opengear
A DIGI Company

LIGHTHOUSE 24.06.02

USER GUIDE

COPYRIGHT ©

Information in this document is subject to change without notice and does not represent a commitment on the part of Opengear. Opengear provides this document “as is,” without warranty of any kind, expressed or implied, including, but not limited to, the implied warranties of fitness or merchantability for a particular purpose.

Opengear may make improvements and/or changes in this manual or in the product(s) and/or the program(s) described in this manual at any time. This product could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes may be incorporated in new editions of the publication.

CONTENTS

WHAT'S NEW

UPDATES FOR 24.06

Following are the new features available in Lighthouse 24.06.

UI ENHANCEMENTS

- Updated user interface and work flows.
- Reworked navigation and menus.
- Updated styling and icons.

CONNECTED RESOURCE GATEWAY

- New feature to build and manage a catalog of resources that are within the Smart Management Fabric (SMF) discovered networks.
- Support for clientless network access to resources via either SSH, HTTP, or HTTPS proxy services.
- Accessed via the new *Resources* page.

RESOURCE TAGGING

- Support for user defined resource filters.
- Support for user defined tags.
- User group access control with resource filters.
- Searching with Resource Filters.
- *Port Tags* have been renamed to *Resource Tags* in the UI and API expanding it’s functionality to cover both Serial Ports and Resources.

CERTIFICATE HANDLING ENHANCEMENTS

- Automatic renewal of node and other certificates used to authenticate to Lighthouse.
- Configuration of certificate management via the new `cert_manage` CLI command

AWS LIGHTHOUSE

- For fresh AWS installs, a new default super user account `lhadmin` is provided and should be used for initial configuration.
- By default, root login is now disabled over SSH for new installs. Existing AWS installs are unaffected.
- By default, password authentication is now disabled over SSH. Existing AWS installs are unaffected.
- Added support for AWS Lighthouse to allow login on IPv6-only subnets, enabling access for instances without public IPv4 addresses or any IPv4 addresses.

SMART GROUP RENAMING

- Smart Groups are now renamed to Node Filters within the Lighthouse UI. However, Smart Groups retain their naming within the CLI and API.

UPDATES FOR 24.06.2

Following are the new features available in Lighthouse 24.06.2

AWS MARKETPLACE

Lighthouse is available on Amazon Web Services (AWS) Marketplace. You can subscribe to get access to our published AMIs, however, you must still contact Sales to obtain a license.

THIRD-PARTY NODES

The Connect Resource Gateway can now be leveraged to proxy via SSH, HTTP or HTTPS to configure devices/third-party nodes.

ABOUT THIS USER GUIDE

This user guide describes how to use Lighthouse and is current as of 24.06.02

When using a minor release, there may or may not be a specific version of the user guide for that release.

LIGHTHOUSE OVERVIEW

Lighthouse allows you to centrally access, manage, and monitor a network of Opengear devices (referred to as nodes throughout this guide). With the Automation Edition subscription, you can access the Smart Management Fabric to monitor resources in your network. Lighthouse is a virtual machine that can be hosted on various platforms like VMware or Hyper-V (on customer hardware) or on supported cloud providers (AWS or Azure).

LIGHTHOUSE ARCHITECTURE

Lighthouse is an API-driven platform that provides secure access to remote networks regardless of how devices are connected or how a user interacts with the system.

In combination with select Opengear Appliances, Lighthouse can push and manage Docker containers to each remote location to provide additional functionality and automation.

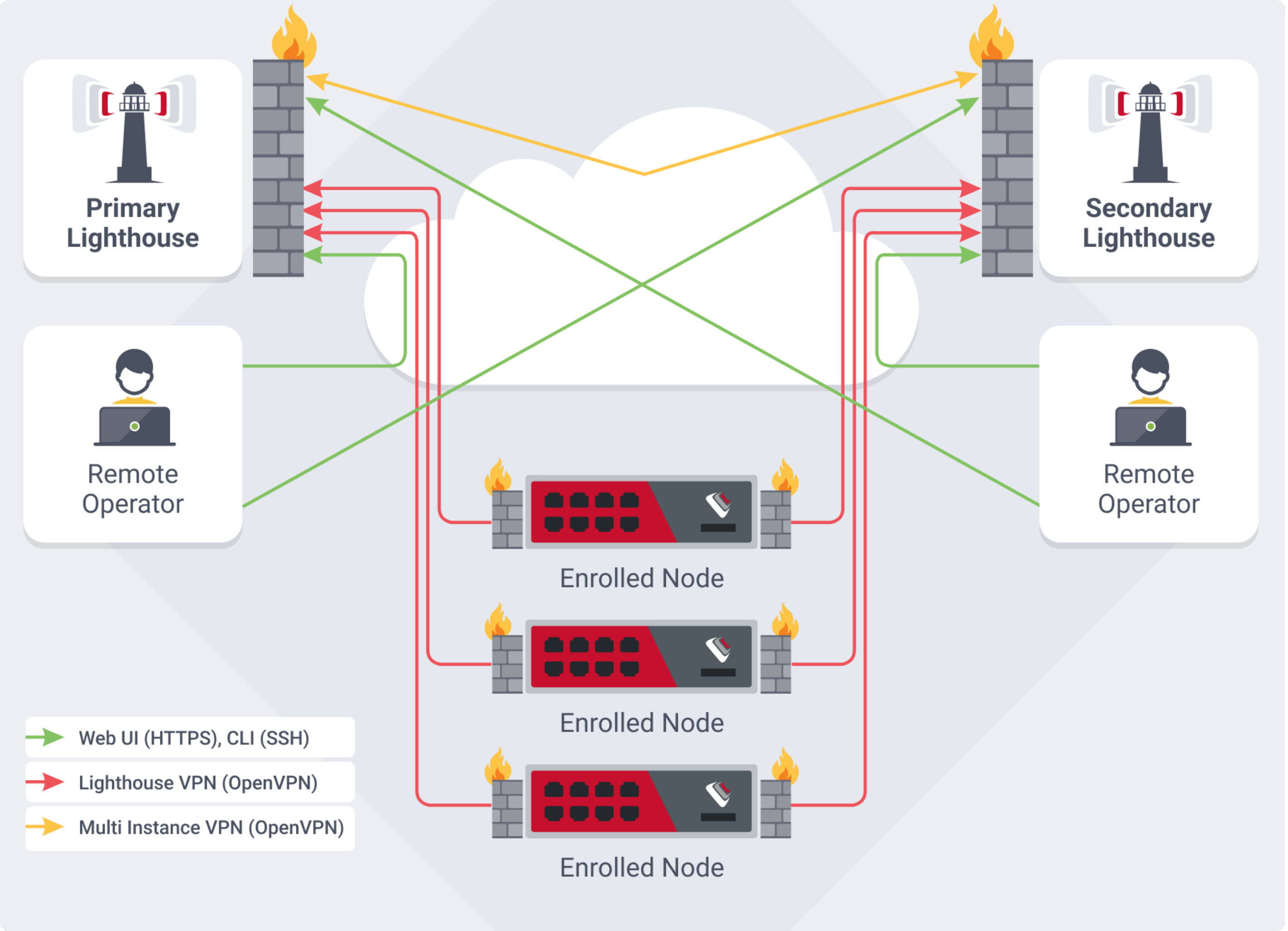
Users with the Lighthouse Automation Edition can enable Smart Management Fabric to allow all devices connected to a supported Opengear node to have direct routed IP access with each other. For example:

- A user with access to Lighthouse from a device that is attached to the management subnet can access any enrolled Opengear node and the third-party network equipment attached to the enrolled Opengear node via routed IP.
- An enterprise user connected to an IP network with a route to an enrolled Opengear device can access any other enrolled Opengear device (via IP).
- Devices attached to Opengear nodes can use IP routed connectivity to other devices attached to other Opengear nodes.
- Network automation and configuration tools deployed on centrally located servers can use IP routed connectivity to devices and systems attached to Opengear nodes. For example, vCenter is the configuration tool and it is used to provision eSXI server instances attached to Opengear nodes.

