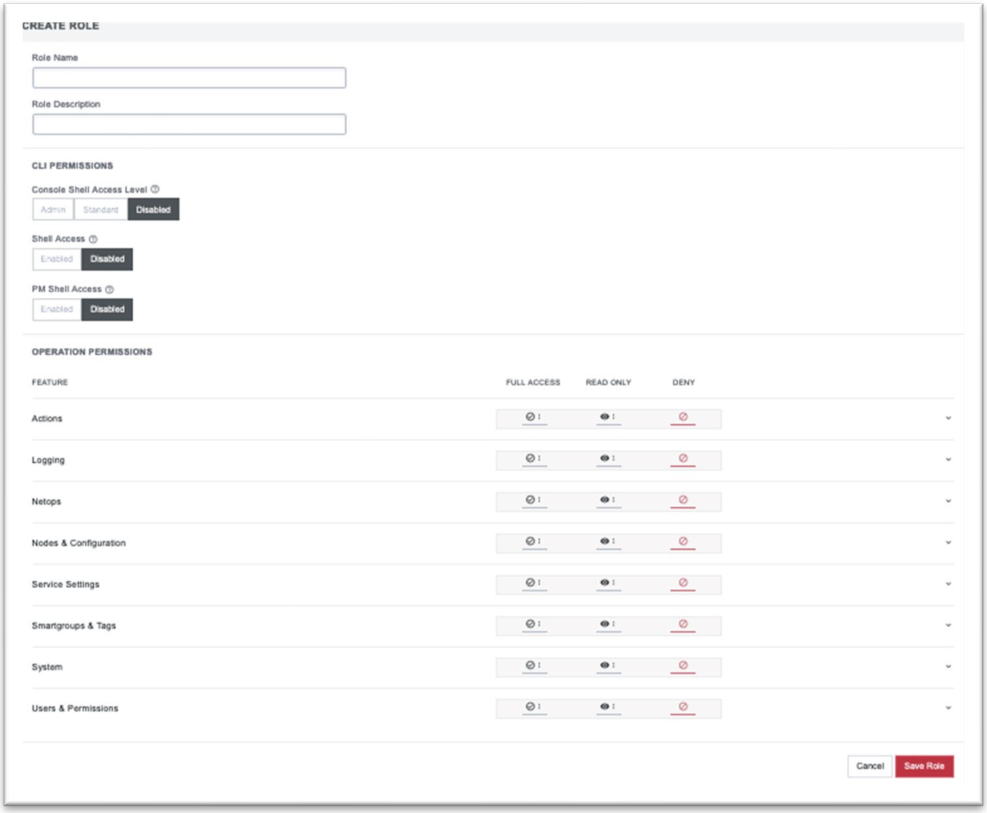
Logging
- Port Logging – Manage port logging settings.
- Syslog – Manage system syslog settings.

Netops
- Netops Modules

Nodes & Configuration
- Nodes & Devices (Base) – Access to dashboard, nodes, managed devices, node enrollment, console gateway, and Node web UI.
- Nodes & Devices (Advanced) – Access to jobs, pending nodes, smart groups, and managed device filters.
- Template Push – Manage templates and push templated to nodes.

Service Settings

  - LHVPN
  - Cell Health
  - Console Gateway
  - Date & Time
  - HTTPS
  - Netops – Install Netops modules and manage local Netops repositories.
  - Node Backup
  - Session Settings
  - SNMP
  - SSH
  - Syslog
- Smartgroups & Tags
  - Bundles – Manage and use bundles.
  - Smart Groups – Manage and use smart groups.
  - Tags – Manage and use tags.
- System
  - Admin & Licensing – Manage access settings for Lighthouse and license settings.
  - Backup & Restore
  - Jobs
  - Multi-instance – Manage multi-instance settings and control state of instances.
  - Network Interfaces – Manage network interface settings.
  - System Upgrade & Reset
- Users & Permissions
  - Authentication – Manage authentication settings including methods, policies, and restrictions.
  - Groups & Roles – Create and edit groups and roles. May not assign them to users.
  - Users – View, manage, create, and delete users.



CREATE ROLE

Role Name

Role Description

CLI PERMISSIONS

Console Shell Access Level: Admin | Standard | [✓ Disabled]

Shell Access: Enabled | [✓ Disabled]

PM Shell Access: Enabled | [✓ Disabled]

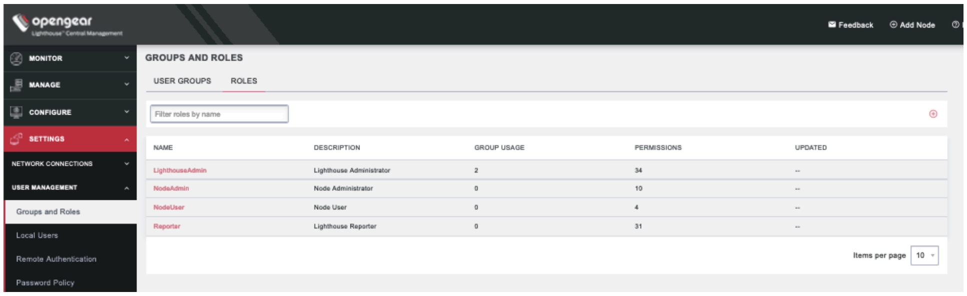
OPERATION PERMISSIONS

FEATURE	FULL ACCESS	READ ONLY	DENY
Actions			
Logging			
Netops			
Nodes & Configuration			
Service Settings			
Smartgroups & Tags			
System			
Users & Permissions			

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.

8.2.2 CREATE A NEW ROLE

1. Select **SETTINGS > USER MANAGEMENT > Groups and Roles.**
2. Click the **ROLES** tab.



GROUPS AND ROLES

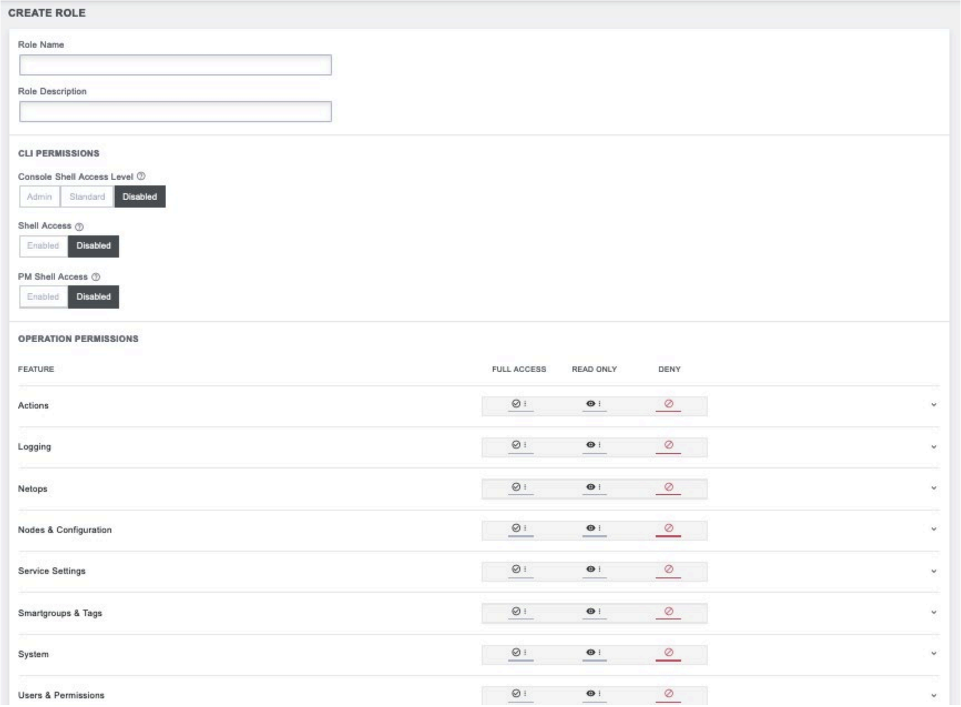
USER GROUPS | ROLES

Filter roles by name

NAME	DESCRIPTION	GROUP USAGE	PERMISSIONS	UPDATED
LighthouseAdmin	Lighthouse Administrator	2	34	--
NodeAdmin	Node Administrator	0	10	--
NodeUser	Node User	0	4	--
Reporter	Lighthouse Reporter	0	31	--

Items per page 10

3. Click **+** on the upper right. The **Create Role** page opens.



CREATE ROLE

Role Name

Role Description

CLI PERMISSIONS

Console Shell Access Level: Admin | Standard | **Disabled**

Shell Access: Enabled | **Disabled**

PM Shell Access: Enabled | **Disabled**

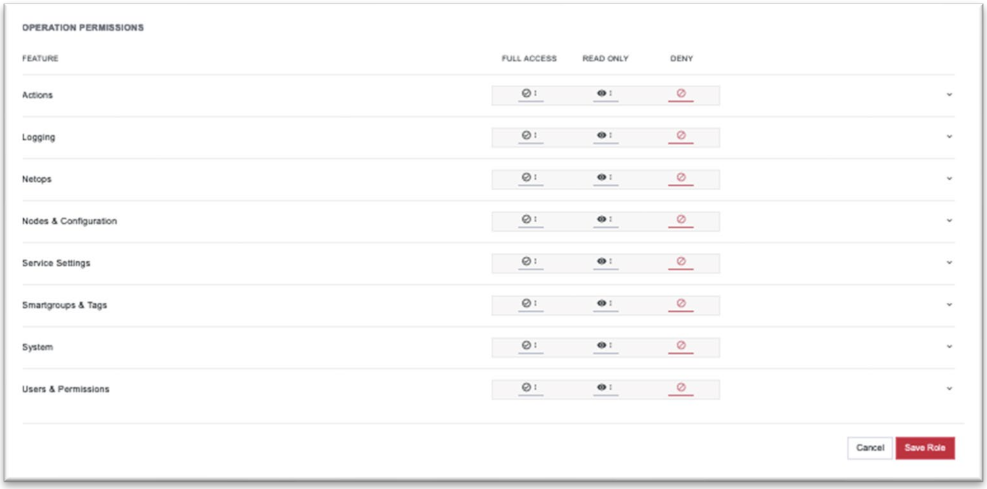
OPERATION PERMISSIONS

FEATURE	FULL ACCESS	READ ONLY	DENY
Actions			
Logging			
Netops			
Nodes & Configuration			
Service Settings			
Smartgroups & Tags			
System			
Users & Permissions			

4. Enter a ***Role Name*** and ***Role Description***.
5. Modify the **CLI Permissions** for as desired.
   Console Shell Access: Ability to connect to nodes' command lines via Lighthouse's SSH.
   Shell Access: Ability to access Lighthouse's command line as administrator.
   PM Shell Access: Ability to connect to serial ports via SSH.

6. You can also control the new roles **Operation Permissions** independently. Specify **Full Access, Read Only,** or **Deny**. Click to the right of each Operation row to see all options.

   NOTE: See ***Available Operations Permissions*** for a list of all options.

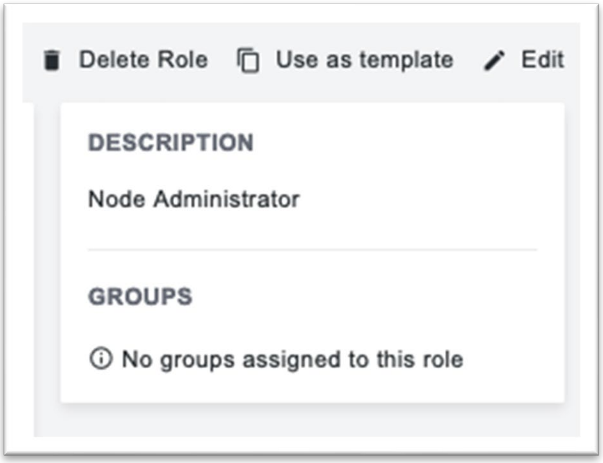


OPERATION PERMISSIONS

FEATURE	FULL ACCESS	READ ONLY	DENY
Actions			
Logging			
Netops			
Nodes & Configuration			
Service Settings			
Smartgroups & Tags			
System			
Users & Permissions			

7. Click **Save Role**.

A new role can also be based on an existing role with the **Use as template** link on the upper right of a role's detail page.



DESCRIPTION

Node Administrator

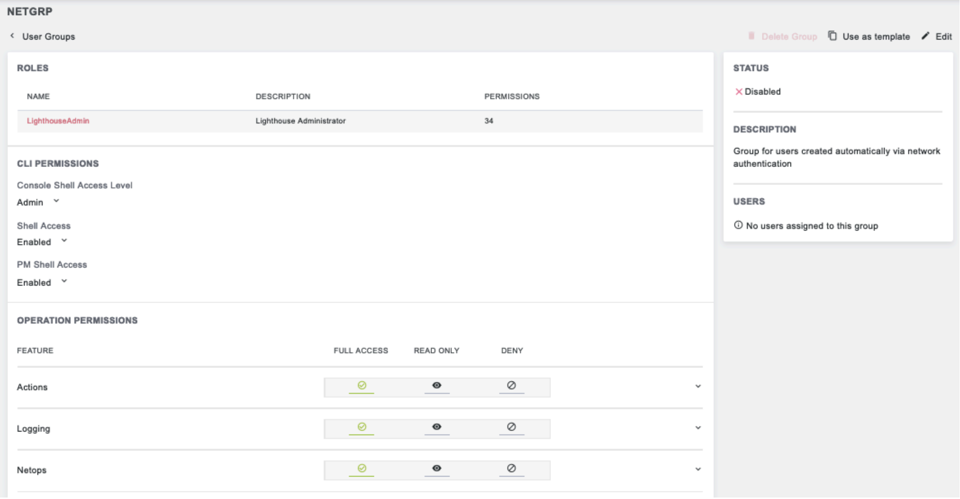
GROUPS

No groups assigned to this role

8.3 Modifying existing groups

To modify an existing group:

1. Select **SETTINGS > Groups and Roles**. If necessary, click on the **USER GROUPS** tab.
2. Click the name of the group to be modified.
3. Click the **Edit** icon on the upper right and make desired changes.
4. Click **Save Group**.



ROLES

NAME	DESCRIPTION	PERMISSIONS
LighthouseAdmin	Lighthouse Administrator	34

CLI PERMISSIONS

Console Shell Access Level: Admin

Shell Access: Enabled

PM Shell Access: Enabled

OPERATION PERMISSIONS

FEATURE	FULL ACCESS	READ ONLY	DENY
Actions	✓	●	⊘
Logging	✓	●	⊘
Netops	✓	●	⊘

STATUS

× Disabled

DESCRIPTION

Group for users created automatically via network authentication

USERS

No users assigned to this group

The group details page 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 *7.5 Creating Smart Groups* 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 you to delete groups. All users who were members of the deleted group lose any access and administrative rights inherited from the group.

NOTE: 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.

8.4 Use an existing group as a template for a new group

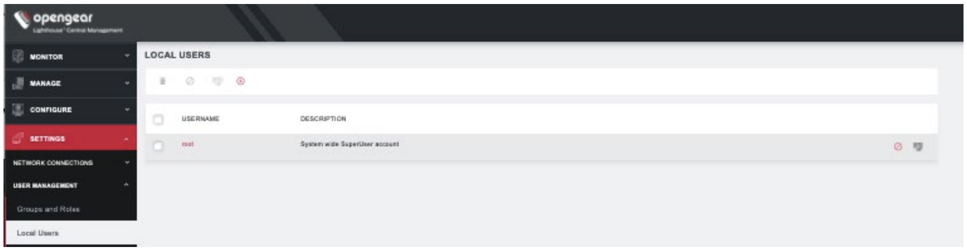
To use an existing group as a template for a new group:

1. Select **SETTINGS > USER MANAGEMENT > Groups and Roles**. If necessary, click on the **USER GROUPS** tab.
2. Click the name of the group to be used as a template.
3. Click the **Use as Template** icon on the upper right. This opens a new group page with the settings from the group you selected as a template.
4. Change the **Group Name** and make and other desired changes.
5. Click **Save Group**.

8.5 Creating new local users

To create a new local user:

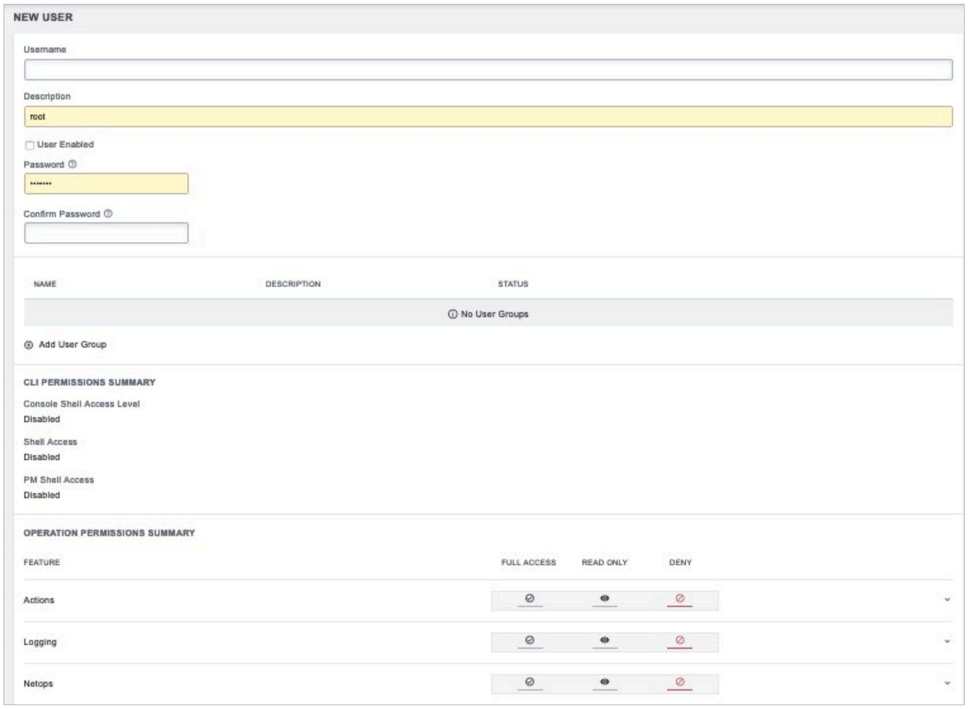
1. Select **SETTINGS > USER MANAGEMENT > Local Users.**
   By default, the root user is the only user listed.