Note: The Smart Management Fabric represents an advanced functionality designed to offer heightened flexibility and accessibility to network and IT professionals throughout the network fabric.

However users must acknowledge the potential risk of overexposing the network, which could lead to bypassing Layer 2 or Layer 3 access control measures.

Nodes connect to a central Lighthouse instance over an OpenVPN tunnel, and are accessed, managed, and monitored via services transported over the VPN tunnel.



Note: This diagram depicts Lighthouse set up in a High Availability configuration. If there is no secondary Lighthouse, the setup remains the same but without the secondary elements.

LIGHTHOUSE VIRTUAL MACHINE REQUIREMENTS

Lighthouse deploys as an application running in a Linux-based virtual machine (VM). 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.

The Lighthouse binary is available in Open, VMware and Hyper-V specific Virtual Machine formats. VM managers such as Boxes, KVM and VirtualBox can make use of the open format. Lighthouse binaries are also available for cloud hosting services including Amazon's AWS, and Microsoft Azure.

NetOps modules are released independently of Lighthouse software or Operations Manager firmware. These releases are shipped to Amazon's ECR Public Gallery, where they can be fetched by Lighthouse then deployed to all activated nodes by Lighthouse. NetOps modules can also be downloaded off the Opengear FTP site, and uploaded manually on Lighthouse.

To host Lighthouse, the VM must be configured to support:

- A minimum 50GB SCSI disk. Start with more if you think your network will expand. Additional space may be required depending on your feature usage on Lighthouse and the scale of the network.
- 1 x network interface card, preferably paravirtualised (virtio, vmxnet3), Realtek rtl8139, or Intel e1000 are also supported, bridged.
- VGA console for initial setup.

Tip: CPU and RAM utilization increase with the number of enrolled nodes, network utilization and storage.

CPU AND RAM GUIDELINES

No. Nodes	Min. CPU Cores	Min. RAM
< 500	4 x 64-bit CPU cores	16GB
500 - 1000	8 x 64-bit CPU cores	32GB
> 1000 Nodes	16 x 64-bit CPU cores	64GB

ENABLING SMART MANAGEMENT FABRIC GUIDELINES

No. Nodes	Min. CPU Cores	Min. RAM
< 2500 Nodes	4 x 64-bit CPU cores	16GB
> 2500 Nodes	8 x 64-bit CPU cores	32GB

Tip: For large deployments, contact us for guidance on the deployment options, including low and zero-touch enrollment.

LIGHTHOUSE TO NODE INTERACTIONS

When a node is enrolled to Lighthouse, a Virtual Private Network (VPN) tunnel is established between Lighthouse and the node. This provides secure encrypted IP networking, which is resilient to changes in the underlying network (such as node failover to cellular).

Lighthouse interacts with the node over the VPN tunnel, mostly via the node's REST API and SSH. Nodes notify Lighthouse of changes to their configuration and status.

The node web UI is accessible directly through Lighthouse and is proxied via the VPN. This allows secure user access to the node even if it is behind a firewall with no direct HTTPS access.

LIGHTHOUSE CERTIFICATE MANAGEMENT

Lighthouse uses X.509 certificates for node authentication to the Lighthouse VPN and REST API. Certificates are issued by the internal Lighthouse certificate authority as part of the node enrollment process, and are automatically renewed by Lighthouse before expiry. The replacement certificates are pushed from Lighthouse to connected nodes.

Lighthouse will manage certificates automatically, and no action is required by the customer. However, there are some things to be aware of:

- Node certificates are revoked by Lighthouse when a node is unenrolled, or when the certificate has been replaced (after the replacement certificate has been used to successfully connect to the Lighthouse VPN). Revoked certificates cannot be used to authenticate to the Lighthouse VPN or REST API.
- The Lighthouse CA can be revoked after it has been renewed when all nodes have been notified of the change. Remediate or unenroll any disconnected nodes to complete this operation.
- If a node is disconnected from Lighthouse for an extended period of time, it may not be possible to push the updated certificate to the node. Lighthouse will retry the push job regularly until the node's existing certificate has expired, at which point the node will have to be manually re-enrolled.

PRECAUTIONS

If an old Lighthouse configuration backup is restored to Lighthouse, the node certificate details in the backup may no longer match those on the nodes themselves, in which case the nodes will fail to connect to Lighthouse. Ensure that configuration backups of Lighthouse are kept up to date.

Similarly, if a node has its configuration restored from an old backup, its certificate may no longer match the one expected in Lighthouse. In these cases, it will be necessary to unenroll and re-enroll the node. To avoid these situations, ensure configuration backups of nodes are kept up to date.

Note: There is a limitation on Operations Manger (OM) and Console Manger CM8XXX nodes where a Lighthouse VPN connection configuration is not retained in the node backup.

The Lighthouse VPN certificate and client certificates validity periods should be no greater than the CA certificate used to issue them. The existing certificate validity periods can be seen by running the show sub-command and the pre-configured defaults by using the `--defaults` option.

Lighthouse will automatically process scheduled certificate updates daily at 1 AM Lighthouse system time. Under normal circumstances there is no requirement to run `cert_manage run` manually.

CONFIGURATION

The `cert_manage` command can be used to control various aspects of certificate management in Lighthouse. The default settings are recommended, and should only be changed with caution.

Only users with sudo access on the primary Lighthouse CLI (for example, via the admin group) can configure certificate management.

Note: All functionality is available only via the Lighthouse CLI. There is no UI or REST API interface for the certificate management feature. The 🗎 **Jobs** page on the Lighthouse UI shows node certificate update jobs.

SCHEDULING

Certificate renewal jobs are scheduled using cron to run at 1 AM (Lighthouse system time), every day. An administrator may choose to update the frequency of the cron job under `/etc/cron.d/rotate_certificates.cron`.

LOG FILE

The certificate management logs can be found in `/var/log/cert_manager.log`.

USING MULTIPLE LIGHTHOUSES

Lighthouse offers high availability with the Multiple Instance feature, which allows you to set up secondary instances of Lighthouse. The secondary Lighthouses automatically replicate the database from the primary Lighthouse instance, and maintains connections to all of its enrolled nodes.

Secondary instances 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 Lighthouse instance, which will then update the information displayed on the secondary instance.

Note: Secondary Lighthouse user interfaces are read-only.

The multiple instance feature has the following limitations:

- If external network addresses on the primary or secondary Lighthouses are updated after a secondary Lighthouse has been enrolled, it may break replication.
- Up to ten secondary instances can be enrolled.
- Ensure that you re-configure instance specific settings such as hostname, external endpoints, and time zone on a secondary instance before adding the secondary instance to the primary Lighthouse in a normal way through UI.

- Only secondary Lighthouse instances with zero nodes can be enrolled to the primary Lighthouse.
- Removing a secondary Lighthouse instance will initiate a factory reset of the removed Lighthouse.