LOCAL USERS

	USERNAME	DESCRIPTION
□	root	System wide SuperUser account

2. Click the **+** button. The **New User** page appears.



NEW USER

Username

Description

root

□ User Enabled

Password

Confirm Password

NAME	DESCRIPTION	STATUS
No User Groups		

Add User Group

CLI PERMISSIONS SUMMARY

Console Shell Access Level
Disabled

Shell Access
Disabled

PM Shell Access
Disabled

OPERATION PERMISSIONS SUMMARY

FEATURE	FULL ACCESS	READ ONLY	DENY
Actions	⊘	●	⊘
Logging	⊘	●	⊘
Netops	⊘	●	⊘

3. Enter a ***Username***, ***Description***, and ***Password***.
4. Re-enter the ***Password*** in the ***Confirm Password*** field.
5. Click on Add User Group to add this user to any existing **User Groups.**
6. Choose which **CLI** options this user should have.
7. Select any **Operation Permissions** this user should have.
8. Select the **User *Enabled*** checkbox.
9. Click **Save User**.

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

1. Select **SETTINGS > USER MANAGEMENT > Remote Authentication**
2. Choose a **Scheme** and enter appropriate settings.
3. Select **SETTINGS > USER MANAGEMENT > Local Users**
4. Click the **+** button. The **New User** dialog loads.
5. Enter a ***Username*** and ***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, database queries, and the time that it occurred:

```
2020-05-22T16:22:46.490627+01:00 localhost rest_api_log[62]: GET 200 (root |
192.168.1.230) - /api/v3.5/users?page=1&per_page=10 RESPONSE={'users': [{'username':
'root', 'description': 'System wide SuperUser account', 'enabled': True, 'id': 'users-
1', 'no_password': False, 'expired': False, 'locked_out': False, 'rights': {'delete':
True, 'modify': True}, 'groups': ['groups-2']}, {'username': 'fred', 'description':
'fred', 'enabled': True, 'id': 'users-2', 'no_password': False, 'expired': False,
'locked_out': False, 'rights': {'delete': True, 'modify': True}, 'groups': ['groups-
2']}], 'meta': {'total_pages': 1}}
```

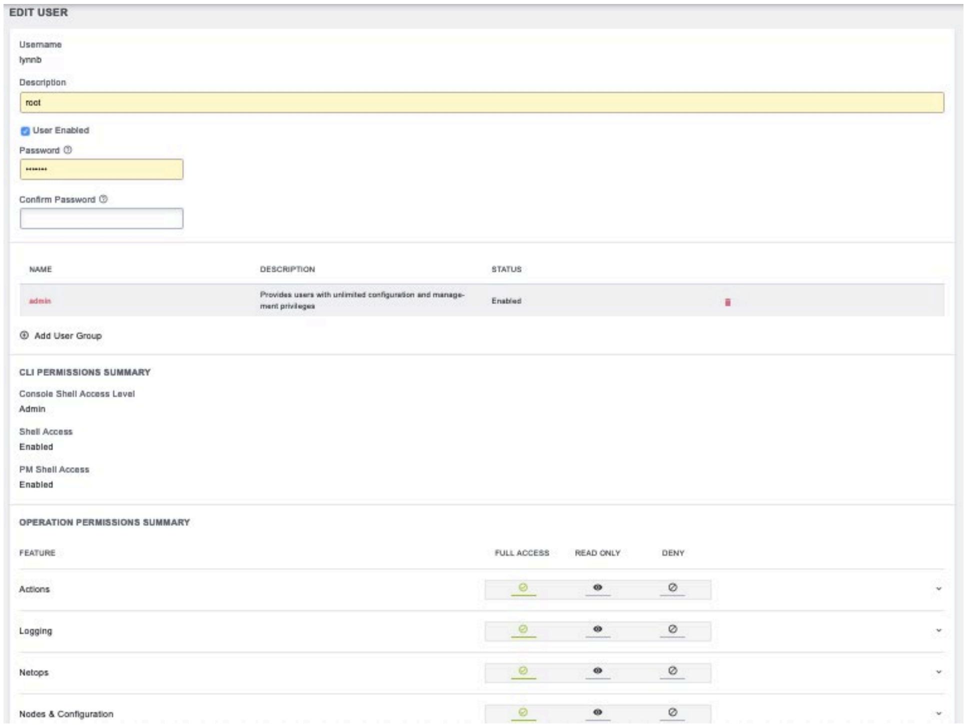
If the created user is set to disabled, the `configurator_users` message does not appear as they have not been added to the passwords file.

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

8.6 Modifying existing users

To edit settings for an existing user:

1. Select **SETTINGS > USER MANAGEMENT > Local Users**
2. Click the **name** of the user to **edit** this user and make desired changes. You may change CLI access and specific access to operations. You may also control which groups this user is a member of.
3. Click **Save User**.



EDIT USER

Username
lynnb

Description
root

☑ User Enabled

Password
•••••••

Confirm Password

NAME	DESCRIPTION	STATUS
admin	Provides users with unlimited configuration and management privileges	Enabled

Add User Group

CLI PERMISSIONS SUMMARY

Console Shell Access Level
Admin

Shell Access
Enabled

PM Shell Access
Enabled

OPERATION PERMISSIONS SUMMARY

FEATURE	FULL ACCESS	READ ONLY	DENY
Actions	✓	👁	⊘
Logging	✓	👁	⊘
Netops	✓	👁	⊘
Nodes & Configuration	✓	👁	⊘

The **Edit Users** dialog also 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.

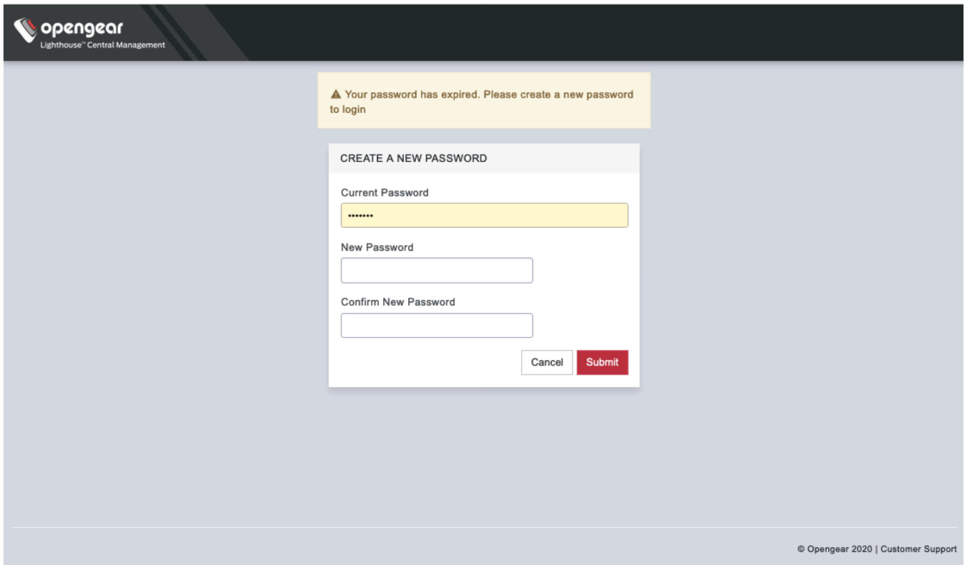
Disabled users cannot login to Lighthouse using either the Web-based interface or via shell-based logins (i.e. *sshusername-disabled@lighthouse-name-or-ip*). The user and the **`/home/`***username-disabled* directory still exist in the Lighthouse VM file system.

8.7 Expire user password

You can set user passwords to expire. To expire a user’s password:

1. Select **SETTINGS > USER MANAGEMENT > Local Users**
2. Click the ***Expire Password*** button in the **Actions** section of the user to be deleted.
3. Click ***Yes*** in the **Confirmation** dialog.

The next time this user logs in, the user will be required to change the password.



opengear
Lighthouse™ Central Management

⚠ Your password has expired. Please create a new password to login

CREATE A NEW PASSWORD

Current Password: •••••••

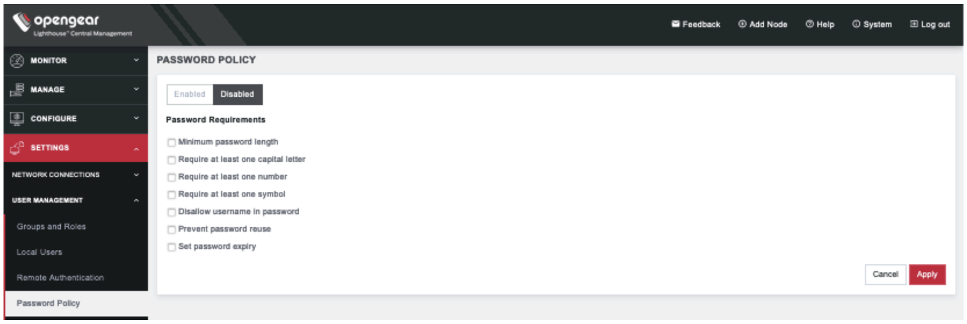
New Password:

Confirm New Password:

Cancel
Submit
© Opengear 2020 | Customer Support

8.8 Setting password policy

Lighthouse Administrators can apply **Password Policies**. To do so, select **SETTINGS > USER MANAGEMENT > Login Restrictions, choose Enabled,** and click **Save**. Click **Password Policy.**



PASSWORD POLICY

Enabled [✓ Disabled]

Password Requirements

- ☐ Minimum password length
- ☐ Require at least one capital letter
- ☐ Require at least one number
- ☐ Require at least one symbol
- ☐ Disallow username in password
- ☐ Prevent password reuse
- ☐ Set password expiry

Choose one or more options from the following:

- Minimum password length – from 1 to 128
- Require at least one capital letter
- Require at least one number
- Require at least one symbol
- Disallow username in password
- Prevent password reuse – choose a number of days or select **Always**
- Set password expiry – select a number of days until passwords expire. At next login, the user will need to reset the password.

8.9 Deleting users

To delete a user:

1. Select **SETTINGS > USER MANAGEMENT > Local Users**
2. Click the ***Delete*** button in the **Actions** section of the user to be deleted.

Click *Yes* in the **Confirmation** dialog.

8.10 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.

An Identity Provider (IdP) stores and manages users' digital identities An IdP may check user identities via username-password combinations and other factors, or it may simply provide a list of user identities that another service provider (like an SSO) checks. An IdP can authenticate any entity connected to a network or a system, including computers and other devices.

8.11 SAML Configuration for SSO

SAML is the framework used to integrate applications with identity providers for single sign-on (SSO). This is mostly (if not completely) reflected when a user logs in and authenticates with their IdP or is already logged in (if they have already authenticated with their identity provider prior to accessing Lighthouse).

Lighthouse supports the independent, concurrent use of both SAML and AAA authentication. SAML authentication is independent of, and does not interact with, other authentication methods.

Note: For release 21.Q4 SAML is only supported for authentication to the lighthouse Web GUI.

When SAML is configured and enabled, users can authenticate to the Lighthouse Web GUI either through SAML or another configured authentication mechanism such as Local or AAA. Users can SSH only via the other configured authentication mechanism (Local or AAA) if Remote Authentication is configured.

The default authentication is Local, with Lighthouse using locally defined users and groups. If AAA (TACACS, RADIUS or LDAP) Remote Authentication is configured, this will be used for Web GUI and SSH login authentication to Lighthouse (except for root which is always locally authenticated). Users are authenticated against AAA server(s) with group membership returned to Lighthouse, which is used to determine user roles and permissions. AAA Remote Authentication can support 2FA/MFA, depending on the AAA server capabilities.

Lighthouse's SAML integrates with the following identity providers:

- OKTA
- Azure Active Directory (Azure AD)
- One Login

Note: In the following instructions, any text in braces **{}** for example, `{main lighthouse address}` needs to be substituted with the value for your environment.

Common values you will need are:

`{main lighthouse address}` - The address (without the protocol or path) most users use to connect to your primary lighthouse's web interface. e.g. `lighthouse.example.com` or `192.168.1.10`

`{provider}` - Each IdP implements the spec slightly differently lighthouse needs to know which style to expect to handle these differences. If your IdP is not one of our officially supported IdPs, try configuring lighthouse using the `generic` provider option as the most widely applicable. (You could also try using our other explicit IdP options but these often expect provider specific intricacies).

Please also review the limitations of the Lighthouse Saml SSO feature.

8.11.1 Generic IdP Setup

Generic IdP Application Integration

Note: You must have your user groups setup in lighthouse prior creating & assigning them via the IdP. See the example in step 6 of the Okta configuration later in this topic.

Note: The `{provider}` in the steps must exactly match one of our provider strings i.e. generic, okta, azure_ad, onelogin.

1. Create an application integration for "Lighthouse" in your IdP
2. Set ACS or consumer URL as `https://{main lighthouse address}/api/v3.7/sessions/saml/sso/{provider}`
3. Set the Allowed SSO URLs or Allowed redirect URLs or ACS URL Validator to include or match the `/saml/sso/` URL for each address of each of your lighthouses that you want users to be able to login from.

Example:

```
https://{main lighthouse address}/api/v3.7/sessions/saml/sso/{provider}
https://{main lighthouse ip address}/api/v3.7/sessions/saml/sso/{provider}
https://{secondary lighthouse
address}/api/v3.7/sessions/saml/sso/{provider}
```

Depending on your IdP you may need to include the `/saml/sp_init/` URLs.

```
https://{main lighthouse address}/api/v3.7/sessions/saml/sso/{provider}
https://{main lighthouse address}/api/v3.7/sessions/saml/sp_init/{provider}
https://{main lighthouse ip address}/api/v3.7/sessions/saml/sso/{provider}
https://{main lighthouse ip
address}/api/v3.7/sessions/saml/sp_init/{provider}
https://{secondary lighthouse
address}/api/v3.7/sessions/saml/sso/{provider}
https://{secondary lighthouse
address}/api/v3.7/sessions/saml/sp_init/{provider}
```

4. Set the Service Provider `EntityID` or Audience as `lighthouse-{provider}`
5. If your service provider requires you to configure the `Recipient`

    - And only allows a single value
    - And you run multiple lighthouses or access lighthouse via multiple addresses
    - Then either:

        - Set the recipient as `lighthouse-{provider}` and use the `onelogin` option as your provider configuration.
        - Or if you only access each via a single address you could create a separate application integration per lighthouse.

6. If your IdP has the option then set the initiator to the `Service Provider`
7. Set your IdP to sign the **Assertion** for SAML.

8.11.2 Generic IdP SAML Attribute

You will also need to configure your IdP to send an additional attribute **`LH_Groups`** as part of the SAML response.

In most IdPs this is done by adding an Attribute Statement or Parameter configuration in your application integration. This parameter should be set as a multi-value parameter I.e. multiple values should be provided by multiple duplicative either Attribute Value tags or Attribute tags in the SAML assertion.

We recommend setting the value of this attribute to be populated with the names of the user's Roles (or Groups) in your IdP. This method allows you to create roles in your IdP with the same names as the user groups on your lighthouse that can be assigned in your IdP to grant users that level of access to lighthouse.

Alternatively, you can populate the `LH Groups` attribute with the names of the lighthouse user groups the user should be granted by any other mechanism that your IdP provides. i.e. custom user properties

Note: Your IdP can populate the `LH_Groups` attribute to place users in any lighthouse user group except lighthouse's default `admin` group. You can allow users to login with admin privileges by simply creating another user group in lighthouse with the admin role and assigning the matching role/group in your IdP to the user (i.e. populate `LH_Groups` to include its value).

8.11.3 Lighthouse Setup

You will need to export an IdP metadata xml file for your lighthouse application integration from your IdP. If your IdP requires that requests be signed by the Service Provider then you will also need to provide an x509 certificate & private key in .pem format (either exported from your IdP or created locally then configured in your IdP).

1. Upload your IdP metadata XML (and if required certificate & private key) to your primary lighthouse `i.e. scp`
2. Use the `saml-idp-metadata` command to configure each lighthouse individually. Each lighthouse is configured individually with the same or a different metadata xml (and certificate + key).

   Note: the commands to configure each lighthouse individually, all must be run from your primary lighthouse.

```
# Example: Configuring a Multi-Instance Lighthouse for Okta IdP
# List initial lighthouse configurations (i.e. none)
saml-idp-metadata list
# Configure Primary lighthouse
saml-idp-metadata create \
--metadata metadata.xml \
--provider okta \
--lh-id 1
#Configure Secondary lighthouse
saml-idp-metadata create \
--metadata metadata.xml \
--provider okta \
--lh-id 2
# List lighthouse configurations (i.e. both lighthouses configured)
saml-idp-metadata list
```

8.11.4 Examples of Specific IdP Setups

The following are examples of how you could configure officially supported IdPs. They are based on the above generic step and the IdP's configuration options as of 10/2021.

Okta

Create an Application

You need to create an application that Okta will be doing authentication on behalf of. **NOTE:** you'll need to know what the addresses of your lighthouses before creating the application.

1. In the Okta web console go to **Applications** - > **Applications**
    a. Click **Create App Integration**
    b. Select **SAML 2.0**
2. Give the application a name: e.g. Lighthouse
    a. click **Next**
3. For the **Single sign on URL** enter:

    ```
    https://{main lighthouse address}/api/v3.7/sessions/saml/sso/okta
    ```

    a. Tick both:
        i. Use this for Recipient URL and Destination URL
        ii. Allow this app to request other SSO URLs
    b. Fill out the **Requestable SSO URLs** with the SSO URLs for every lighthouse address you want to be able to sign in with. i.e. IP addresses and DNS address for both your primary and secondary lighthouses. Example:

    ```
    https://{main lighthouse address}/api/v3.7/sessions/saml/sso/okta
    https://{main lighthouse ip address}/api/v3.7/sessions/saml/sso/okta
    https://{secondary lighthouse address}/api/v3.7/sessions/saml/sso/okta
    ```

4. For the **Audience URI (SP Entity ID)** enter `lighthouse-okta`
5. Set **Name ID format** to email.
6. Set **Application username** to email.
7. There are many ways you could configure Okta to populate the `LH_Groups` attribute, our recommended way is to populate it from and manage it via the user's Okta groups:
    a. Add a Group Attribute Statement

        i. Name: `LH_Groups`
        ii. Name Format: `Basic`
        iii. Filter: `Matches Regex .*`

8. Click **Next** and finish.

IdP Metadata

In the **Sign On** tab for your Okta application, find the line:

"Identity Provider metadata is available if this application supports dynamic configuration".

Open the link and save the page as an XML file (recommend naming the file okta_metadata.xml). This is the metadata XML file that you will need to configure lighthouse.

Configure Lighthouse

1. Copy the Identity Provider metadata XML to your primary lighthouse.

2. Using `saml-idp-metadata` on your primary lighthouse, configure each of your lighthouses to use your IdP
   e.g. `saml-idp-metadata -p {root password} create -m /path/to/okta_metadata.xml -P okta -n "My Okta display name" -l {LH id number}`

Groups setup

1. If you do not already have your own User groups setup in lighthouse:
   a. Login to Lighthouse as a local user (or any non-SAML user) i.e. root

      Note: After this initial setup, you will be able to login as a SAML user.

   b. Create the User groups with the Roles and permission that you desire. See "10.1 Creating new users and groups templates" in the Lighthouse User manual.
2. In Okta go to **Directory** > **Groups**
3. Click **Add Group**
4. Enter the Group name that matches a Group name on lighthouse.
5. Open your new group
6. Go to **Manage Apps**
7. Search for your lighthouse app and click **Assign**.
8. Click **Done**.
9. Go to **Manage People**
10. Search for and click on the users you wish to add to the group.