EULA AND GPL

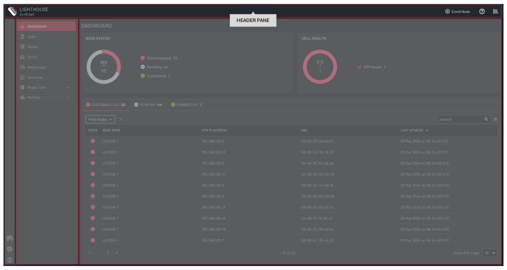
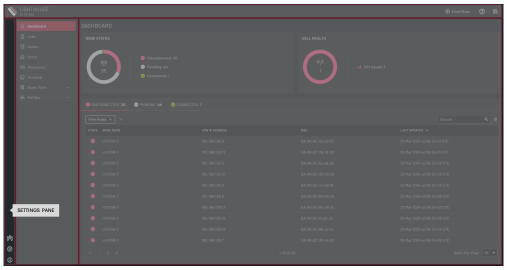
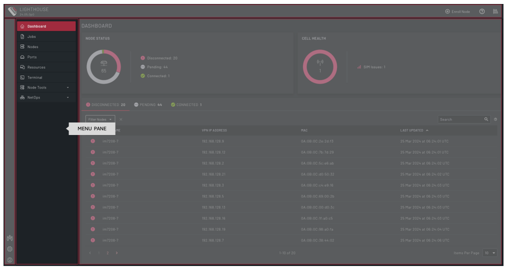
The current Opengear end-user license agreement can be found at Opengear License EULA | Digi International.

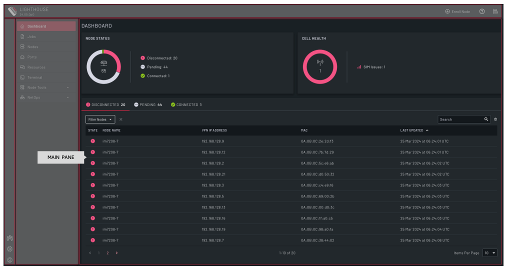
LIGHTHOUSE USER INTERFACE

Lighthouse offers a easy to navigate interface to manage your network.

LIGHTHOUSE USER INTERFACE SECTIONS

The user interface consists of the following main sections:

Section	Description	
Header pane	Provides access to the help menu and information for the Lighthouse instance as well as a shortcut to *enroll a node*	
Settings panel	Location for the Lighthouse settings including: *users and account management* menu, *settings* menu and *user* options.	
Menu pane	Access to the key dashboards for managing *nodes*, *ports*, *connected resources*, *jobs* along with tools for configuring nodes, access to a local web-based terminal and NetOps management.	

Section	Description	
Main pane	This is the main information pane and displays the Lighthouse pages.	

DASHBOARD

NODE STATUS

65

Disconnected: 20
Pending: 44
Connected: 1

CELL HEALTH

1

SIM Issues: 1

DISCONNECTED 20 · PENDING 44 · CONNECTED 1

STATE	NODE NAME	VPN IP ADDRESS	MAC	LAST UPDATED
!	im7208-7	192.168.128.9	0A:0B:0C:2e:2d:f3	25 Mar 2024 at 06:24:01 UTC
!	im7208-7	192.168.128.12	0A:0B:0C:7b:7d:29	25 Mar 2024 at 06:24:01 UTC
!	im7208-7	192.168.128.2	0A:0B:0C:5c:e6:ab	25 Mar 2024 at 06:24:02 UTC
!	im7208-7	192.168.128.21	0A:0B:0C:d0:50:32	25 Mar 2024 at 06:24:02 UTC
!	im7208-7	192.168.128.3	0A:0B:0C:c4:e9:16	25 Mar 2024 at 06:24:03 UTC
!	im7208-7	192.168.128.5	0A:0B:0C:69:00:2b	25 Mar 2024 at 06:24:03 UTC
!	im7208-7	192.168.128.13	0A:0B:0C:00:d0:3c	25 Mar 2024 at 06:24:03 UTC
!	im7208-7	192.168.128.16	0A:0B:0C:1f:a0:c5	25 Mar 2024 at 06:24:03 UTC
!	im7208-7	192.168.128.19	0A:0B:0C:98:a0:fa	25 Mar 2024 at 06:24:04 UTC
!	im7208-7	192.168.128.7	0A:0B:0C:38:44:02	25 Mar 2024 at 06:24:06 UTC

1 2

1-10 of 20

SETTINGS PANE

The settings pane provides access for the following:

	Users & Accounts	Expand this to access options for managing Lighthouse users and accounts including configuration of groups and roles, authentication policies and SSH settings.
	Settings	Expand this to access options for managing Lighthouse settings including configuration of filters and tags, multi instance settings, Lighthouse services, and general system settings.
	Personalization	Expand to set the Lighthouse user interface theme and access the sign out option.

USERS & ACCOUNTS MENU ITEMS

Menu Item	Description
Groups and Roles	Access to managing user groups and roles within Lighthouse.

Menu Item	Description
Local Authentication Policy	Manage the Lighthouse password policy as well as login restrictions.
Local Users	Access to managing the local Lighthouse users, which user group they are linked to and the inherited permission sets applied from the user groups.
Remote Authentication	Configure and manage settings for remote authentication. Lighthouse supports local users only or remote authentication via Radius, TACACS+ and LDAP servers.
SSH Authentication	Access to configuring the SSH for password authentication and connection as Root User.

⚙ SETTINGS MENU ITEMS

FILTERS AND TAGS

Menu Item	Description
Filters	Manage filters for nodes, ports and resources. The filters are selectable in the main dashboards.
Tags	Manage tags that can be applied for nodes and resources.

MULTI INSTANCE

Menu Item	Description
Multi Instance VPN	Access to setup the range for VPN tunnels between the primary and Secondary

MULTI INSTANCE	
Menu Item	**Description**
	Lighthouses.
Secondary Lighthouse	Manage secondary Lighthouses.

SERVICES	
Menu Item	**Description**
Alerting and Monitoring	Manage setup and configuration for SNMP and SYSLOG services.
Cell Health Reporting	Enable and configure cell health reporting signal quality range and frequency of reporting.
Console Gateway	Configure the port delimiter and SSH address for the console gateway.
HTTPS Certificate	Manage the Lighthouse SSL certificate.
Lighthouse VPN	Setup the range for VPN tunnels between the Primary Lighthouse and Nodes, as well as the Smart Management Fabric network range.
Node Backup	Enable backup and storage of the node backups.
Smart Management Fabric	Enable functionality utilizing dynamic routing protocols.

SYSTEM	
Menu Item	**Description**
Backup, Restore and Reset	Backup and restore Lighthouse configurations, and access to performing a reset of the system to factory defaults.
Network Interfaces	Manage and configure the available network interfaces.
Network Settings	Setup the network settings and manage any external IP addresses or DNS names of the system.
Session Settings	Access to managing idle timeout periods for both Web UI and CLI sessions.
Subscriptions	Manage available subscriptions and node assignments.
System Upgrade	Perform a system upgrade.
Time Settings	Manually configure timezone information or set to automatic and configure the Network Time Protocol servers.

PERSONALIZATION MENU ITEMS

Menu Item	Description
Theme	Select between Light, System and Dark themes.
Sign Out	Sign out of the Lighthouse session.

MENU PANE

The navigation pane is used to access the following:

	Menu Section	Description
⌂	Dashboard	The Dashboard provides an overview of the Lighthouse instance showing current status of the nodes and the cell health.
📄	Jobs	Provides an overview of currently running jobs and access to those jobs previously completed.
🗄	Nodes	Displays a grid of nodes either enrolled in Lighthouse or are pending enrollment.
🔌	Ports	Dashboard showing the status of all ports connected to node for which you have permissions to view/edit.
💻	Resources	Access to managing connected resources.
>_	Terminal	Access to the Lighthouse local web-based terminal.
⚙	Node Tools	Expand this menu to manage config templates, enrollment bundles and firmware updates.
🖧	NetOps	Expand this menu for access to components that enable automation of specific operational scenarios deployed as Docker containers.

NODE TOOLS MENU ITEMS

The Node Tools menu section provides access to the following:

Menu Item	Description
Config Templates	Create and manage config templates to push to the nodes.
Enrollment Bundles	Create and manage enrollment bundles.
Enrollment Settings	Manage the token nodes use to request an enrollment and the default subscript to use for call home node enrollments.
Firmware Upgrade	Schedule and manage firmware updates.