The assigned users are now able to login to lighthouse with the permission levels which that group grants them.

Onelogin

Create an Application.

1. Go to **Applications** > **Add App** > Search for and choose `SAML Custom Connector (Advanced)`
2. Name your connector i.e. Lighthouse
3. In the Configuration tab for your new app
   a. Set **Audience (EntityID)** to `lighthouse-onelogin`
   b. Set **Recipient** to `lighthouse-onelogin`
   c. Set **ACS (Consumer) URL** to:

```
https://{main lighthouse address}/api/v3.7/sessions/saml/sso/onelogin
```

   d. Set **ACS (Consumer) URL Validator** to a regex expression that matches only all your lighthouses' SSO addresses (IP & DNS for Primary & Secondary lighthouses).
      i. Ensure it begins with **^** and ends with **$** to match the whole url.
      ii. Recommended pattern:
      `^https:\/\/` {lighthouse addresses}
      `\/api\/v3\.7\/sessions\/saml\/sso\/onelogin$`
      iii. For example to allow Onelogin login for lighthouse addresses `192.168.1.10` and `lighthouse.example.com`, you could use the following: (note the additional `()` around your hostnames and the `|` separating them.

```
^https:\/\/(192\.168\.1\.10|lighthouse\.example\.com)\/api\/v3\.7\/sessions\/
saml\/sso\/onelogin$
```

e. Set **Login URL** to

```
https://{main lighthouse address}/api/v3.7/sessions/saml/sp_init/onelogin
```

f. Set **SAML initiator** to `Service Provider`
g. Set **SAML signature element** to `Assertion`

4. The recommended method to populate **`LH_Groups`** is with Onelogin Roles.

    a. Go to **Parameters** then click **Add.**
    b. Set **Field Name** to `LH_Groups`
    c. Tick `Include in SAML assertion`
    d. Tick `Multi-value parameter`
    e. Click **Save.**
    f. Set Default value to `User Roles`
    g. If you intend on filtering the Roles that are sent to lighthouse (using a Rule) set `no transform` otherwise set `semicolon delimited`.

    An example Rule to filter roles:

    - “Set LH_Groups in”
    - for each `role`
    - with a value that matches `LH_.*`

    h. Save the parameter.

5. Save the connector.

IdP Metadata

1. Open your Onelogin application.
2. Go to **More Actions** > **SAML Metadata.** This is the metadata xml file that you will need to configure lighthouse.

Configure Lighthouse

1. Copy the metadata xml to your primary lighthouse.
2. Using `saml-idp-metadata` on your primary lighthouse, configure each of your lighthouses to use your IdP.
   e.g. `saml-idp-metadata -p {root password} create -m /path/to/metadata.xml -P onelogin -n "My Onelogin display name" -l {LH id number}`

Roles Setup

If you do not already have your own **Usergroups** setup in lighthouse:

1. Login to Lighthouse as a local user (or any non-SAML user) i.e. root

   Note: After this initial setup, you will be able to login as a SAML user.

2. Create the Usergroups with the Roles and permission that you desire. See "*8.2 Creating new groups and roles*" in the Lighthouse User manual.
3. In Onelogin Go to **Users** > **Roles**
4. Click **New Role**

   b. Set the Role's name to match the lighthouse group you want it to map to.
   c. Select your Lighthouse app to associate the role with.
   d. Save.

5. Open your role.
6. Go to the **Users** tab on the left.
7. Search for and add your users or create a mapping to automatically add multiple users.
8. Save.

   a. If you used a mapping then go to **Users** > **Mappings** and run **Reapply All Mappings**

Done; the assigned users are now able to login to lighthouse with the permission levels which that Onelogin Role/Lighthouse group grants them.

Azure Active Directory

Lighthouse can be added as an **Enterprise application** to Azure Active Directory. This example uses "App roles" to grant users permissions.

Create an Application (Enterprise applications)

1. Go to **Azure Active Directory**
2. Go to **Enterprise applications**
3. Click **New Application**
4. Click **Create your own application**
5. Select **Integrate any other application you don't find in the gallery (Non-gallery)**
6. Name your Application i.e. Lighthouse, then click **Create**
7. Go to **Properties**
   a. Set **`Assignment required`** to **Yes**
   b. Set **`Enabled for users to sign-in`** to **Yes**
   c. Click **Save**
8. Go to **Single sign-on**
   a. Select **SAML**
   b. Edit **Basic Configuration**
      i. Add an **Entity Id** `lighthouse-azure_ad` and set it as default.
      ii. In **Reply URL (Assertion Consumer Service URL)** add the SSO URL for each address of each lighthouse that you want to be able to sign in on. i.e. IP addresses and DNS address for both your primary and secondary lighthouses.

```
https://{primary lighthouse address}/api/v3.7/sessions/saml/sso/azure_ad
https://{primary lighthouse IP address}/api/v3.7/sessions/saml/sso/azure_ad
https://{secondary lighthouse address}/api/v3.7/sessions/saml/sso/azure_ad
https://{secondary lighthouse IP address}/api/v3.7/sessions/saml/sso/azure_ad
```

iii. Set **Sign on URL** to `https://{main lighthouse address}/api/v3.7/sessions/saml/sp_init/azure_ad`

iv. Click **Save**

c. Edit **Attributes & Claims**

i. Remove the default claims from **Additional claims**.

ii. Click **Add new claim** and Enter:

1. Name: `LH_Groups`
2. Source Attributes: `user.assignedroles`

IdP Metadata

1. Go to the **Azure Active Directory**
2. Go to **Enterprise applications** and open your application
3. Go to **Single sign-on**
4. In **3. SAML Signing Certificate** find and download `Federation Metadata XML`

Configure Lighthouse

1. Copy the federation metadata XML to your primary lighthouse.
2. Using `saml-idp-metadata` on your primary lighthouse, configure each of your lighthouses to use your IdP
   e.g. `saml-idp-metadata -p {root password} create -m /path/to/metadata.xml -P azure_ad -n "My Azure display name" -l {LH id number}`

App Roles Setup

Note: If you do not already have your own Usergroups setup in lighthouse:

1. Login to Lighthouse as a local user (or any non-SAML user) i.e. root

   Note: After this initial setup, you will be able to login as a SAML user.

2. Create the Usergroups with the Roles and permission that you desire. See "8.2 Creating new groups and roles" in the Lighthouse User manual.

See Add app roles and get them from a token - Microsoft identity platform for up to date documentation on how to create and assign App Roles.

1. Go to **Azure Active Directory**
2. Go to **App registrations**
3. Open your app (Use the **All Applications** tab to see Enterprise apps)
4. Go to **App Roles**
5. Click **Create App Role**

    a. Set the **value** to match your usergroup on lighthouse.
    b. Set **Allowed member types** to `Both (Users/Groups + Applications)`.
    c. Set the other fields as you desire.

6. Go to **Azure Active Directory**
7. Go to **Enterprise applications**
8. Open your App
9. Go to **Users and groups**
10. Click **Add user/group**
11. Select a user and one of your App roles then click **Assign**.

The assigned users are now able to login to lighthouse with the permission levels which that App Role/Lighthouse group grants them.

8.11.5 Limitations

IdP metadata certificate expiry

The IdP metadata XML file that you export to configure lighthouse contains a certificate that is used to authenticate that SAML response came from your IdP. Different IdPs have different expiry periods for these certificates, consult your IdP's documentation to find their expiry period. When your IdP's certificate expires you will need to regenerate it then re-export your IdP metadata and update your lighthouse configurations. If your IdP supports sending expiry notifications to your admin, we recommend you enable these notifications.

User Permissions Changes

When you change the permissions assigned to a lighthouse user in your IdP (via **`LH_Groups`** SAML attribute), the changes will not take effect until the user logs out and back into lighthouse.

If you need to quickly restrict a user's access, consider altering the permissions of or deleting that user's usergroups on lighthouse, see lighthouse user manual "8.3 Modifying existing groups". You can also set a low **`Web Session Timeout`**. See Lighthouse user manual "5.9 Examine or modify Lighthouse Session Settings".

SAML SSO Usergroups

The **`LH_Groups`** attribute can be used to place SSO users in any lighthouse usergroup except lighthouse's default **`admin`** group. You can allow users to login with admin privileges by simply creating another usergroup in lighthouse with the admin role and assigning the matching role/group in your IdP to the user (i.e. populate **`LH_Groups`** to include its value).

SAML SSO Users

SAML Users can only be managed in your IdP and will not appear under lighthouse User Management.

8.12 Configuring AAA

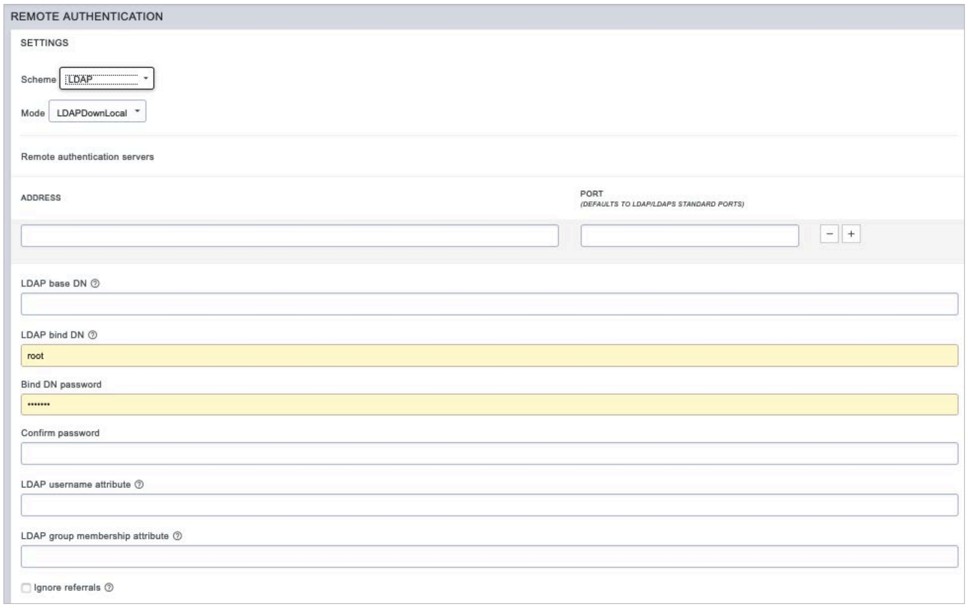
Lighthouse supports three AAA systems:

- 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**.

8.12.1 LDAP Configuration



REMOTE AUTHENTICATION

SETTINGS

Scheme: LDAP

Mode: LDAPDownLocal

Remote authentication servers

ADDRESS	PORT (DEFAULTS TO LDAP/LDAPS STANDARD PORTS)

LDAP base DN

LDAP bind DN: root

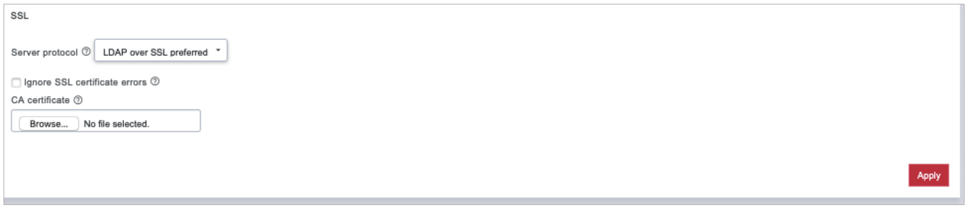
Bind DN password: •••••••

Confirm password

LDAP username attribute

LDAP group membership attribute

□ Ignore referrals



SSL

Server protocol: LDAP over SSL preferred

□ Ignore SSL certificate errors

CA certificate: Browse... No file selected.

1. Select ***LDAP*** from the ***Scheme*** drop-down menu.
2. Choose the desired **Mode** from the drop-down menu.
   - LDAPDownLocal
   - LDAP: Default behavior
   - LDAP/Local
   - Local/LDAP

NOTE: See the *Glossary* for more information about these modes.

3. Add the ***Address*** and optionally the ***Port*** of the LDAP server to query.
4. 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**

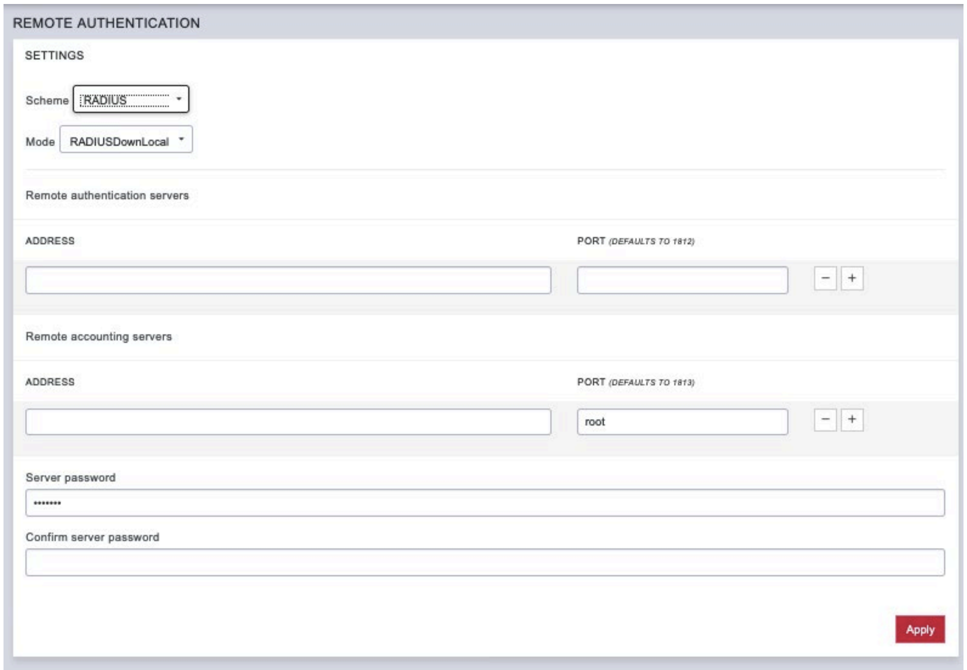
5. Add the **LDAP *Bind DN***. This is the distinguished name of a user with privileges on the LDAP system to perform the lookups required for retrieving the username of the users, and a list of the groups they are members of.
6. Add and confirm a password for the binding user.
7. Add the **LDAP *username attribute***. This depends on the underlying LDAP system. Use **sAMAccountName** for Active Directory systems, and **uid** for OpenLDAP based systems.
8. Add the **LDAP *group membership attribute***. This is only needed for Active Directory and is generally **memberOf**.
9. 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.
10. Under the **SSL section**, choose the desired **Server protoco**l.
    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
11. If desired, check **Ignore SSL certificate errors** to ignore any SSL certificate errors.
12. **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.
13. Install the **CA certificate** by clicking the **Browse…** button and locating the appropriate file.
14. Click **Apply**.

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

8.12.2 RADIUS configuration

To configure RADIUS:

1. Select **SETTINGS > USER MANAGEMENT > Remote Authentication**.



Scheme: RADIUS
Mode: RADIUSDownLocal
PORT (DEFAULTS TO 1813): root
Server password: •••••••

2. In the **Settings** section, select ***RADIUS*** from the ***Scheme*** drop-down menu.
3. Choose the desired **Mode** from the drop-down menu.

    - RADIUSDownLocal
    - Radius
    - RADIUS/Local
    - Local/RADIUS

    NOTE: See the *Glossary* for more information about these modes.

4. Add the ***Address*** and optionally the ***Port*** of the RADIUS authentication server to query.
5. Add the ***Address*** and optionally the ***Port*** of the RADIUS accounting server to send accounting information to.
6. 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:

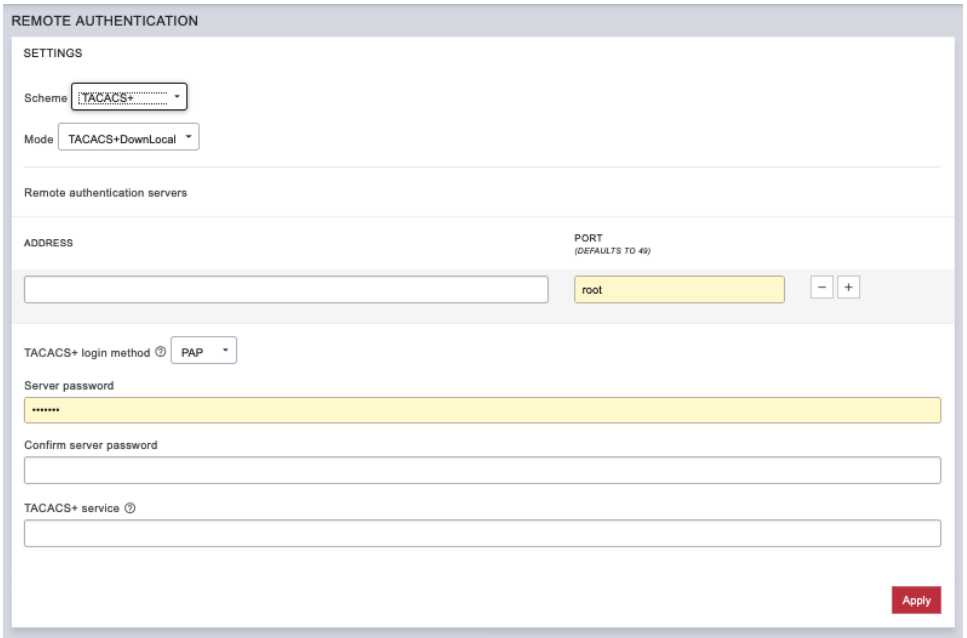
```
operator1 Auth-Type := System
      Framed-Filter-ID = ":group_name=west_coast_admin,east_coast_user:"
```

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

8.12.3 TACACS+ configuration

To configure TACACS+:

1. Select **SETTINGS > USER MANAGEMENT > Remote Authentication**.



REMOTE AUTHENTICATION

SETTINGS

Scheme: TACACS+

Mode: TACACS+DownLocal

Remote authentication servers

ADDRESS	PORT (DEFAULTS TO 49)
	root

TACACS+ login method: PAP

Server password: •••••••

Confirm server password:

TACACS+ service:

2. Select ***TACACS+*** from the ***Scheme*** drop-down menu.
3. Choose the desired **Mode** from the drop-down menu.

    - TACACSDownLocal:
    - TACACS: Default behavior
    - TACACS/Local
    - Local/TACACS

    NOTE: See the *Glossary* for more information about these modes.

4. Add the ***Address*** and optionally the ***Port*** of the TACACS+ authentication server to query.
5. Select the ***Login Method***. ***PAP*** is the default method. However, if the server uses DES-encrypted passwords, select ***Login***.
6. Add the ***Server password***, also known as the TACACS+ Secret.
7. 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
          }
}
```

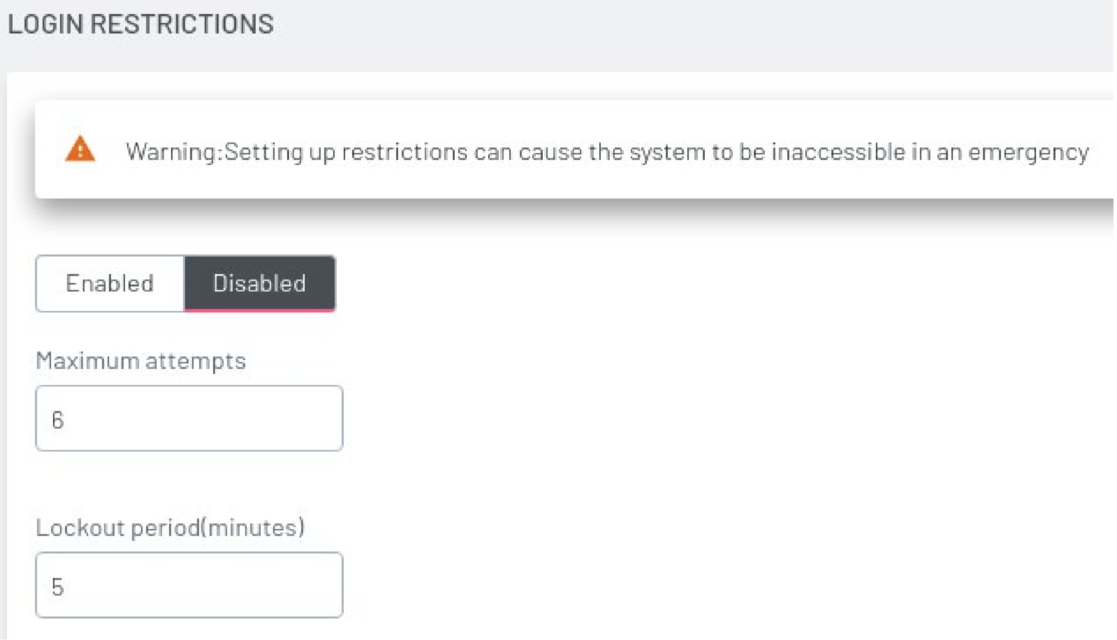
To do this with Cisco ACS, see *Setting up permissions with Cisco ACS 5 and TACACS+* on the Opengear Help Desk.

8.13 Setting Login Restrictions

Login restrictions can be applied by administrator users to prevent unauthorized login attempts via the UI and REST API.

NOTE: This does not affect SSH or Console logins.

Select **SETTINGS > USER MANAGEMENT > Login Restrictions.**



Warning:Setting up restrictions can cause the system to be inaccessible in an emergency

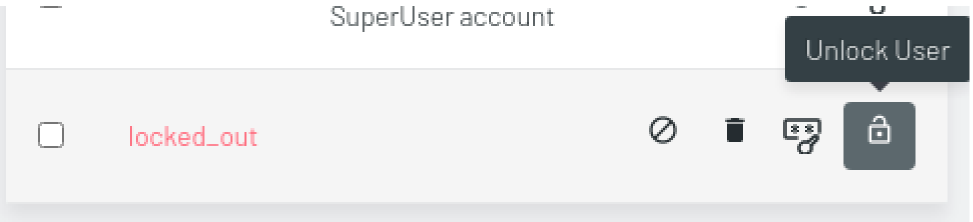
Maximum attempts: 6

Lockout period(minutes): 5

Enter a value for either or both of the following:

- **Maximum attempts** – choose a number of attempts a user can enter an incorrect password before being locked out.
- **Lockout period** – enter a number of minutes until a user can try to login again after reaching maximum incorrect login attempts.

After a user has been locked out, an administrator can unlock the account. To do this, go to **SETTINGS > USER MANAGEMENT > Local Users**. Click the **Unlock** button associated with the locked out user's row in the User table.


SuperUser account
Unlock User
□
locked_out

9. Notifications

Lighthouse has a MessageBus functionality that allows you to connect Lighthouse to third party systems for notifications, ticket creation, and alerts.

Lighthouse allows integrations such as:

If a node goes into failover mode

- Create a trouble ticket with your companies Zendesk trouble ticketing system
- Trigger an alert to the companies on call scheduled via PagerDuty

If Lighthouse license is 45 days from expiry

- Send an email to purchasing
- Create a calendar notification

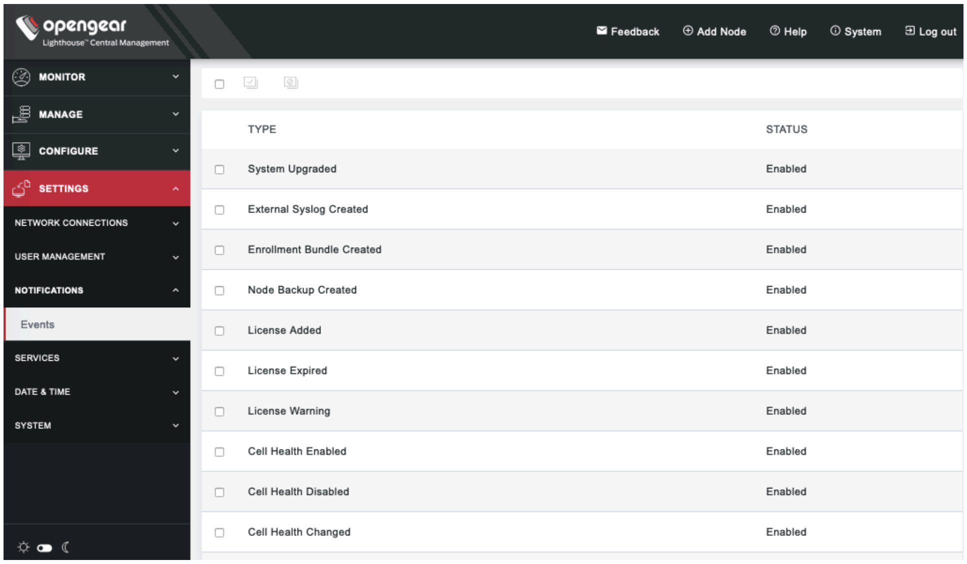
Zapier is the first integration that is available within Lighthouse. In order for events to be fired from Lighthouse and received by Zapier:

- The Lighthouse should have a valid enterprise license.
- The event should be enabled on Lighthouse.
- There must be a Zap created to handle the event.

Once these requirements are met, a user will be able to receive Lighthouse event messages via third-party apps integrated with Zapier.

To enable events, go to **SETTINGS > NOTIFICATIONS > Events**. Click the checkbox next to the desired event to select it and click the **Enable** button at the top of the table to enable it. To enable all events, click the checkbox on the top left of the screen and click the **Enable** button.

To disable an event, select the checkbox next to it, then the **Disable** button at the top of the screen.



	TYPE	STATUS
□	System Upgraded	Enabled
□	External Syslog Created	Enabled
□	Enrollment Bundle Created	Enabled
□	Node Backup Created	Enabled
□	License Added	Enabled
□	License Expired	Enabled
□	License Warning	Enabled
□	Cell Health Enabled	Enabled
□	Cell Health Disabled	Enabled
□	Cell Health Changed	Enabled

Known Issues/Limitations/Warnings

- Secondary Lighthouses will not be able to send events to messagebus.
- Unknown behavior and errors can be expected in case the user tries to integrate a secondary Lighthouse with Zapier and creates zaps.
- Modifying the system time will result in unexpected behavior with events being triggered.
- Zapier does not support IPv6.

Here are the specific events currently supported.

- **System Upgraded**: Event fired when Lighthouse is upgraded
- **External Syslog Created:** Event fired by Lighthouse when a user creates external syslog server.
- **Enrolment Bundle Created**: Event fired when a user creates a new enrolment bundle.
- **Node Backup Created:** Event fired when a node backup is created by the user. This event will only fire if there are nodes enrolled and a backup is successfully created.
- **License Added:** Event fired when a user imports a new license.
- **License Expired:** Event fired when a license expires.
- **License Warnings (60/45/30/15)**: Event fired warning user that their Lighthouse license will expire in 60 → 45 → 30 → 15 days.
- **Cell Health Enabled:** Event fired when user enables cell health on Lighthouse.
- **Cell Health Disabled**: Event fired when user disables cell health on Lighthouse.
- **Cell Health Changed**: Event fired when a node’s cell health changes in the order Good > Moderate, Moderate > Bad, Good > Bad and all in reverse.
- **Node Enrolled:** Event fired when a node enrolls into Lighthouse.
- **Node Unenrolled**: Event fired when a node unenrolls from Lighthouse.
- **Node Failed Over**: Event fired when a node fails over to the designated failover interface.
- **Node Failed Back**: Event fired when a node fails back to primary interface.
- **Node Port Logging Enabled**: Event fired when node port logging is enabled.
- **Node Port Logging Disabled**: Event fired when node port logging is disabled.
- **User Created**: Event fired when a user is created.
- **User Deleted**: Event fired when user is deleted.

- **Group Created**: Event fired when group is created.
- **Group Deleted**: Event fired when group is deleted.
- **Role Created**: Event fired when role is created.
- **Role Deleted**: Event fired when role is deleted.

10. 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.

10.1 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.

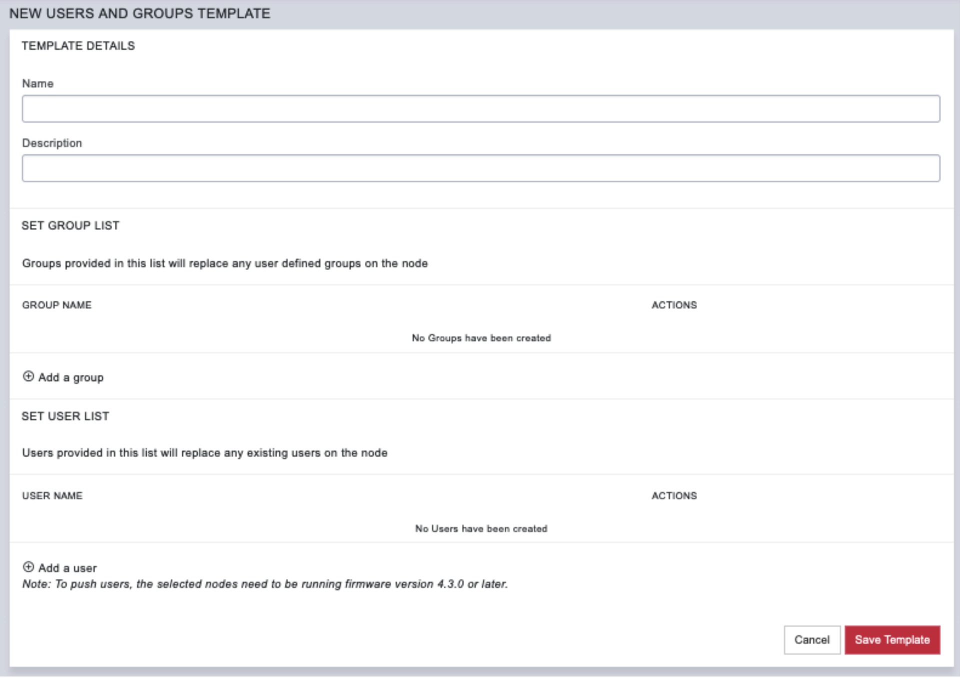
Each template contains a list of user-defined groups and/or individual users. Each group has a defined role which determines what privileges group members have. User roles are defined by the groups they are a member of.

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** page loads.



NEW USERS AND GROUPS TEMPLATE

TEMPLATE DETAILS

Name

Description

SET GROUP LIST

Groups provided in this list will replace any user defined groups on the node

GROUP NAME	ACTIONS
No Groups have been created	

⊕ Add a group

SET USER LIST

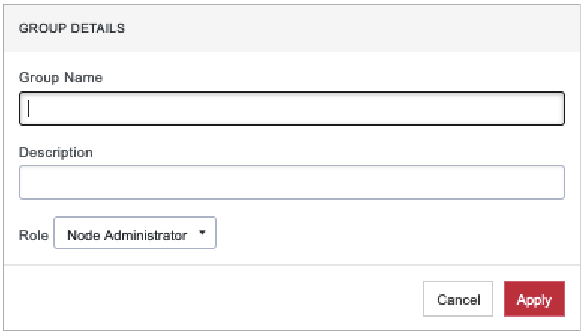
Users provided in this list will replace any existing users on the node

USER NAME	ACTIONS
No Users have been created	

⊕ Add a user

Note: To push users, the selected nodes need to be running firmware version 4.3.0 or later.

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