NETOPS MENU ITEMS

The NetOps menu section provides access to the following:

Menu Item	Description
Manage NetOps Modules	Manage and redeploy NetOps Modules.
NetOps Installation	Install NetOps modules.
IP Access	Manage settings for the IP Access NetOps module.
Automation Gateway	Manage your network using automation tools.

HELP AND SYSTEM INFORMATION PANE

The help and system information pane provides access for the following:

⊕	Enroll Node	Shortcut to enroll a node. Selecting this displays the ENROLL NODES dialog.
?	Help	Expand this menu to access the help options.
☰	System Information	Display the system information panel that shows the software and API version information, and the licensing subscription summary. Selecting *View Details* displays the subscriptions dashboard.

? HELP MENU ITEMS

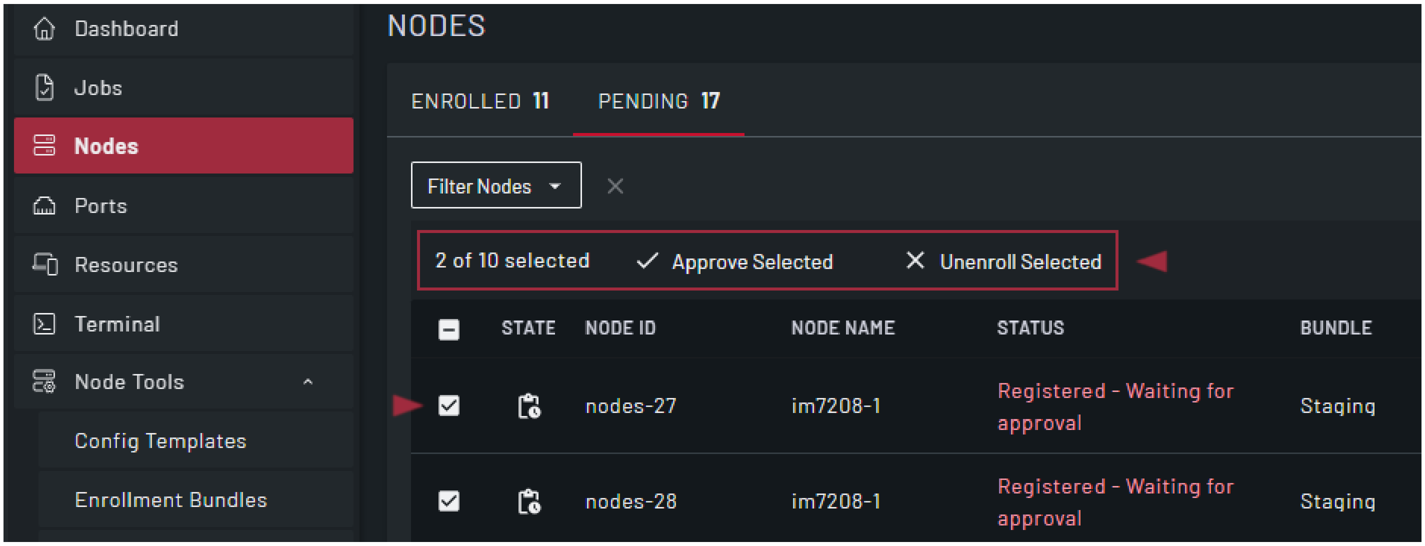
Menu Item	Description
Generate Technical Support Report	Create and download a report to assist the support teams.
View User Manual	Link to the latest published Lighthouse user documentation.
Visit Support Website	Link to technical support contact information as well as the Opengear Customer Portal resources.

GRIDS AND CONTROLS

On some pages in Lighthouse, data is displayed in grids. You can select items, sort the information, and filter the data displayed.

MULTI-SELECT AND ACTIONS

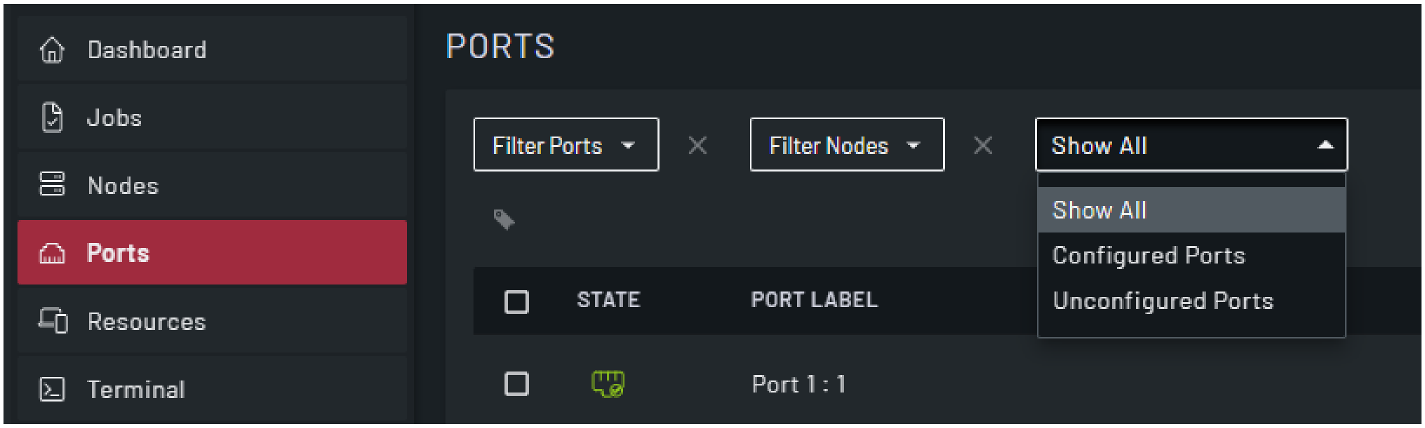
Select check boxes to the left of the rows to take action on multiple items at the same time. The available actions are displayed above the top row of the grids. If you click the check box at the top, it selects all check boxes under it.



☐	STATE	NODE ID	NODE NAME	STATUS	BUNDLE
☑		nodes-27	im7208-1	Registered - Waiting for approval	Staging
☑		nodes-28	im7208-1	Registered - Waiting for approval	Staging

FILTERING THE DATA

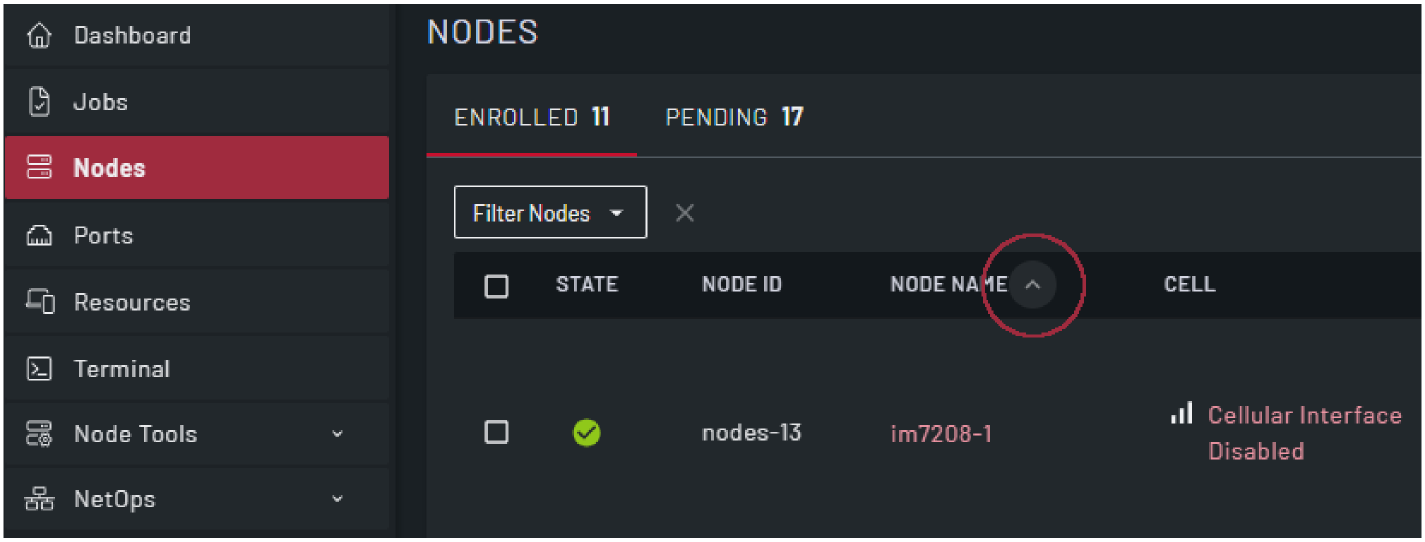
Select the Filter control to select and manage custom filters.



☐	STATE	PORT LABEL
☐		Port 1 : 1

SORTING DATA ON A GRID COLUMN

Some columns support sorting the values in the column in ascending or descending order. Hover over the column name to show the sort control. Click to modify the column sort order.



□	STATE	NODE ID	NODE NAME	CELL
□	✓	nodes-13	im7208-1	Cellular Interface Disabled

HOW TO SEARCH

Enter a term in the search control to filter the results. The AND operator is used if more than one search term is entered. To enter a multi word search phrase, enclose them in double quotes.

INSTALLING LIGHTHOUSE

To install Lighthouse you require:

- A virtual machine (VM) that can support a 50GB disk at minimum.
- The correct image file.

LIGHTHOUSE VIRTUAL MACHINES

Lighthouse VM is available in several formats on our secure ftp sites, from where you can download and verify the checksums and install the appropriate files. Ensure you use the correct file for your upgrade.

Note: SHASUMS files to verify the download are available with the install packages.

LIGHTHOUSE VM FILES

Filename Format	Description
`lighthouse-<year>-<month>-<version>-ovf.zip`	An Open Volume Format file inside a PKZIP archive. This is for use with virtual machine managers such as KVM and Virtual Box.
`lighthouse-<year>-<month>-<version>-vmx.zip`	A VMware configuration file inside a PKZIP archive. This is for use with virtual machine managers from VMware.
`lighthouse-<year>-<month>-<version>.ova`	An Open Virtual Appliance file. 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.
`lighthouse-<year>-<month>-<version>.lh_upg`	An upgrade file.

LIGHTHOUSE AWS VM FILES

Filename Format	Description
`lighthouse-<year>-<month>-<version>.aws.lh_upg`	An upgrade file.
`llighthouse-<year>-<month>-<version>.aws.raw.tar`	AWS deployment file.
`lighthouse-aws-bootstrap.sh`	A shell script for deploying on AWS.

LIGHTHOUSE AZURE VM FILES

Filename Format	Description
`lighthouse-<year>-<month>-<version>>.azure.lh_upg`	An upgrade file.
`lighthouse-<year>-<month>-<version>.azure.zip`	Azure deployment files.

LIGHTHOUSE HYPER-V VM FILES

Fileilename Format	Description
`lighthouse-<year>-<month>-<version>.hyperv.zip`	Hyper-V deployment file.

INSTALL LIGHTHOUSE VM ON VMWARE

This section describes how to install Lighthouse VMs on VMware hosts including:

- VMware vSphere 6.0 client on Windows
- VMware Workstation Player on Windows
- VMware Workstation Pro on Windows

VMWARE VSPHERE 6.0 CLIENT ON WINDOWS

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

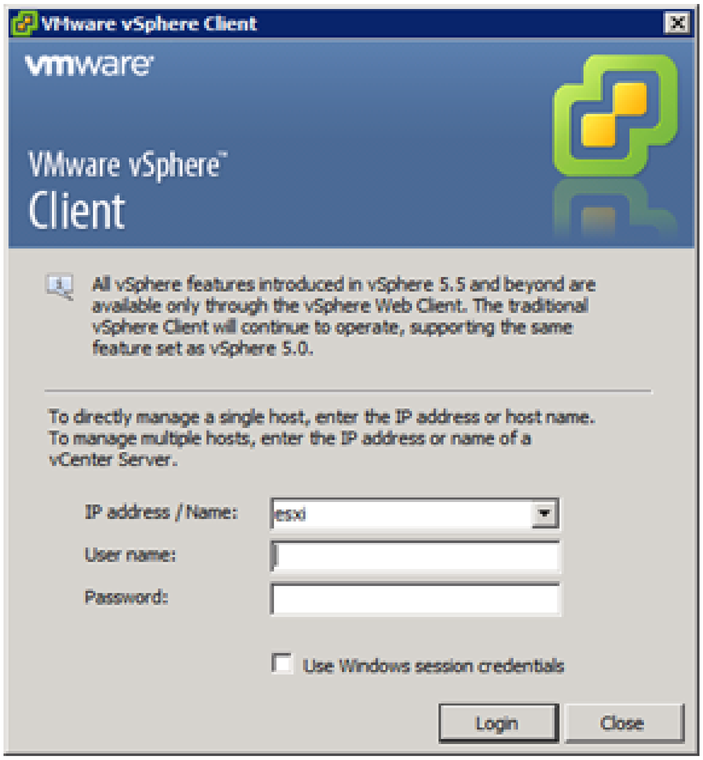
REQUIREMENTS

Ensure the following preconditions are met before you start installation:

- VMware vSphere 6.0 is installed and running on available hardware.
- Access to a Windows computer on which the VMware vSphere 6.0 client is installed.
- The installed client application must be able to connect to and manage the VMware vSphere 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.

LAUNCH THE VSPHERE CLIENT AND CONNECT TO A VSPHERE INSTANCE.

1. Launch the VMware vSphere Client: **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

User name:

Password:

□ Use Windows session credentials

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 **Enter**.

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

   Connecting

   Loading inventory

   Loading main form

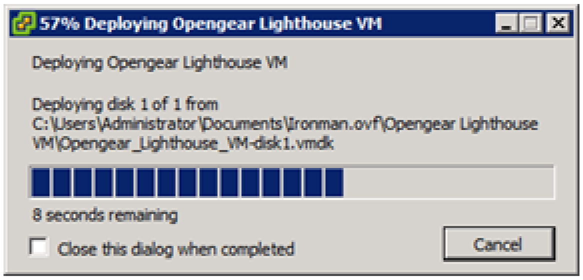
   Displaying main form

   The vSphere main form window opens.

IMPORT THE LIGHTHOUSE VM OPEN VOLUME FORMAT IMAGE

To import the Lighthouse VM:

1. From the vSphere Client menu bar, choose **File > Deploy OVF Template**.
   The **Deploy OVF Template** window displays, with the first stage, **Source**, pre-selected.
2. If the file `Opengear Lighthouse VM.ovf` is on a remote computer via a URL, enter the URL in the **Deploy from a file or URL** field. Otherwise, click **Browse**. An Open dialog displays.
   a. Navigate to the directory containing the file `Opengear Lighthouse VM.ovf`.
   b. 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 displays, showing basic information about the Lighthouse VM encapsulated by the `.ovf` file. Click **Next**.
5. The **Name and Location** screen displays 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 displays, 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**.
   The **Deploying Opengear Lighthouse VM** progress dialog displays.



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

When deployment is finished, the **Deployment Completed Successfully** alert displays.

9. Click **Close**.

   The new virtual machine is now deployed and displays in the inventory list.