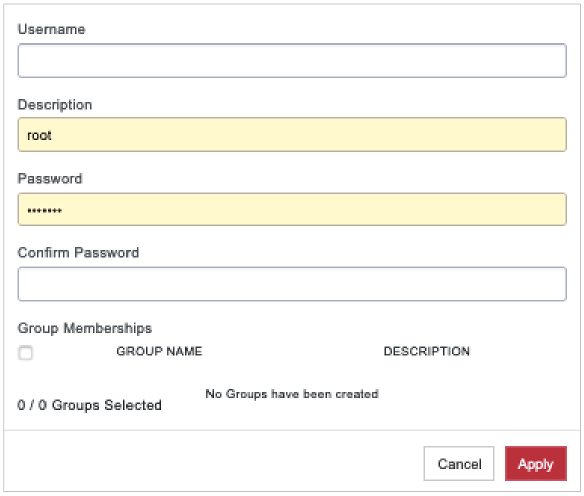
GROUP DETAILS

Group Name

Description

Role: Node Administrator ▾

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 **+ Add a user** button in the **Set User List** section to add new users. The **User Details** dialog loads.



Username

Description: root

Password: •••••••

Confirm Password

Group Memberships

□	GROUP NAME	DESCRIPTION
	No Groups have been created	

0 / 0 Groups Selected

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**.

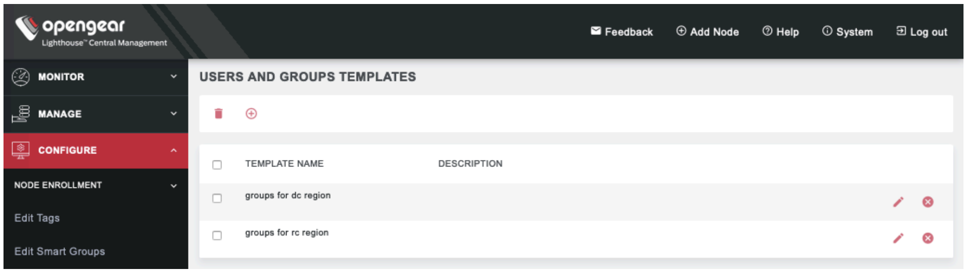
NOTE: When a **users and groups template** is pushed to a node, all custom groups on that node are replaced by groups defined in the template. If no users are in the new template, existing users will remain on the node. To push users, the selected nodes need to be running firmware version 4.3.0 or later.

10.2 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:

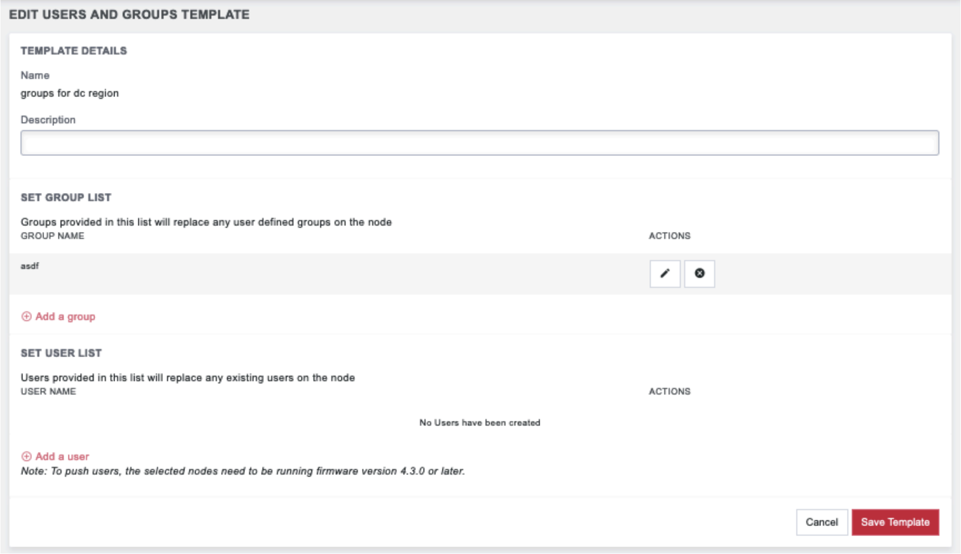
1. Select **CONFIGURE > CONFIGURATION TEMPLATING > Users and Groups Templates**.



USERS AND GROUPS TEMPLATES

TEMPLATE NAME	DESCRIPTION
groups for dc region	
groups for rc region	

2. Click **Edit button** next to the template to be modified. The **Edit Users and Groups Template** dialog appears.



EDIT USERS AND GROUPS TEMPLATE

TEMPLATE DETAILS

Name
groups for dc region

Description

SET GROUP LIST

Groups provided in this list will replace any user defined groups on the node

GROUP NAME	ACTIONS
asdf	

SET USER LIST

Users provided in this list will replace any existing users on the node

USER NAME	ACTIONS

No Users have been created

Note: To push users, the selected nodes need to be running firmware version 4.3.0 or later.

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**.

10.3 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.

10.4 Deleting users and groups templates

To delete a template:

1. Select **CONFIGURE > CONFIGURATION TEMPLATING > Users and Groups Templates**.
2. Click the **Delete** button next to the template to be removed. The ***Confirmation*** alert box appears.
3. Click **Yes** in the **Confirmation** dialog. The template is deleted.

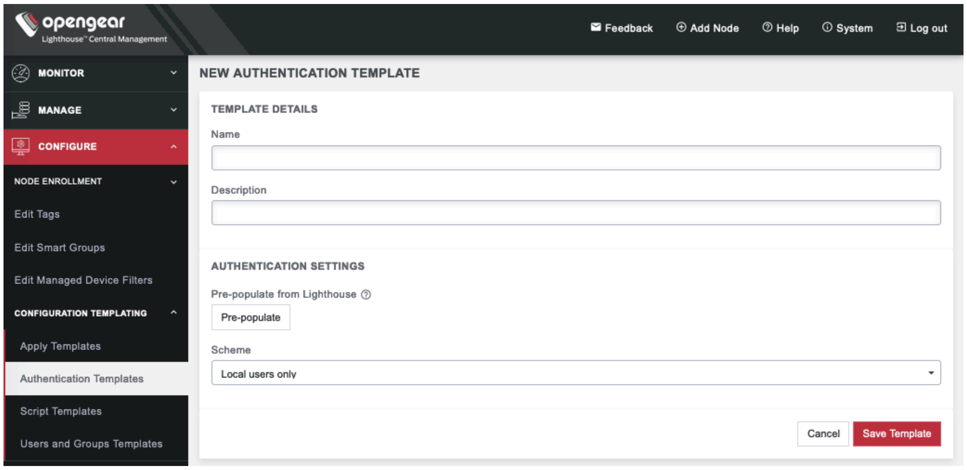
10.5 Creating new authentication templates

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** page loads.



NEW AUTHENTICATION TEMPLATE

TEMPLATE DETAILS

Name

Description

AUTHENTICATION SETTINGS

Pre-populate from Lighthouse

Scheme: Local users only

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 8.10 Configuring AAA 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.

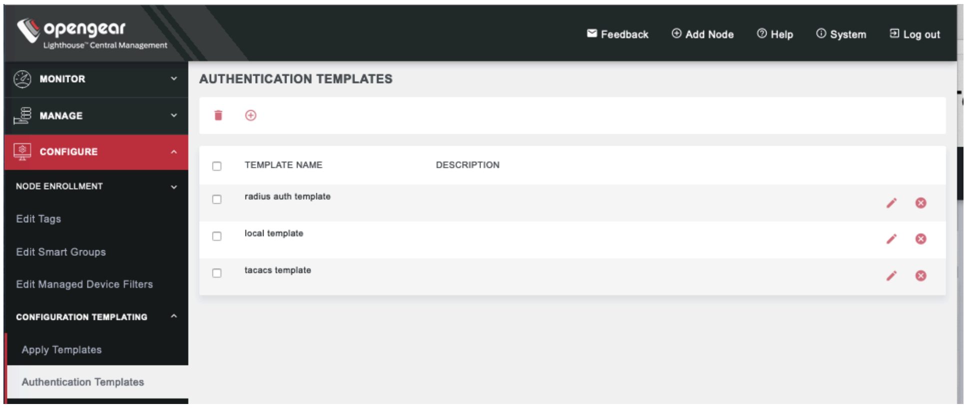
NOTE: The authentication templates do not support the full list of settings that the Opengear console servers support. However, templates can be applied, and then additional settings configured manually.

10.6 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:

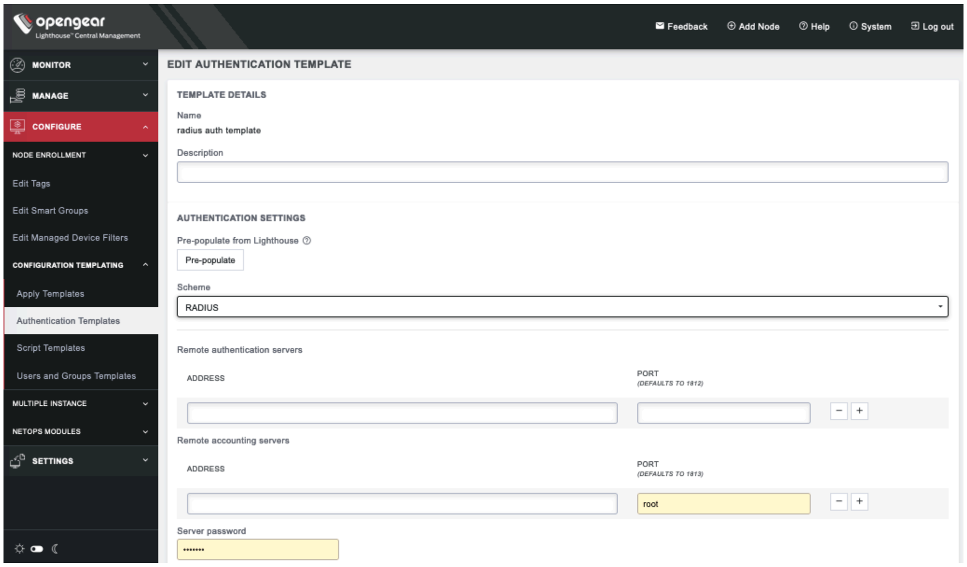
1. Select **CONFIGURE > CONFIGURATION TEMPLATING > Authentication Templates**.



AUTHENTICATION TEMPLATES

□	TEMPLATE NAME	DESCRIPTION
□	radius auth template	
□	local template	
□	tacacs template	

2. Click **Edit** next to the template to be modified. The **Edit Authentication Template** dialog appears.



EDIT AUTHENTICATION TEMPLATE

TEMPLATE DETAILS

Name
radius auth template

Description

AUTHENTICATION SETTINGS

Pre-populate from Lighthouse

Scheme
RADIUS

Remote authentication servers

ADDRESS	PORT (DEFAULTS TO 1812)

Remote accounting servers

ADDRESS	PORT (DEFAULTS TO 1813)
	root

Server password
•••••••

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

10.7 Deleting authentication templates

To delete an authentication template:

1. Select **CONFIGURE > CONFIGURATION TEMPLATING > Authentication Templates**.
2. Click **Delete** next to the template to be removed. The ***Confirmation*** alert box appears.
3. Click **Yes** in the **Confirmation** dialog. The authentication template is deleted.

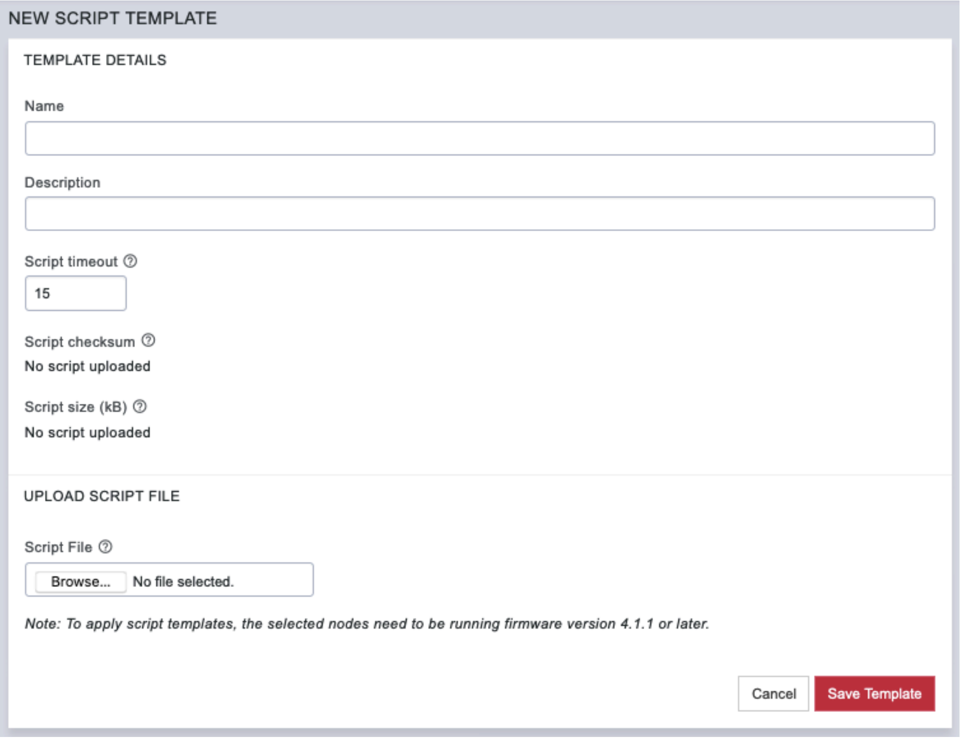
10.8 Creating new script templates

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.



NEW SCRIPT TEMPLATE

TEMPLATE DETAILS

Name

Description

Script timeout ⓘ
15

Script checksum ⓘ
No script uploaded

Script size (kB) ⓘ
No script uploaded

UPLOAD SCRIPT FILE

Script File ⓘ
No file selected.

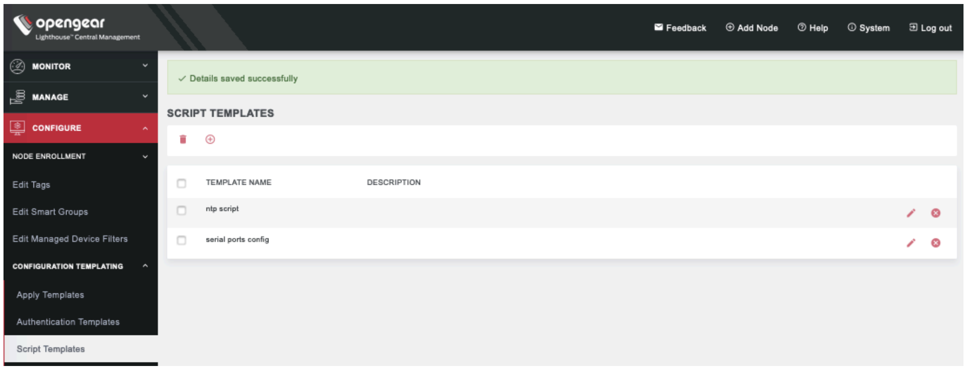
Note: To apply script templates, the selected nodes need to be running firmware version 4.1.1 or later.

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.

10.9 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**.

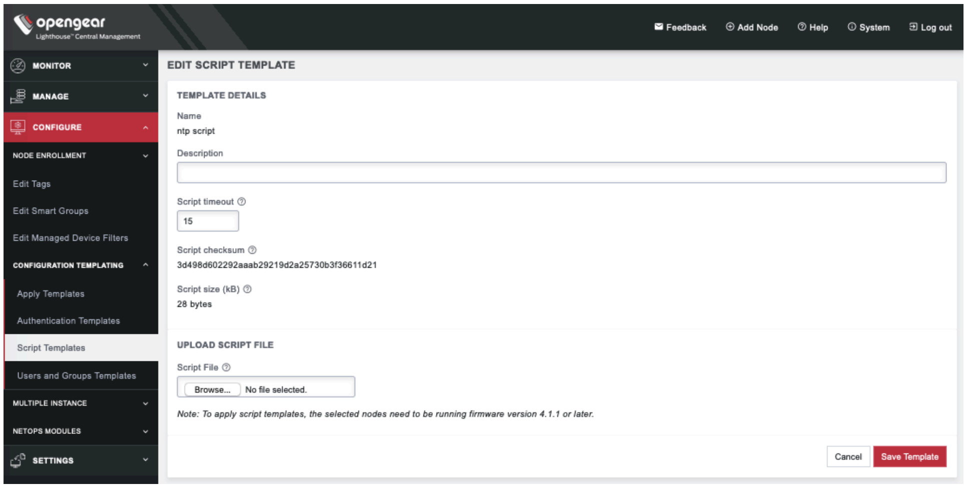


✓ Details saved successfully

SCRIPT TEMPLATES

☐	TEMPLATE NAME	DESCRIPTION
☐	ntp script	
☐	serial ports config	

2. Click **Edit** next to the template to be modified. The **Edit Script Template** dialog appears.



EDIT SCRIPT TEMPLATE

TEMPLATE DETAILS

Name
ntp script

Description

Script timeout
15

Script checksum
3d498d602292aaab29219d2a25730b3f36611d21

Script size (kB)
28 bytes

UPLOAD SCRIPT FILE

Script File
No file selected.

Note: To apply script templates, the selected nodes need to be running firmware version 4.1.1 or later.

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

10.10 Deleting script templates

To delete a script template completely:

1. Select **CONFIGURE > CONFIGURATION TEMPLATING > Script Templates**.
2. Click **Delete** next to the template to be removed. The ***Confirmation*** alert box appears.
3. Click **Yes** in the **Confirmation** dialog. The script template is deleted.

10.11 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.

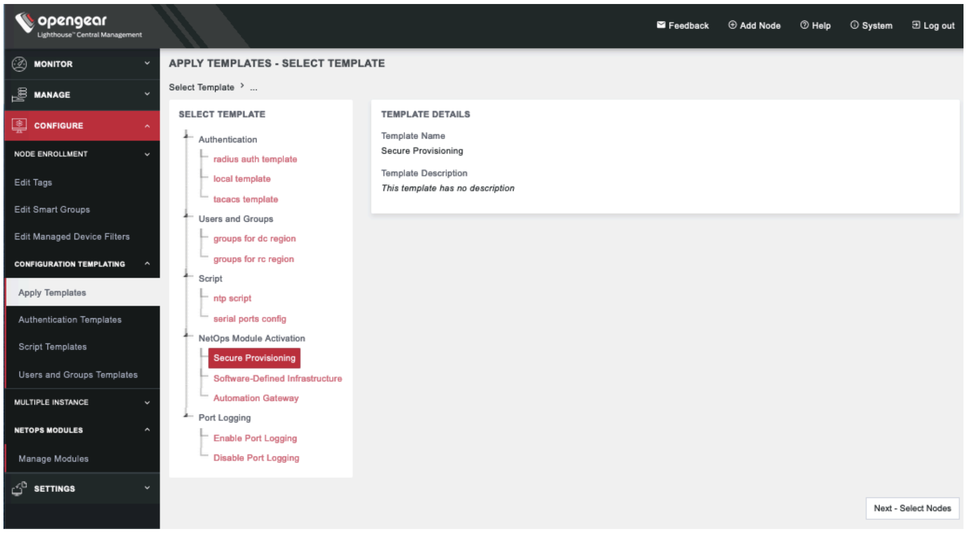
Users with Node Administrator privileges (i.e. users with the Node Administrator role or users who are members of groups with the Node Administrator role) can access **CONFIGURE > CONFIGURATION TEMPLATING > Apply Templates** and execute templates affecting nodes in Smart Groups linked to their role.

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**.



APPLY TEMPLATES - SELECT TEMPLATE

Select Template › ...

SELECT TEMPLATE

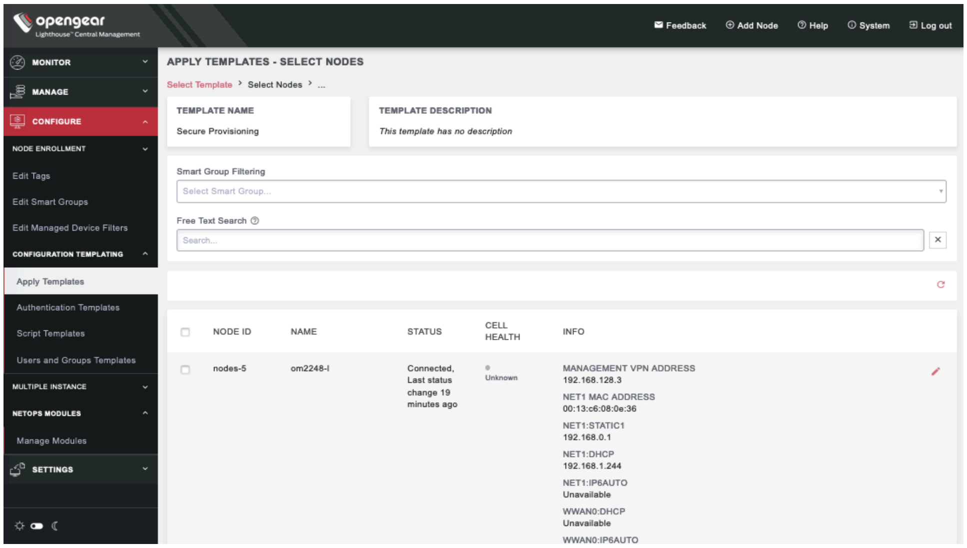
- Authentication
  - radius auth template
  - local template
  - tacacs template
- Users and Groups
  - groups for dc region
  - groups for rc region
- Script
  - ntp script
  - serial ports config
- NetOps Module Activation
  - Secure Provisioning
  - Software-Defined Infrastructure
  - Automation Gateway
- Port Logging
  - Enable Port Logging
  - Disable Port Logging

TEMPLATE DETAILS

Template Name
Secure Provisioning

Template Description