LAUNCH LIGHTHOUSE

The vSphere Client provides several ways of launching a Virtual Machine hosted on a vSphere instance. Begin by selecting the **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 run it as a headless system, that is, with no display on a monitor. However, you can access Lighthouse via the web UI or SSH.

ACCESS THE CONSOLE OF A RUNNING BUT HEADLESS 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 inventory list for the vSphere Client, then do one of the following:

- Select **Inventory > Virtual Machine > Open Console**.
- Right-click and select **Open Console** from the context menu that displays.

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-<year>-<month>-<version>-vmx.zip` archive to **C:\Users\%USERNAME%\Virtual Machines**.
2. Right-click the archive and select **Extract All** from the contextual menu.
   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.
3. Uncheck the **Show Extracted Files When Complete** checkbox and then click **Extract**.
   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 on it to add the Lighthouse virtual machine to the VMware Workstation 12 Player virtual machines list. If more than one file displays, double-click **Lighthouse.vmx**.
   The Lighthouse virtual machine is added to the VMware Workstation 12 Player virtual machines list.
5. With **Opengear Lighthouse VM** selected in the VMware Workstation 12 Player virtual machine list, click **Play Virtual Machine** to boot Lighthouse.

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 required.

IMPORT THE LIGHTHOUSE VM FILE INTO VMWARE WORKSTATION PRO

Step 1. Preparation:

This step prepares the Lighthouse VM file for import into VMware Workstation Pro.

1. Move the `lighthouse-<year>-<month>-<version>-vmx.zip` archive to **C:\Users\%USERNAME%\Virtual Machines**.
2. Right-click the `lighthouse-<year>-<month>-<version>-vmx.zip` archive and select **Extract All** from the contextual menu.

   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.
3. Uncheck the **Show Extracted Files When Complete** checkbox and then click **Extract**.

   A folder called lighthouse is created inside **C:\Users\%USERNAME%\Virtual Machines**.

Step 2. Import Lighthouse VM file:

1. Click **Open a Virtual Machine**.
2. Navigate to **C:\Users\%USERNAME%\Virtual Machines\lighthouse\**.

   VMware Workstation Pro points to **Libraries > Documents** and this library includes **C:\Users\%USERNAME%\My Documents**.
3. Double-click **Virtual Machines** and then double-click **Lighthouse**.
4. If only one file, Lighthouse, displays, double-click the file to add the Lighthouse virtual machine to the VMware Workstation Pro virtual machines list. If more than one file displays, double-click **Lighthouse.vmx**.

   The Lighthouse virtual machine is added to the VMware Workstation Pro virtual machines list.
5. 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.

INSTALL LIGHTHOUSE VM ON HYPER-V ON WINDOWS

This section describes how to install Lighthouse VMs on Hyper-V on Windows:

- Hyper-V running on Windows 10 or Windows Server 2016.
- VirtualBox deployments.

LOCAL DEPLOYMENT ON HYPER-V

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

1. Unzip `lighthouse-<year>-<month>-<version>-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.

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 (or later) host machine and the required Zip `lighthouse-<year>-<month>-<version>-hyperv.zip` is in **C:\Users\%USERNAME%$\Downloads**. Windows 10 is already configured to manage Hyper-V on Windows Server 2016.

> **Note:** 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-<year>-<month>-<version>-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.

VIRTUALBOX DEPLOYMENTS

Lighthouse can be installed on the following hosts:

- VirtualBox on Windows as host
- VirtualBox on macOS as host
- VirtualBox on Ubuntu as Host
- VirtualBox on Fedora Workstation as host

VIRTUALBOX ON WINDOWS AS HOST

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-<year>-<month>-<version>-ovf.zip` is in **C:\Users\%USERNAME%$\Downloads**.

1. Unzip **lighthouse-ovf**.
   It may appear as `lighthouse-<year>-<month>-<version>-ovf.zip` depending on the Windows Explorer preference settings.
2. Right click the **lighthouse-ovf** archive and select **Extract** all from the context menu.
   The **Select a Destination** and **Extract Files** dialog opens. The destination is **C:\Users\%USERNAME%\Downloads\Lighthouse-ovf**.
3. Uncheck the **Show extracted files when complete** checkbox and edit the destination by removing **Lighthouse-ovf** from the path.
4. Click **Extract**.
   A folder called **lighthouse-ovf** is created inside **C:\Users\%USERNAME%\Downloads\**.
5. Launch VirtualBox.
   The Oracle VM VirtualBox Manager window displays.
6. Choose **File > Import Appliance**.
   The **Appliance to Import** dialog opens.
7. Click **Expert Mode**.
   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**.
   The **Open File** dialog displays with **C:\Users\%USERNAME%\Documents** as the current folder.
9. Navigate to **C:\Users\%USERNAME%\Downloads\Lighthouse.ovf\Opengear Lighthouse VM**.
10. Select the file **Opengear Lighthouse VM** and click **Open**.
11. Double-click the text **vm** in the **Name** row and **Configuration** column to make it editable.

12. Type **Opengear Lighthouse VM** and press **Enter**.
13. Click **Import**.

    A new virtual machine, called **Opengear Lighthouse VM** is added to the list of virtual machines available to **Virtual Box**.
14. Select **Opengear Lighthouse VM** from the list.
15. Select **Machine > Settings** or click the **Settings** icon in the **VirtualBox Manager** toolbar or press **Control+S**.

    The **Opengear Lighthouse VM - Settings** dialog displays.
16. Click the **System** option in the list of options running down the left-hand side of the dialog.

    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 **Enter**.
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.

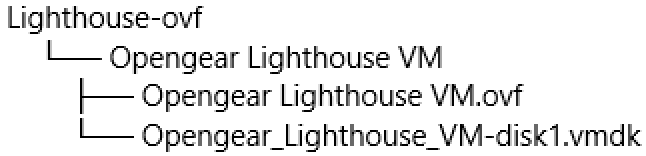
> **Note:** Selecting the **Hardware Clock in UTC Time** checkbox is necessary because Lighthouse expects the hardware clock to be set to UTC, not local time. Unlike other Virtual Machine Managers, Virtual Box both exposes this option as a user-adjustable setting and does not set it to UTC by default.

VIRTUALBOX ON MACOS AS HOST

VirtualBox should already installed on the host macOS machine and the required PKZIP archive, `lighthouse-<year>-<month>-<version>-ovf.zip` is in **~/Downloads**.

1. Unzip `lighthouse-<year>-<month>-<version>-ovf.zip`.

   The folder **Lighthouse-ovf** is created in **~/Downloads** and 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 displays.

3. Select **File > Import Appliance** or press **Command+I**.

   The **Appliance to Import** dialog sheet slides down from the Oracle VM VirtualBox Manager toolbar.

4. Click **Expert Mode**.

   The **Appliance to Import** dialog sheet 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 sheet slides down from the **Oracle VM VirtualBox Manager** toolbar.

   This sheet opens with **~/Documents** as the current folder.

6. Navigate to **~/Downloads/Lighthouse.ovf/Opengear Lighthouse VM/**.

7. Select **Opengear Lighthouse VM** and click **Open**.

   Depending on the **Finder Preferences** settings, the file may present as **Opengear Lighthouse VM.ovf**.

8. Double-click the text vm in the **Name** row and **Configuration** column to make it editable.

9. Type **Opengear Lighthouse VM** and press **Enter**.

10. Click **Import**.

    A new virtual machine, called Opengear Lighthouse VM is added to the list of virtual machines.

11. Select **Opengear Lighthouse VM** from the list.

12. Choose **Machine > Settings**. Or click the **Settings** icon in the **VirtualBox Manager** toolbar. The **Opengear Lighthouse VM Settings** dialog displays.
13. Click the **System** option in the dialog's toolbar. 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.
14. In the **Motherboard** tab, select the **Hardware Clock in UTC Time** checkbox.
15. Click **OK** or press **Enter**.
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.

Notes:

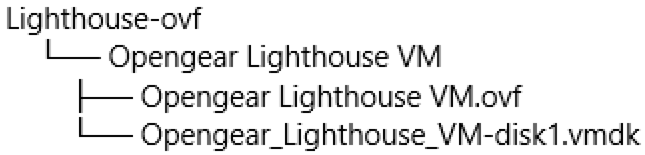
- Selecting the Hardware Clock in UTC Time checkbox is necessary because Lighthouse expects the hardware clock to be set to UTC, not local time. Unlike other Virtual Machine Managers, Virtual Box both exposes this option as a user-adjustable setting and does not set it to UTC by default.
- By default,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

VIRTUALBOX ON UBUNTU AS HOST

Before you begin the procedure, make sure that VirtualBox and all required support files are installed on the host machine and the PKZIP archive, **-ovf.zip**, is in **~/Downloads**.

1. Unzip `lighthouse-<year>-<month>-<version>-ovf.zip`.

   The folder **Lighthouse-ovf** is created in **~/Downloads** and 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 displays.

3. Choose **File > Import Appliance**.

   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 positioned to the right of the **Appliance to Import field**.

   A file navigation dialog, **Choose a Virtual Appliance to Import**, opens with **~/Documents** as the current folder.

6. Navigate to **~/Downloads/Lighthouse.ovf/Opengear Lighthouse VM/**.

7. Select **Opengear Lighthouse VM.ovf** 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 press **Enter**.

10. Click **Import**.

    A new virtual machine, called **Opengear Lighthouse VM** is added to the list of virtual machines available to Virtual Box.

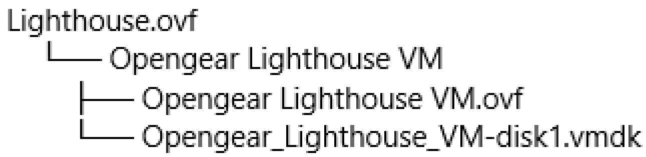
11. Select **Opengear Lighthouse VM** from the list and click **Start** in the **Oracle VM VirtualBox Manager** toolbar to boot Lighthouse. Double-clicking **Opengear Lighthouse VM** in the list also boots Lighthouse.

> **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.

VIRTUALBOX ON FEDORA WORKSTATION AS HOST

Before you begin, make sure that VirtualBox and all required support files are already installed on the host machine and the PKZIP archive, **-ovf.zip** is in **~/Downloads**.

1. Unzip `lighthouse-<year>-<month>-<version>-ovf.zip`.
   The **Lighthouse.ovf** folder is created in **~/Downloads** and 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 displays.
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 press **Enter**.
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.

INSTALL LIGHTHOUSE VM ON LINUX HOSTS

This section describes how to install Lighthouse VMs on the following Linux hosts:

- Ubuntu
- Fedora Workstation
- RHEL

VIRTUAL MACHINE MANAGER ON UBUNTU AS HOST

Virtual Machine Manager and all required support files should be installed on the host machine.

1. Expand `lighthouse-<year>-<month>-<version>-raw.hdd.tar`.
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. Select the file `lighthouse-<year>-<month>-<version>-raw.hdd` and click **Open** in the top right-corner of the dialog.

   A **Review** window opens providing basic information about the virtual machine or box, as Boxes calls them, to be created.
6. Click **Create** in the top right corner of the **Review** window.

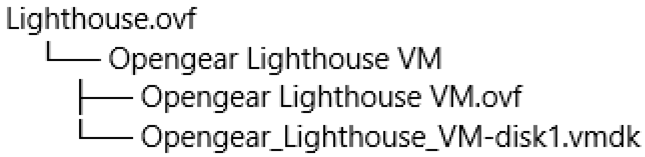
   A new virtual machine instance, **Opengear_Lighthouse_VM-disk1**, is created and presented in the **Boxes** window.

BOXES ON FEDORA WORKSTATION AS HOST

Boxes and all required support files should be installed on the host machine and `lighthouse-<year>.<month>.<release number>-ovf.zip` is in **~/Downloads**.

1. Unzip `lighthouse-<year>-<month>-<version>-ovf.zip`.

   The **Lighthouse.ovf** folder is created in **~/Downloads** and 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.

   A new virtual machine instance, **Opengear_Lighthouse_VM-disk1** is created and presented in the **Boxes** window.

8. To rename the virtual machine instance, right-click on the machine instance and choose **Properties** from the contextual menu that displays. Click anywhere in the **Name** field to select and edit the name. Click **Close** to save the changes.

BOXES ON RHEL AND COMPATIBLE DISTRIBUTIONS

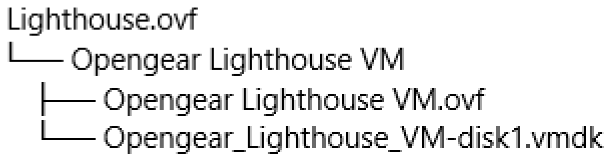
CentOS should be installed, complete with the Gnome desktop environment as the host operating system. CentOS includes the full complement of KVM-centric virtualization tools including the GUI-based virtualization management tools **Boxes** and **virt-manager** and the shell-based virtualization management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

This procedure assumes **Boxes** is used to setup and manage the Lighthouse VM and that the required PKZIP archive, `lighthouse-<year>-<month>-<version>-ovf.zip` is in **~/Downloads**.

To install Lighthouse on CentOS:

1. Unzip `lighthouse-<year>-<month>-<version>-ovf.zip`.

   The **Lighthouse.ovf** folder is created in **~/Downloads** and 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.

INSTALL IN THE CLOUD

This section describes how to install Lighthouse on the following supported cloud environments:

- Azure
- AWS

INSTALL IN AZURE

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 with at least 50GB storage space.
4. Navigate to the newly created storage account, click **Containers** under **Data Storage** and create a new blob container.

5. Download a Lighthouse VHD image, the latest Lighthouse image can be found in a zip file at the following URL: https://ftp.opengear.com/download/lighthouse_software/current/lighthouse/azure.
6. Copy the Lighthouse VHD image into the Azure storage container. (Using AzCopy is recommended, as the VHD image is large and the upload can take a long time to complete through the Microsoft Azure portal.)
   a. If you haven't already, install AzCopy following instructions provided by Microsoft. Click here to read the instructions on Microsoft's website.
   b. Generate a SAS token to use in your AzCopy commands
      i. While viewing the newly created storage container, click **Access policy**.
      ii. Click **Add policy** under Stored access.
      iii. Enter the *Identifier*.
      iv. Set the Permissions and select *Read, Write and Create* from the *Permissions* drop-down.
      v. Set a valid *Start Date* and *Expiry Date*.
      vi. Click **Okay**.
      vii. Click on the **Shared access** token.
      viii. Select the policy *Identifier* (as created from the previous steps) as *Stored access policy*.
      ix. Click **Generate SAS and URL**.
      x. Copy the *Blob SAS token* and *Blob SAS URL*.

      > **Note:** The *Blob SAS token* and *Blob SAS URL* must be copied at this point in the process as you will not be able to view the information again after this step.

   c. Copy the Lighthouse VHD image into the Azure storage container using the following format, make sure to fill in the path to your local Lighthouse VHD image and your Blob SAS URL generated during the previous step: `./azcopy copy <path_to_local_image_file>`

```
"<blob_sas_url>"
```

> **Note:** A SAS token can also be created using Azure CLI. Click here to read the instructions on Microsoft’s website.

7. Create an image:
   a. In the Azure Portal, under *Azure Services*, click the **Images** icon.
   b. Click **Create** to create a new image, make sure that the location is the same as your storage account, the **OS type** is set to *Linux* and **VM generation** is set to *Gen 1*.
   c. Click **Browse** on the *Storage blob* field and select the Lighthouse VHD file you uploaded during a previous step.
   d. Click **Create** to create the image.
8. Go to the newly created image and click **Create VM**. Ensure the selected image is correct.
9. Choose the required virtual machine instance size.
10. Enter the details for the Microsoft Azure admin user with either password OR SSH key authentication.