This template has no description

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.



APPLY TEMPLATES - SELECT NODES

Select Template › Select Nodes › ...

TEMPLATE NAME
Secure Provisioning

TEMPLATE DESCRIPTION
This template has no description

Smart Group Filtering
Select Smart Group...

Free Text Search
Search...

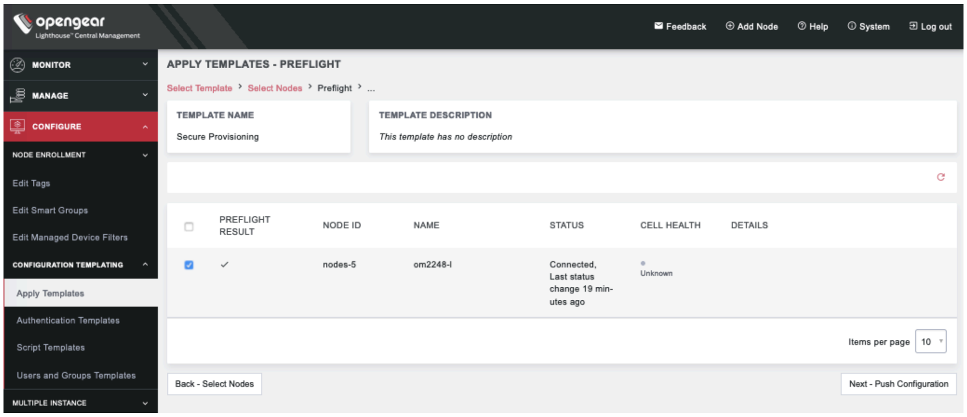
□	NODE ID	NAME	STATUS	CELL HEALTH	INFO
□	nodes-5	om2248-l	Connected, Last status change 19 minutes ago	Unknown	MANAGEMENT VPN ADDRESS 192.168.128.3 NET1 MAC ADDRESS 00:13:c6:08:0e:36 NET1:STATIC1 192.168.0.1 NET1:DHCP 192.168.1.244 NET1:IP6AUTO Unavailable WWAN0:DHCP Unavailable WWAN0:IP6AUTO

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.

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

5. Scroll to the bottom of the page and 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.



APPLY TEMPLATES - PREFLIGHT

TEMPLATE NAME
Secure Provisioning

TEMPLATE DESCRIPTION
This template has no description

☐	PREFLIGHT RESULT	NODE ID	NAME	STATUS	CELL HEALTH	DETAILS
☑	✓	nodes-5	om2248-l	Connected, Last status change 19 minutes ago	Unknown	

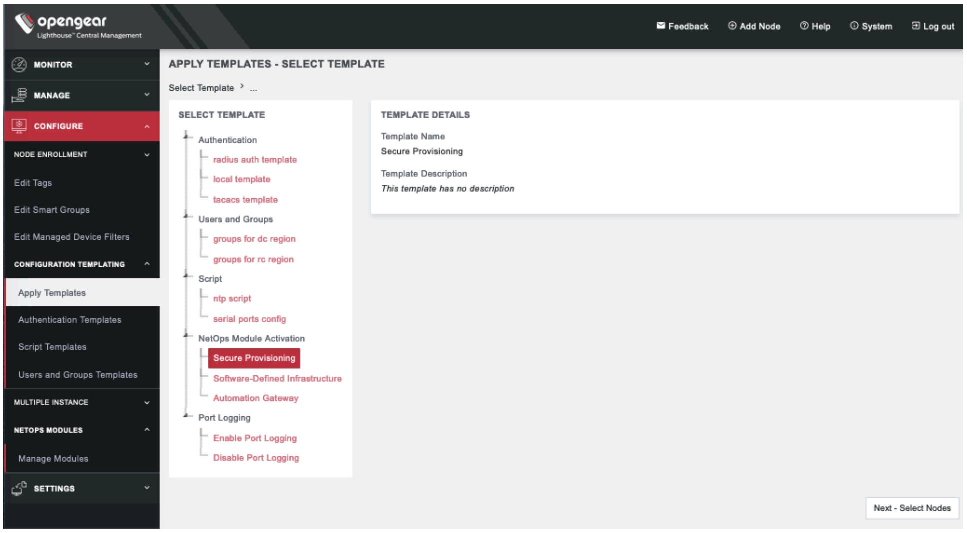
6. Select desired nodes for template execution and click **Next — Push Configuration**. The **Configuration Status** stage loads. This stage requires manual refresh to retrieve updated **Push Result** and **Details**.

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

10.12 Manually Activate Secure Provisioning or Software Defined Infrastructure via Template

Users assigned to the ***Lighthouse Administrator*** role can manually apply the **Secure Provisioning NetOps Module** or **Software Defined Infrastructure** to desired OPERATIONS MANAGER nodes.

1. As a Lighthouse administrator, choose **CONFIGURE > CONFIGURATION TEMPLATING > Apply Templates**
2. Click **Secure Provisioning** or **Software Defined Infrastructure** under **NetOps Module Activation**.



APPLY TEMPLATES - SELECT TEMPLATE

TEMPLATE DETAILS

Template Name
Secure Provisioning

Template Description
This template has no description

3. Click **Next – Select Nodes**
4. Choose the desired OPERATIONS MANAGER 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**.

11. Multiple Instance

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

11.1 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.

NOTE: The certificate will be used on all instances. For the multiple instance feature, we recommend using a wildcard certificate.

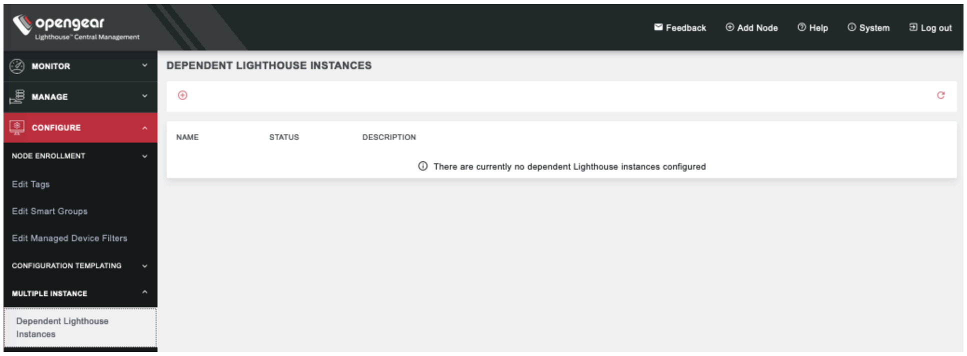
If a multiple instance license is not installed:

- 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.

11.2 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.**

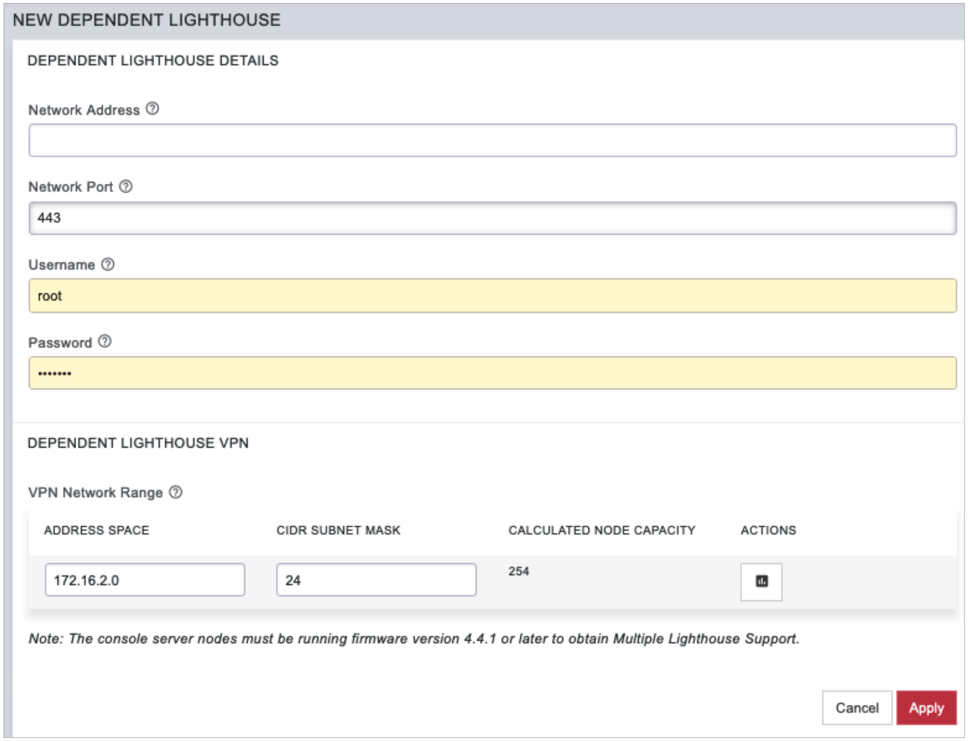


DEPENDENT LIGHTHOUSE INSTANCES

NAME	STATUS	DESCRIPTION

There are currently no dependent Lighthouse instances configured

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.



NEW DEPENDENT LIGHTHOUSE

DEPENDENT LIGHTHOUSE DETAILS

Network Address	
Network Port	443
Username	root
Password	•••••••

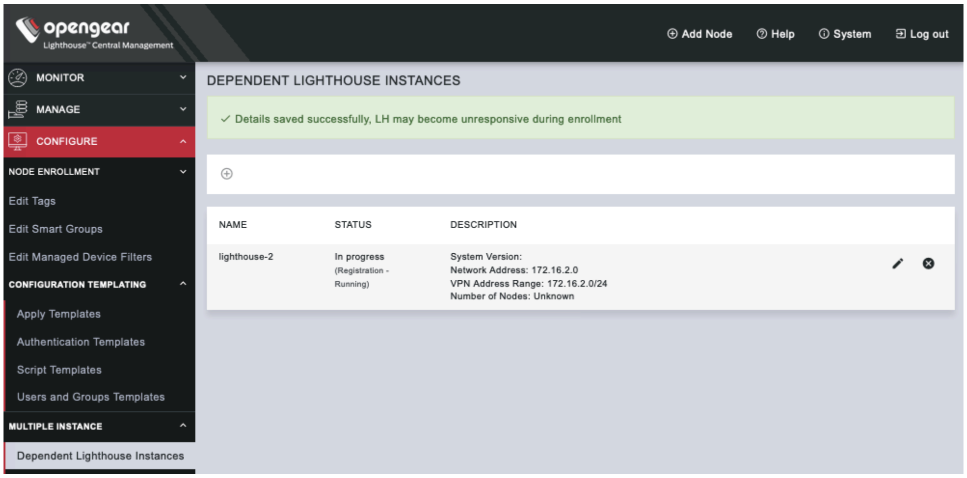
DEPENDENT LIGHTHOUSE VPN

VPN Network Range

ADDRESS SPACE	CIDR SUBNET MASK	CALCULATED NODE CAPACITY	ACTIONS
172.16.2.0	24	254	

Note: The console server nodes must be running firmware version 4.4.1 or later to obtain Multiple Lighthouse Support.

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

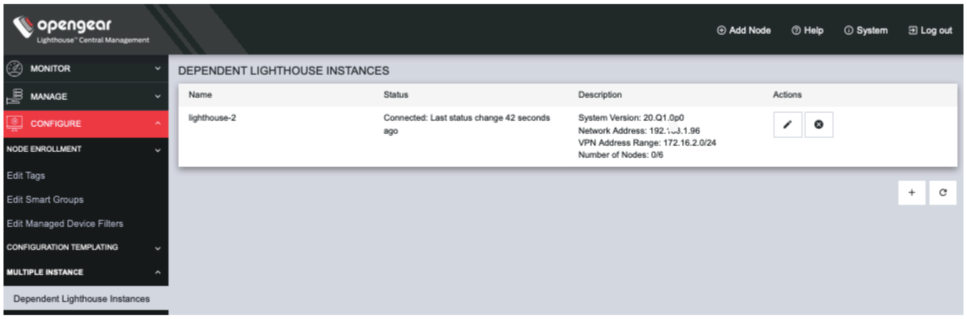


DEPENDENT LIGHTHOUSE INSTANCES

✓ Details saved successfully, LH may become unresponsive during enrollment

NAME	STATUS	DESCRIPTION
lighthouse-2	In progress (Registration - Running)	System Version: Network Address: 172.16.2.0 VPN Address Range: 172.16.2.0/24 Number of Nodes: Unknown

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.



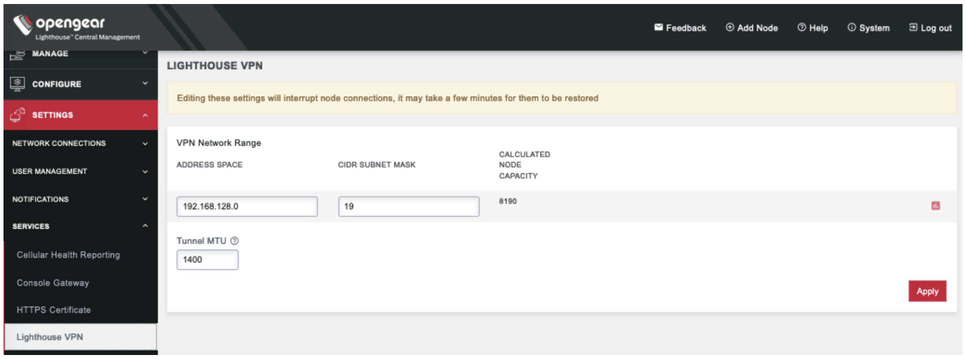
DEPENDENT LIGHTHOUSE INSTANCES

Name	Status	Description	Actions
lighthouse-2	Connected: Last status change 42 seconds ago	System Version: 20.Q1.0p0 Network Address: 192.1[illegible]3.1.96 VPN Address Range: 172.16.2.0/24 Number of Nodes: 0/6	

11.3 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. **Calculated Node capacity** displays the addressable nodes based on the network address and CIDR mask.



LIGHTHOUSE VPN

Editing these settings will interrupt node connections, it may take a few minutes for them to be restored

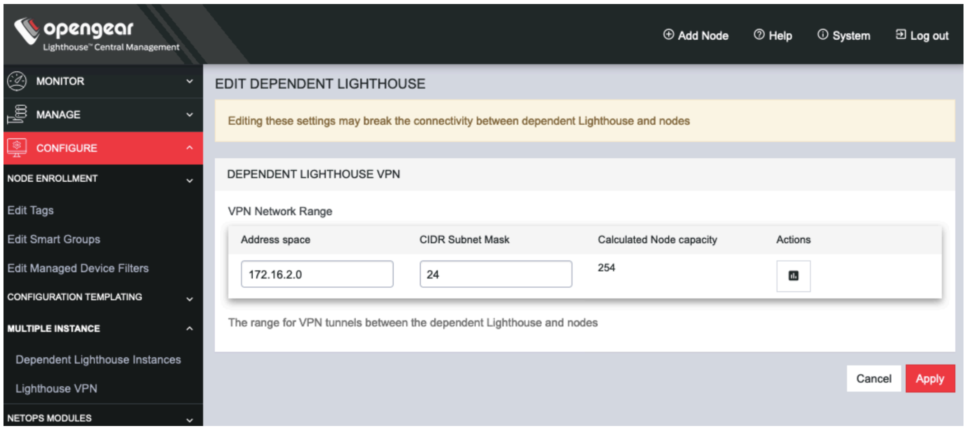
VPN Network Range

ADDRESS SPACE	CIDR SUBNET MASK	CALCULATED NODE CAPACITY
192.168.128.0	19	8190

Tunnel MTU: 1400

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**.



EDIT DEPENDENT LIGHTHOUSE

Editing these settings may break the connectivity between dependent Lighthouse and nodes

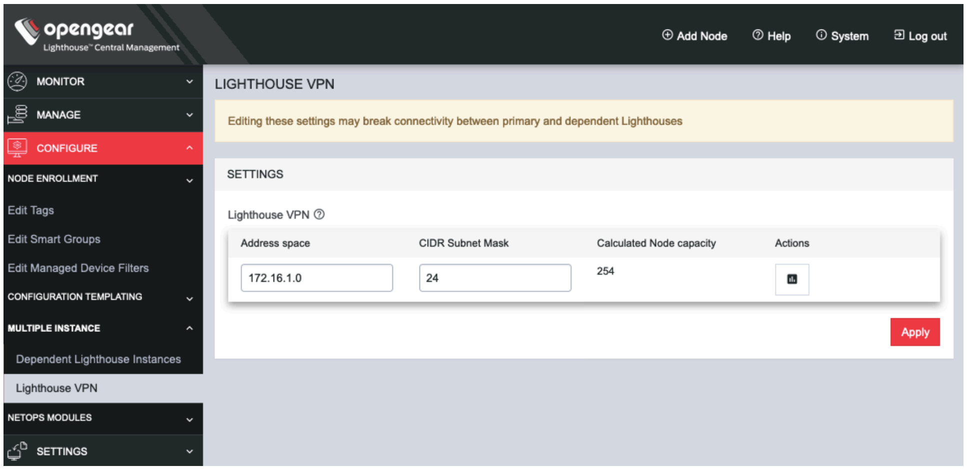
DEPENDENT LIGHTHOUSE VPN

VPN Network Range

Address space	CIDR Subnet Mask	Calculated Node capacity	Actions
172.16.2.0	24	254	

The range for VPN tunnels between the dependent Lighthouse and nodes

The subnet between the primary Lighthouse and dependent Lighthouse instance can be modified on the primary Lighthouse under **CONFIGURE > MULTIPLE INSTANCE > Lighthouse VPN**



LIGHTHOUSE VPN

Editing these settings may break connectivity between primary and dependent Lighthouses

SETTINGS

Lighthouse VPN

Address space	CIDR Subnet Mask	Calculated Node capacity	Actions
172.16.1.0	24	254	

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:

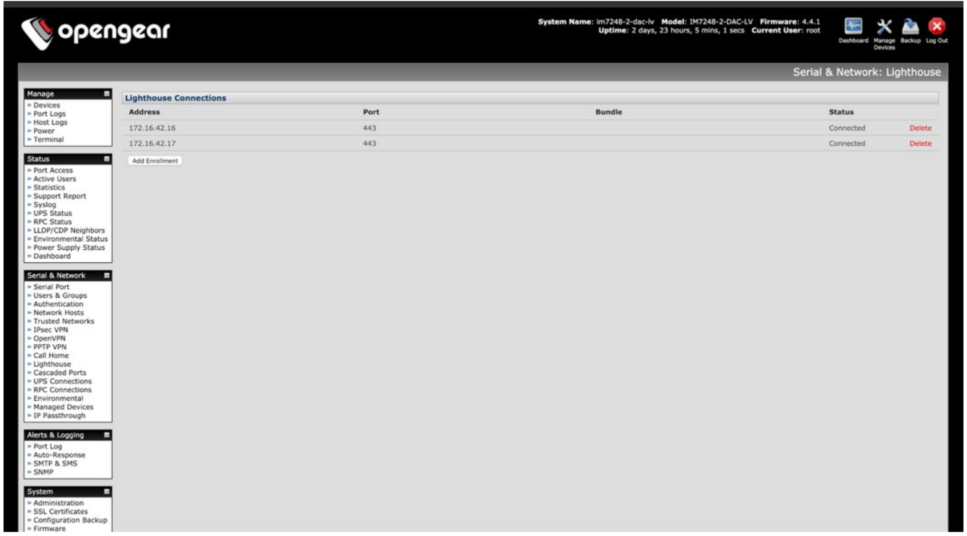
```
ogconfig-cli
set lighthouse_configurations[1].hostname new_name
push
```

11.4 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.



System Name: im7248-2-dac-lv **Model:** IM7248-2-DAC-LV **Firmware:** 4.4.1
Uptime: 2 days, 23 hours, 5 mins, 1 secs **Current User:** root

Serial & Network: Lighthouse

Lighthouse Connections

Address	Port	Bundle	Status	
172.16.42.16	443		Connected	Delete
172.16.42.17	443		Connected	Delete

Add Enrollment

11.5 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.

11.6 Upgrading a multiple instance Lighthouse

To upgrade a Multiple Instance Lighthouse:

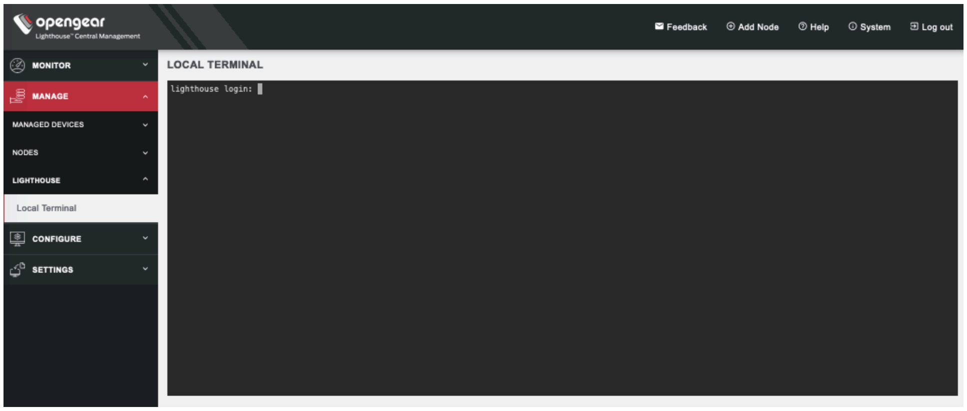
When the primary Lighthouse is updated, any secondary Lighthouses will be updated in a rolling fashion after the primary has successfully booted. If any Lighthouse fails to successfully update along the way, the update will stop.

Please see 14.3 Upgrading Dependent Multiple Instances of Lighthouse for additional information.

12. Command line tools

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

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



```
lighthouse login:
```

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
```

12.1 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
```

12.2 node-upgrade

node-upgrade is a tool for running firmware upgrades on multiple managed console servers with a single command and returns the results in tabular form to `stdout`.

node-upgrade accepts the following arguments:

Short Argument	Long Argument	Description
`-h`	`--help`	Display usage information and exit
`-q`	`--quiet`	Suppress log messages
`-V`	`--verbose`	Display logs generated while upgrading
`-l`	`--list-nodes`	Display nodes and their upgradeable paths without executing upgrade
`-D`	`--debug`	Display detailed log messages, implies `--verbose`
`-I`	`--ignore-version`	Ignore firmware version warnings for upgrade
`-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
`-A`	`--all`	Select all available nodes
`-p`	`--product=<family>`	Select node by product family
`-g`	`--smartgroup=<name>`	Select nodes by smartgroup filter
`-f`	`--firmware-dir=<directory>`	The directory of the firmware files(s)
`-F`	`--firmware-file=<path>`	The firmware image to use for upgrade
`-v`	`--version=<version>`	The firmware version to upgrade to

12.2.1 An example `node-upgrade` run

The following is an example node-upgrade command. It sets `/mnt/data/nvram/latest-firmware` 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 4.11.0.

```
# node-upgrade --all --firmware-dir /mnt/data/nvram/latest-firmware/ --version 4.11.0

NODE (UUID)           MODEL       FAMILY     ADDRESS         VERSION   RESULT
----------------------------------------------------------------------------------------
cm7116-2 (nodes-4)    CM7116-2    CM71XX     192.168.128.5   4.11.0    SUCCESS
im7208-2 (nodes-6)    IM7208-2    IM72XX     192.168.128.7   4.10.0    SUCCESS
cm7196a-2 (nodes-5)   CM7196A-2   CM7196     192.168.128.6   4.10.0    SUCCESS
acm7004-2 (nodes-2)   ACM7004-2   ACM700X    192.168.128.3   4.11.0    SUCCESS
acm5508-2 (nodes-1)   ACM5508-2   ACM550X    192.168.128.2   4.1.1u2   SUCCESS
acm7004-5 (nodes-3)   ACM7004-5   ACM7004-5  192.168.128.4   4.11.0    SUCCESS
om2216-l (nodes-8)    OM2216-L    OMXXXX     192.168.128.9   21.Q2.1   FileNotFoundError
om1208-8e (nodes-7)   OM1208-8E   OMXXXX     192.168.128.8   21.Q2.1   FileNotFoundError
```

- Version shows the device version prior to upgrade.
- Result shows whether the upgrade for each device succeeded, or returns an error with more detail.

12.2.2 Results and Error Messages in node-upgrade

When the node-upgrade command is run with valid arguments and parameters, the program will return exit status 0 and the following results and error messages may be returned for each node listed.

Result	Causes
SUCCESS	Node upgrade succeeded
FileNotFoundError	No upgrade file found matching provided device family or version
UpgradeError	Device already has same or higher firmware version
	Network connection lost
IncompatibleFirmwareError	Firmware file provided does not match the product family

In addition, the following exit statuses may be returned.

Exit Status	Description
0	Command exited normally
1	Invalid parameter
2	Unknown argument

12.3 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.

12.4 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
```

12.4.1 Commands to try from within the `ogconfig-cli` tool

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

12.4.2 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'
```

12.4.3 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
```

12.4.4 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
```

12.4.5 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.

12.4.6 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
```

12.4.7 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
```

12.5 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.

12.6 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.

12.7 sysflash

`sysflash` is the shell-based tool for upgrading a Lighthouse instance's system. Sysflash will warn you if you do not have enough available space to upgrade to, though this is unlikely as space is reserved specifically for the upgrade process.

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**

12.8 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"
```

12.8.1 Select all nodes

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

12.8.2 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.

12.9 Add a custom 2nd NIC to a Lighthouse instance

1. In the Hypervisor configuration, add a 2nd network interface, and bind it to the required external network.
2. Reboot Lighthouse, and verify that net2 is visible.
3. Edit /etc/config/conman.conf and add two custom conns. The first conn configures the physical interface. The 2nd conn will vary depending if you want DHCP on the 2nd interface, or a Static IP address. Do not put both network-services-conns in.

```
# Custom Conns for 2nd interface

conn network-services-conn-init_net2
    var ifname net2
    start ip addr flush dev %ifname%
    start ip link set dev %ifname% up
    start mii-tool --restart %ifname%
    start sleep 2
    start ifconfig %ifname% up
    start sleep 2
    start bash -c "infod_client -o push -p
%ifname%.link_local -d $( ifconfig %ifname% | grep fe80 |
sed -r 's/.*(fe80::[^ ]+).*/\1/' )"
    stop ifconfig %ifname% down
```

```
# Use this conn if you want DHCP
conn network-services-conn
    parent network-services-conn-init_net2
    start infod_client -o push -p udhcpc.%ifname%.status -d
down
    up expect-return none bash -c "/sbin/udhcpc --syslog  --
release --now --interface %ifname% --foreground --script
'/usr/share/udhcpc/default.script' --vendorclass
'Opengear/Lighthouse' -x hostname:`hostname` --retries 3 --
pidfile /var/run/udhcpc.%ifname%.pid"
    stop ifconfig %ifname%:dhcp 0.0.0.0
    stop infod_client -o delete -p udhcpc.%ifname%
    stop infod_client -o push -p udhcpc.%ifname%.status -d
down
    test infod-equals udhcpc.%ifname%.status up
    testperiod 15
    testthreshold 20
    testthreshold 5

# Use this conn if you want a Static IP
conn network-services-conn
    parent network-services-conn-init_net2
    start ip addr add 192.168.0.1/255.255.255.0 broadcast
192.168.0.255 dev %ifname% label %ifname%:static1
    stop ip addr del 192.168.0.1/255.255.255.0 dev %ifname%
```

4. Restart conman to bring the 2nd interface up, then validate that net2 has an address.

```
root@lighthouse:~# pkill -HUP conman
... Wait 30 seconds
# If you have used DHCP
root@lighthouse:~# ifconfig net2:dhcp
net2:dhcp Link encap:Ethernet  HWaddr 52:54:00:8c:38:73
          inet
addr:192.168.82.39  Bcast:192.168.82.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1

root@lighthouse:~#

#if you have used Static
root@lighthouse:~# ifconfig net2:static1
net2:static1 Link encap:Ethernet  HWaddr 52:54:00:8c:38:73
          inet
addr:192.168.82.39  Bcast:192.168.82.255  Mask:255.255.255.0
```

```
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
```

5. Add a firewall rule to allow relevant incoming traffic on net2. If /etc/config/scripts/firewall-post does not exist, create the subdirectory.

```
root@lighthouse:~# mkdir -p /etc/config/scripts/
root@lighthouse:~# vi /etc/config/scripts/firewall-post
```

6. Editing the file, add this line after any existing rules,

```
  iptables -I INPUT -i net2 -j WanInput
```

7. Make sure the file is marked with execute permissions,

```
  chmod +x /etc/config/scripts/firewall-post
```

8. Force the firewall configurator to run, to install the new firewall rule

```
root@lighthouse:~# configurator_firewall --force
```

9. Verify you can access the device via the IP of net2.
10. Run configurator_local_network and re-test connectivity to verify that the changes will survive system configuration changes.

```
root@lighthouse:~# configurator_local_network
root@lighthouse:~# ifconfig net2:dhcp
net2:dhcp Link encap:Ethernet  HWaddr 52:54:00:8c:38:73
          inet
addr:192.168.82.39 Bcast:192.168.82.255 Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500 Metric:1

root@lighthouse:~#
```

13. Lighthouse CLI, Serial Port and REST API logging

Lighthouse offers command line interface (CLI) and REST API logs. Logging is disabled by default. This chapter covers:

Enabling logging

13.1 Logging overview and limitations

Once enabled, CLI and REST API logs can be found in `/var/log/messages`. All passwords are masked in the logs so that sensitive information is not stored in plain text or leaked.

When you enable logging, you do not need to restart or log out and in again.

There are a few caveats:

- CLI logging only works for Interactive (human-controlled) terminals. Commands generated by automated scripts will be logged.
- Commands such as ssh or telnet do not produce logs for the commands sent over the connection.
- Requests can be logged for all endpoints, however, only endpoints implemented in Lipy can have responses logged.
- The time format is temporarily different between logs made in Lipy and those made with the old system. In a future release, all endpoints will be implemented in Lipy, have consistent logs, and be able to log responses.

NOTE: These logs are not intended to be used as a definitive record of all commands that have ever been run. A malicious user with full root access can circumvent anything.

13.2 Using ogconfig-cli to enable logging

Logging should be enabled using `ogconfig-cli`. Use the following commands in a Lighthouse terminal to view, enable or disable logging:

- `system.logging_cli_`enabled - Enable/disable logging commands entered in the Lighthouse Terminal.
- `system.logging_rest_enabled` - Enable/disable basic logging for the REST API. These logs report the following information about every REST API call:
  - Time
  - Request type (GET/POST/PUT/DELETE)
  - HTTP status code
  - Username
  - Source IP Address
  - Endpoint
- `system.logging_rest_request_enabled` - Enable/disable request logging for every REST API call. In addition to the basic logging, also logs the request body that was provided by the client, if any.
  NOTE: Requires system.logging_rest_enabled to be enabled.

- `system.logging_rest_response_enabled` - Enable/disable response logging for every REST API call. In addition to the basic logging, also logs the response body that was sent to the client, if any.

  NOTE: Requires `system.logging_rest_enabled` to be enabled**.**

13.2.1 Add node and port to Lighthouse logs

Node and port logs will log all access to nodes via Lighthouse using the **pmshell** function, including which console, and which port was accessed by the user when logged in.

When system logging is enabled, user and node selection and user and Port selection is logged.

To enable node and port logging on Lighthouse:

1. Navigate to **Lighthouse > Manage > Lighthouse > Local Terminal**.
2. Login to a user that has rights to use CLI and ogconfig-cli.
3. Node and port logging can only be enabled through *ogconfig-cli* as follows:

```
root@lighthouse:~# ogconfig-cli

ogcfg> set system.logging_cli_enabled true

root-1-system_logging_cli_enabled: Integer <True>

ogcfg> push

OK

ogcfg> exit
```

Once node and port logging is enabled, you can view any logs recorded in your Lighthouse's syslog, located at /var/log/messages.

13.3 Example logs

Here is an example of logs without request or response logging:

```
2020-03-17 15:29:37,237 INFO  [root:117][waitress] POST 400 (root |
192.168.1.1) - /api/v3.4/system/licenses/file

2020-03-17 15:30:23,034 INFO  [root:117][waitress] GET 200 (root |
192.168.1.1) - /api/v3.4/users?page=1&per_page=10
```

Here is an example of logs with request or response logging

NOTE: These logs differ slightly due to being logged with different systems:

```
2020-05-11T05:45:09.567214+00:00 lighthouse rest_api_log[2465]: PUT 200 (root
| fd07:2218:1350:4b:a438:f8ff:fe4f:65fc) -
/api/v3.4/system/cli_session_timeout
REQUEST={"system_cli_session_timeout":{"timeout":0}}

2020-05-11 05:45:18,999 INFO  [lipy.logging.rest_api:62][waitress] GET 200
(root | fd07:2218:1350:4b:a438:f8ff:fe4f:65fc) -
/api/v3.4/users?page=1&per_page=10 RESPONSE={'users': [{'username': 'root',
'description': 'System wide SuperUser account', 'enabled': True, 'id':
'users-1', 'no_password': False, 'expired': False, 'locked_out': False,
'rights': {'delete': True, 'modify': True}, 'groups': ['groups-2']}], 'meta':
{'total_pages': 1}}
```

13.4 Checking if logging is enabled

To establish if logging is enabled run these commands on the Lighthouse local terminal:

```
root@lighthouse:~# ogconfig-cli

ogcfg> print system
```

This will produce output with Boolean values:

- `system.logging_cli_enabled (bool): false`
- `system.logging_rest_enabled (bool): false`
- `system.logging_rest_request_enabled (bool): false`
- `system.logging_rest_response_enabled (bool): false`

13.5 Enable logging

To enable logging, run these commands on the Lighthouse local terminal:

```
root@lighthouse:~# ogconfig-cli

ogcfg> set <value> true

root-1-<value>: Integer <True>

ogcfg> push

OK

ogcfg> exit
```

Replace *`<value>`* with the desired setting:

- `system.logging_cli_enabled`
- `system.logging_rest_enabled`
- `system.logging_rest_request_enabled` (requires `system.logging_rest_enabled`)

- `system.logging_rest_response_enabled` (requires `system.logging_rest_enabled`)

13.6 Disable logging

```
root@lighthouse:~# ogconfig-cli

ogcfg> set <value> false

root-1-<value>: Integer <False>

ogcfg> push

OK

ogcfg> exit
```

Replace *<value>* with the desired setting:

- `system.logging_cli_enabled`
- `system.logging_rest_enabled`
- `system.logging_rest_request_enabled`
- `system.logging_rest_response_enabled`

14. System upgrades

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

NOTE: AWS requires `.aws.lh_upg` and Microsoft Azure requires `.azure.lh_upg.`

NOTE: Due to disk restructuring, there were no lh_upg files for ***20.Q3.x***.

NOTE: Incremental upgrades to Lighthouse using lh_upg files are only supported from ***20.Q3.x*** and not earlier releases.

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.

From the release of ***20.Q3.x***, Lighthouse uses Logical Volume Management (LVM) for the disks to support expanding the available disk space for the Lighthouse virtual machine, as well as enable other benefits of LVM such as snapshot and restore functionality.

Upgrading from pre-LVM Lighthouse to LVM Lighthouse

NOTE: You cannot directly upgrade from pre-LVM Lighthouse to LVM based Lighthouse. In order to upgrade, you must have the release ***20.Q2.x*** installed before upgrading to **20.Q3.x** and later upgrades by following the process outlined below.

You need to:

1. Perform a configuration backup of your pre-LVM primary Lighthouse and keep it safe.
2. Unenroll any secondary Lighthouse instances from your primary Lighthouse.
3. Upgrade your Lighthouse system to **20.Q2.x** using the `.lh_upg` file. Skip this step if your Lighthouse is already running 20.Q2.x.
4. Perform another configuration backup of Lighthouse running **20.Q2.x** and save it as a separate file. This configuration backup will be imported into the LVM based Lighthouse.
5. Shut down the **20.Q2.x** Lighthouse before continuing these steps.
6. Deploy a new instance of Lighthouse version **20.Q3.x** (LVM).
   - The new instance must have **all the same IP addresses and hostname** as the previous Lighthouse instance; this allows nodes and secondary Lighthouse instances to reconnect.
7. Import the configuration backup of **20.Q2.x** into the new primary Lighthouse.
8. Confirm nodes successfully connect to the new primary Lighthouse.
9. Upgrade the primary Lighthouse to each major release in sequence up to the latest release.
10. Deploy new LVM based secondary Lighthouse instances of the latest release and make sure their external endpoint addresses are correct.
11. Enroll the new secondary Lighthouse instances, one by one, to the new primary Lighthouse.

Upgrading LVM Lighthouse to LVM Lighthouse

You will be able to upgrade from LVM based version to a new LVM based version following the steps in sections 14.1 and 14.2.

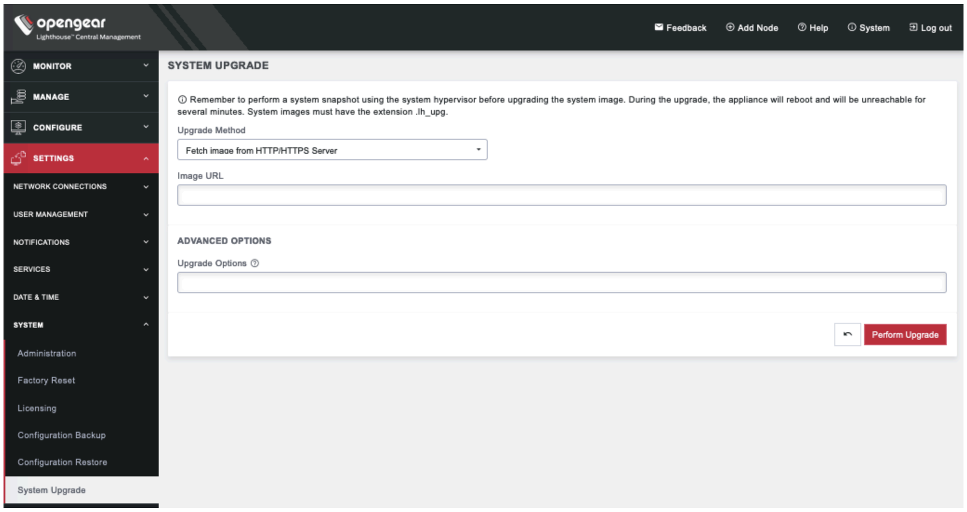
NOTE: There is a logical volume called **lh_upg** which is reserved for performing upgrades. You should not delete or use this logical volume.

14.1 Upgrading the system from within Lighthouse

> **NOTE:** Starting with ***20.Q3.x*** release, Lighthouse must be upgraded in succession. For example, to upgrade to a version that is several releases newer than your current release, you will need to install all the major releases in between to install the newest one.

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***.



SYSTEM UPGRADE

Remember to perform a system snapshot using the system hypervisor before upgrading the system image. During the upgrade, the appliance will reboot and will be unreachable for several minutes. System images must have the extension .lh_upg.

Upgrade Method: Fetch image from HTTP/HTTPS Server

Image URL

ADVANCED OPTIONS

Upgrade Options

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.

14.2 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**.

NOTE: Before using sysflash, we recommend that you check available disk space when manually uploading .lh upgrade files. We also suggest you use `/mnt/nvram` as the path.

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.

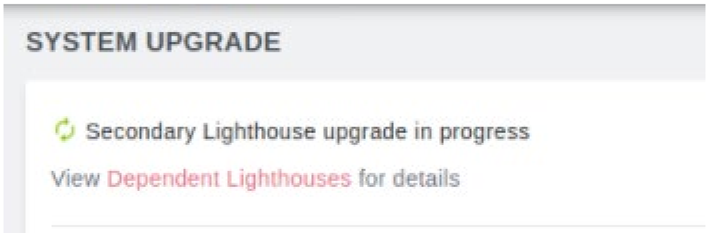
14.3 Upgrading Dependent Multiple Instances of Lighthouse

NOTE: Starting with ***20.Q3.x*** release, Lighthouse must be upgraded in succession. For example, to upgrade to a version that is several releases newer than your current release, you will need to install all the major releases in between to install the newest one.

Before a multiple instance upgrade is attempted, compatibility and status checks are performed on primary/secondary instances to pre-empt possible failure points.

Secondary Lighthouse upgrades are performed in parallel (not in a queue) to speed up the overall process of rolling upgrades.

Where there are multiple instances of Lighthouse, when a system upgrade is being performed the status of dependent instances is flagged in the System Upgrade page:

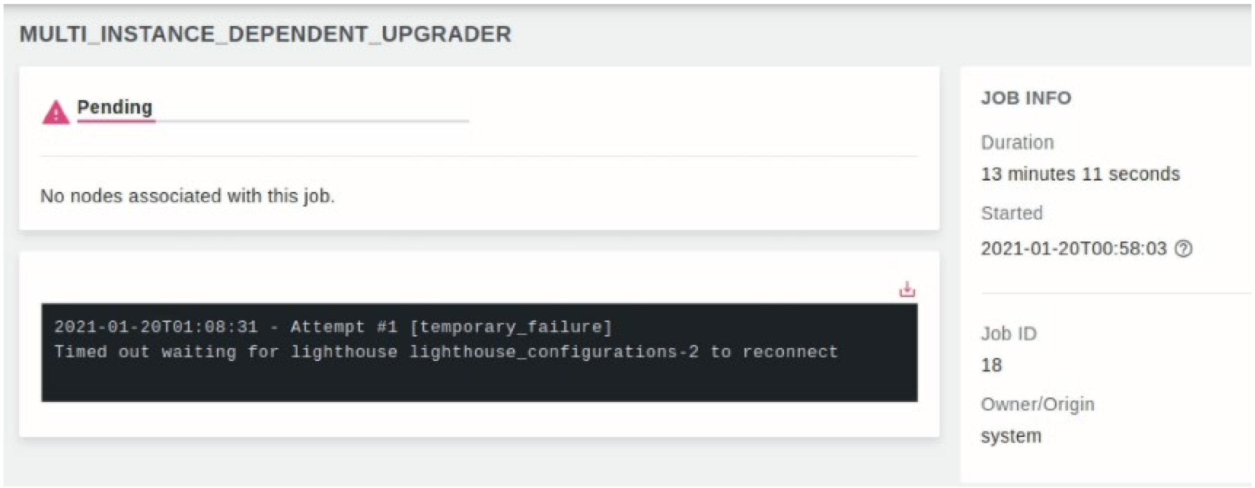


SYSTEM UPGRADE

Secondary Lighthouse upgrade in progress

View Dependent Lighthouses for details

Click the **Dependent Lighthouses** link (red text) to view the upgrade process status for dependent Lighthouse nodes. Click **View Job Details** link to see details of the update progress and any problems.



MULTI_INSTANCE_DEPENDENT_UPGRADER

Pending

No nodes associated with this job.

```
2021-01-20T01:08:31 - Attempt #1 [temporary_failure]
Timed out waiting for lighthouse lighthouse_configurations-2 to reconnect
```

JOB INFO

Duration
13 minutes 11 seconds

Started
2021-01-20T00:58:03

Job ID
18

Owner/Origin
system

During a system upgrade, notification/status elements are flagged in the following scenarios:

- When an upgrade is attempted, a pass/fail notification on the instance.
- When an upgrade is attempted on a secondary instance, a pass/fail notification on the associated primary instance.

The system is designed to be stable enough in a post-failure situation for an administrator to diagnose and fix any error(s) manually and re-attempt the upgrade. In cases where an upgrade on an instance is attempted and fails, Lighthouses in the MI environment will make a "best effort" attempt to return to a stable state.

Information about the upgrade progress and status is visible in the Lighthouse **Jobs** page.

15. Adding Disk Space to Lighthouse

Additional physical volumes can be added to the volume group as required, and the logical volumes extended using `lvextend` and `resize2fs` to take advantage of the additional space.

15.1 Adding a New Disk

AWS specific instructions

Launch an LVM Lighthouse instance as per our guidelines or your own deployment processes and note the instance ID.

To add a volume to an AWS Lighthouse without having to shut down the LH:

1. In the AWS web console, go to **Volumes** and create a new 5GB volume.
   NOTE: Make sure this volume in the same availability zone as your LH instance.
2. Once the volume is created, select it and click the **Actions** button and select **Attach Volume**.
3. Enter the LH instance ID for the instance field and `/dev/xvdb` (or /`dev/xvdd, /dev/xvde` and so on) as the device and click **Attach**.

When you SSH into the LH you should be able to see the new volume as `/dev/xvdb` (or whatever device name you gave it).

qemu specific instructions

Launch a qemu Lighthouse instance as per our guidelines or your own deployment.

To add a volume to the instance:

1. Shutdown the instance with the following command:
   ```
   shutdown -h now
   ```

2. Create a new disk for the LH. You can use a different number for “count” which is in MiB.

   ```
   dd if=/dev/zero of=/tmp/new_lh_disk.lh_hdd bs=1024k count=256
   qemu-img convert -p -f raw -O qcow2 /tmp/new_lh_disk.lh /tmp/new_lh_disk.qcow2
   ```

3. Restart your qemu instance but make sure to add the new qcow2 disk to the command.

   Here is an example of what you should add to your qemu command when launching the instance:
   ```
   -drive if=scsi,file=/tmp/new_lh_disk.qcow2
   ```

   NOTE: this is just an example. You should specify the disk in a similar way to how you specified the primary Lighthouse disk. and you should make sure that the new disk is specified last, otherwise your disk will appear out of order when you boot the Lighthouse.

4. Once the LH boots you should have a new /dev/sdX device and the 'unused_disks' command should report that disk when you log in.

Azure specific instructions

Launch the LVM Lighthouse instance as per our guidelines or your own deployment.

To add a volume to the instance, use the following link to attach a new disk to your Lighthouse VM. Stop before you reach the section, "**Connect to the Linux VM to mount the new disk**."

https://docs.microsoft.com/en-us/azure/virtual-machines/linux/attach-disk-portal

Hyper-V specific instructions

Launch the LVM Lighthouse instance as per our guidelines or your own deployment.

To add a volume to the instance:

1. Shutdown your Hyper-V Lighthouse instance.
2. Open your Hyper-V manager.
3. Navigate to the VM list and locate your Lighthouse VM.
4. Right click on the instance and click **Settings**.
5. Click on the **SCSI controller.**
6. Select **Hard drive** on the right and click **Add**.
7. Select **Virtual hard disk** and click **New**.
8. Follow the prompts and select the options that best suit your needs and environment.
9. Once you've created the disk, click **Apply** in the VM settings window.
10. Restart the Lighthouse.

VirtualBox specific instructions

Launch the LVM Lighthouse instance as per our guidelines from the `.ova` file or your own deployment.

To add a volume to the instance:

1. Shutdown the Lighthouse instance.
2. In the VirtualBox UI, locate your Lighthouse instance and right-click it.
3. Select **Settings**.
4. Select **Storage** on the left.
5. Click the **Controller: SCSI** in the disk list.
6. You will see two small icons, both with a green '+' symbol. Hover your mouse over the one that says **Adds a hard disk** and click it.
7. Click the **Create** icon.
8. Follow the prompts to create a new disk image.
9. Select the new disk image and click the **Choose** button.
10. Click **Ok** to exit the VM settings window.
11. Restart the Lighthouse.

15.2 Using the new disk to increase the lh_data logical volume

1. Add the new disk to the LH VM (platform dependent, see above).
2. Log into the shell on Lighthouse. you should see the new "unused" disk listed in the welcome message. This is the case for any non-system disks aren't currently being used by the LVM system.
3. Create a partition on the new disk:

   ```
   fdisk /dev/sdb (or /dev/xvdb, or /dev/(sd|xvd)X )
   ```

   NOTE: Be sure specify the correct disk, it might be `/dev/xvdb` on AWS.

4. Type 'n' and ENTER to create a new partition.
5. Type 'p' and ENTER to create a primary parition.
6. Continue hitting ENTER to accept the defaults to use the whole disk.
7. Type 'w' and ENTER to write the changes and exit fdisk.
8. Add the new partition as a physical volume (assuming you are now using `/dev/sdb1,` note that `/dev/xvdb1` will now be mapped to `/dev/sdb1` so make sure you use `sdb1`).

   ```
   pvcreate /dev/sdb1
   ```

9. Extend the volume group with the new physical volume.

   ```
   vgextend lhvg /dev/sdb1
   ```

10. Assuming the new disk gives you at least 2GB of extra space, expand the **`lh_data`** logical volume.

    ```
    lvextend -L +2G /dev/mapper/lhvg-lh_data
    ```

11. Update the file system of the lh_data disk to use the extra space.

    ```
    resize2fs /dev/mapper/lhvg-lh_data
    ```

12. When you log into the shell, the disk should no longer be listed as "unused".

16. Troubleshooting

This chapter covers:

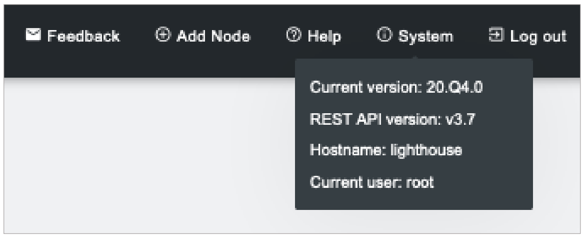
- Finding the current Lighthouse version
- Generating technical support reports
- Backing up and restoring Lighthouse configuration

16.1 Finding the current Lighthouse instance version

There are two ways to find the current Lighthouse version.

16.1.1 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**.



Current version: 20.Q4.0
REST API version: v3.7
Hostname: lighthouse
Current user: root

16.1.2 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
2021.Q4.0
```

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.

16.1.3 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
2021.Q4.0
```

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 Lighthouse product version.

```
# cat /etc/issue
Opengear Lighthouse 2021.Q4.0 \n \l
```

16.2 Technical support reports

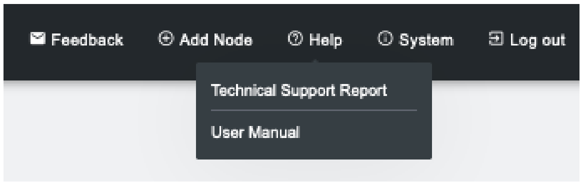
Lighthouse can generate a technical support report that includes Lighthouse configuration information and the current system log for the Lighthouse VM.

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

16.2.1 Generate a support report via the Lighthouse interface

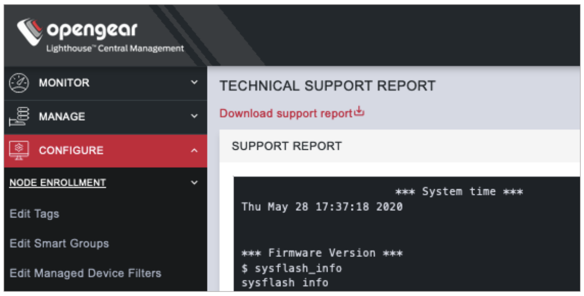
To generate a complete configuration and status report regarding a given *Lighthouse VM*:

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**.



```
                    *** System time ***
Thu May 28 17:37:18 2020


*** Firmware Version ***
$ sysflash_info
sysflash info
```

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.

16.2.2 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.

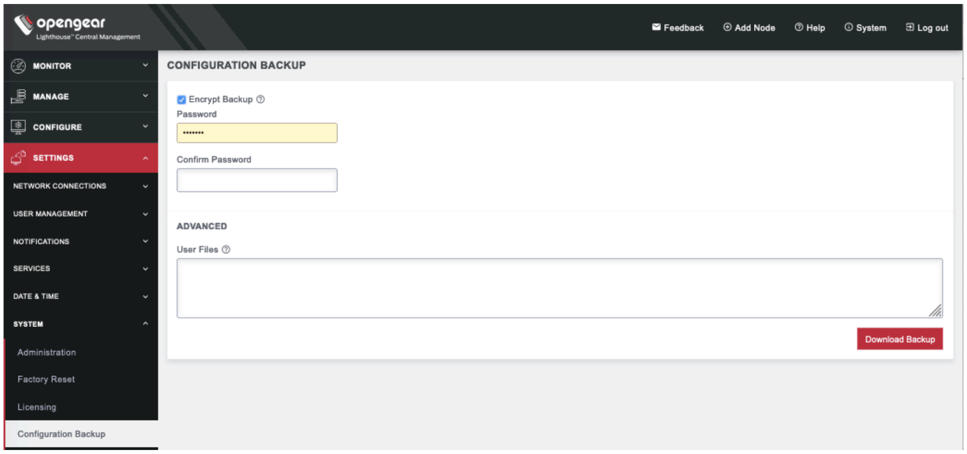
- Use the FTP client on Lighthouse to copy the file to an FTP server. Passive mode must be used for this to work. Example:

  ```
  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.
  ```

16.3 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**.



CONFIGURATION BACKUP

☑ Encrypt Backup

Password: •••••••

Confirm Password:

ADVANCED

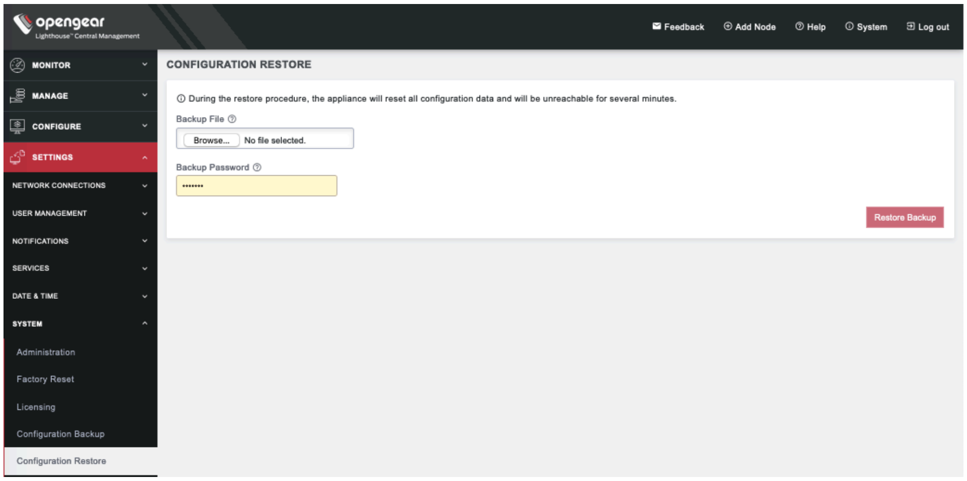
User Files:

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`

16.4 Configuration Restore

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

1. Select **SETTINGS > SYSTEM > Configuration Restore**.



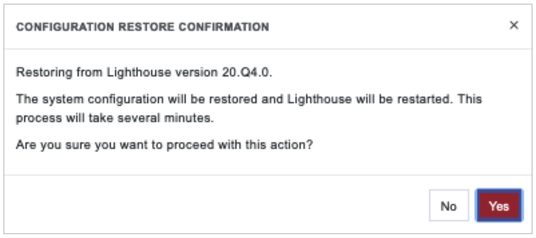
CONFIGURATION RESTORE

During the restore procedure, the appliance will reset all configuration data and will be unreachable for several minutes.

Backup File: No file selected.

Backup Password: •••••••

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. A **Configuration Restore Confirmation** dialog opens, click **Yes**.



CONFIGURATION RESTORE CONFIRMATION

Restoring from Lighthouse version 20.Q4.0.

The system configuration will be restored and Lighthouse will be restarted. This process will take several minutes.

Are you sure you want to proceed with this action?

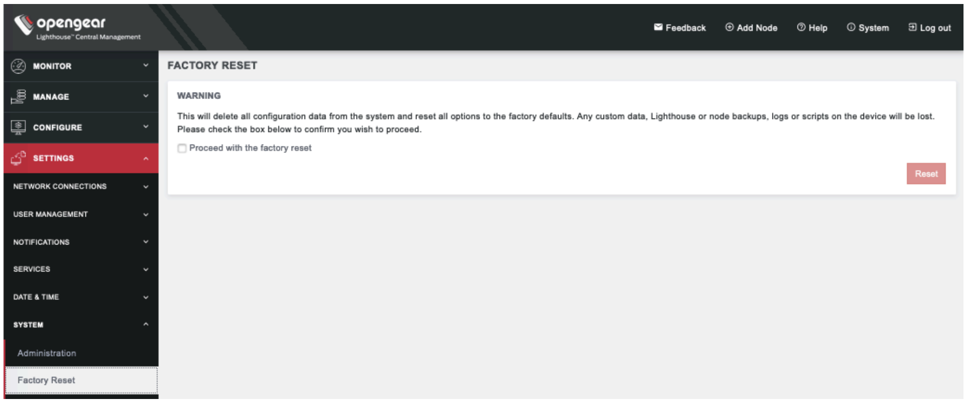
6. Lighthouse will restore the backup and any included user files and restart.

16.5 Returning a Lighthouse instance to factory settings

NOTE: During this process, the current Lighthouse configuration will be overwritten 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**.



FACTORY RESET

WARNING

This will delete all configuration data from the system and reset all options to the factory defaults. Any custom data, Lighthouse or node backups, logs or scripts on the device will be lost. Please check the box below to confirm you wish to proceed.

□ Proceed with the 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**.

17. Changing Docker IP Ranges

Docker powers the NetOps platform within Lighthouse. By default, Docker and NetOps utilize the 172.17.0.0/16 and 172.18.0.0/16 subnets. This has the potential to cause collisions inside of some networks.

To avoid this, you can change these settings.

To update Dockers subnet, you need to alter 2 parameters, Docker's default subnet and the NetOps modules subnet. To do so:

1. Login to the Lighthouse shell CLI as a Lighthouse Administrator or the root user
2. Ascertain the number of running containers to ensure you select an appropriate subnet size

   ```
   sudo docker ps -q | wc -l
   ```

3. Open a config CLI session on the Lighthouse Server and run the following to enter configuration management

   ```
   ogconfig-cli
   ```

4. Set the IP Range of the Docker subnet in CIDR format

   ```
   set services.nom.default_subnet "10.123.17.1/24"
   ```

5. Set the IP Range of the NetOps subnet in CIDR format

   ```
   set services.nom.netops_subnet "10.123.18.0/24"
   ```

6. Push the config to become the running config

   ```
   push
   ```

7. Exit the configuration management

   ```
   exit
   ```

8. Restart Docker

   ```
   sudo /etc/init.d/docker.init restart
   ```

9. Restart the NetOps Module(s)

   ```
   sudo /etc/init.d/docker.init reset
   ```

   NOTE: The network mask selected for these subnets limits the maximum number of containers that can run on Lighthouse. NetOps currently runs up to approximately 10 containers.

18. EULA and GPL

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