    > **Note:** If SSH key authentication is selected, the user will be created without a password and will be unable to access the UI.

11. Select the inbound ports enabled for the Lighthouse instance (SSH, HTTPS).
12. Navigate to the next page of configuration (Disks) and select the required storage option for the boot disk.
13. Go to the **Review** page and after validation passes, click **Create**.
14. Go to the **Virtual Machines** page, select the virtual machine and open the Serial Console. Lighthouse should now be deploying on Microsoft Azure.

15. To allow nodes to enroll in Lighthouse, you must add the following firewall rules on the **Networking** page under **Settings** on the virtual machine you deployed:
    1. Add a rule to allow UDP connections from any source to port 1194 on the instance's internal network address (10.0.0.x).
    2. Add a rule to allow UDP connections from any source to port 1195 on the instance's internal network address (10.0.0.x).
    3. HTTPS and SSH should already be allowed from the initial setup If not, add them.
    4. Other ports may require to be opened, depending on feature usage. For example:
        1. SNMP (UDP/161 or TCP/161) – SNMP Management
        2. OpenVPN (UDP/1195) – Lighthouse Multiple Instance VPN
        3. HTTPS (TCP/8443) – Alternate REST API port
16. Confirm that the Azure instance public IP address has been added to external endpoints in **Settings > System > Network Settings**.

SET A PASSWORD ON LIGHTHOUSE VIA SSH

If you are logged into Lighthouse via SSH keys, you must set a password to login via GUI. Use the `ogpasswd` utility to do this.

```
ogpasswd -u lh_admin -p MySecretPassword
```

> **Note:** Your username must be the same as the Microsoft Azure admin user created in step 10.

INSTALL IN AMAZON WEB SERVICES

Before you perform any procedures on this page, ensure that you have an account on AWS with an IAM user, a key pair and an access key:

- The IAM user should have, at a minimum, permissions to create, attach, delete, and snapshot EBS volumes as well as create an Amazon Machine Image (AMI).
- If you are using IAM Identity Center, you can use an IAM Identity Center user with the same permissions instead. Consult Amazon documentation for more information if required.

LAUNCH A LIGHTHOUSE FROM THE AWS MARKETPLACE

Lighthouse is available on Amazon Web Services (AWS) Marketplace. You can subscribe to get access to our published AMIs.

> **Note:** You must still contact Sales to obtain a license.

1. Go to AWS Marketplace: Opengear Lighthouse Software AMI.
2. Follow the steps to subscribe.
3. After you subscribe, navigate to your **AWS Marketplace** > **Manage subscriptions** page.
4. Under **Actions** for the Lighthouse AMI subscription, click **Launch**.
   The **Launch new instance** page displays.
5. From the **Software version** drop-down, select the latest version of Lighthouse.
6. From the **Region** drop-down, select the region.
7. Click the **Continue to launch through EC2** button.
8. Continue to Launching in Amazon Web Services.

LAUNCH A LIGHTHOUSE ON AWS FROM A MANUALLY CREATED AMI

To use Lighthouse with Amazon Web Services (AWS), you must first create an Amazon Machine Image (AMI) containing Lighthouse in the AWS region in which you want to deploy Lighthouse. A temporary Linux “build-box” EC2 instance should be used to create a private Lighthouse AMI.

> **Note:** This is a one-time procedure. The AMI can be used to create multiple instances of Lighthouse, and upgrades can be performed through the Lighthouse Web UI.

1. Create an AWS EC2 Linux instance, with the following settings:

   - Amazon Linux 2 or Amazon Linux 2023
   - `t3.large` instance type with default (8 GiB) root volume
   - 50GB `gp3` volume

   Consult Amazon documentation for more information if required.

2. Create a Lighthouse AMI, using the `lighthouse-aws-bootstrap.sh` script (usage information can be displayed by using the -h option) on the EC2 instance created in the previous step. The steps are detailed as follows:

   a. Connect via SSH to your instance on AWS using the username `ec2-user` and the private key you created previously. All subsequent steps must be performed on the instance.

   b. Configure AWS using the following command:

   ```
   aws configure
   ```

   c. Provide the access key and region details (other settings may be left unchanged). If you are using IAM Identity Center, you must instead configure using `aws configure sso`, and set the CLI Profile Name to be `default`.

   d. Download the aws-bootstrap script:

   ```
   wget http://ftp.opengear.com/download/lighthouse_
   software/current/lighthouse/aws/lighthouse-aws-bootstrap.sh
   ```

   e. Run the lighthouse-aws-bootstrap.sh script as follows:

   ```
   bash ./lighthouse-aws-bootstrap.sh -n Lighthouse -r
   https://ftp.opengear.com/download/lighthouse_
   software/current/lighthouse/aws/lighthouse-<year>-<month>-
   <version>.aws.raw.tar
   ```

f. Wait while the Lighthouse AMI is created. This can take some time (up to 30 minutes).

g. After the AMI has been created, terminate the Linux EC2 instance to avoid incurring additional costs.

3. Continue to Launching in Amazon Web Services.

RUN THE BOOTSTRAP SCRIPT - EXAMPLE

```
$ bash ./lighthouse-aws-bootstrap.sh -n Lighthouse -r \
> http://ftp.opengear.com/download/lighthouse_
software/current/lighthouse/aws/lighthouse-24.06.02.aws.raw.tar
Downloading image...
Image size is 54049899008 bytes (51 GiB)
Creating volume...
Attaching volume vol-09fb0b463f5a59eaf to EC2 instance...
Cloning image onto volume...
0+852971 records in
0+852971 records out
54049899008 bytes (54 GB, 50 GiB) copied, 845.072 s, 64.0 MB/s
Creating snapshot of volume...
Waiting for snapshot snap-0f83746856d985070 to complete...
Creating AMI from snapshot snap-0f83746856d985070...
Done!
Cleaning up...
```

CHANGES TO DEFAULT SETTINGS

> **Caution:** With *Lighthouse version 24.06.0 and later*, changes have been introduced related to the root user and SSH Password Authentication default settings that impact any newly launched Lighthouse AWS instances.

Changes impacting the *root user*:

- For security purposes the `root` user is *disabled* by default. The root user can be enabled by going to 👥 **> USERS & ACCOUNTS > Local Users**.
- A new user called `lhadmin` is provided that should be used for initial configuration.
- SSH connection for `root` is also *disabled* by default. To enable, navigate to 👥 **> USERS & Accounts > SSH Authentication**.
- Ensure to run all commands with appending `sudo`.

Changes impacting *SSH Password Authentication*:

- SSH Password Authentication is disabled by default.
- User accounts require a Public SSH Key associated with the account.
- Users will use their Private SSH key to connect via SSH.
- The `lhadmin` user will default with the AWS key pair that was used to create the Lighthouse Instance.
- To associate a Public SSH Key to the account navigate to 👥 **> USERS & ACCOUNTS > Local Users** and add the SSH Authentication Key to the user.

LIMITATIONS

AWS support is currently 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 `lhadmin` user.
- Running custom scripts on startup (see above).
- Providing a `lhadmin` password via userdata (see above).

At this time Lighthouse does not support:

- Using AWS database services.
- Using AWS redis services.
- Using any of AWS scalability functionality.

Note: If you want to deploy Lighthouse across different AWS regions, an AMI is required in each region. Amazon supports copying AMIs between regions and offers a walkthrough of the necessary steps to do this.

LAUNCH IN AMAZON WEB SERVICES

LAUNCH A LIGHTHOUSE INSTANCE ON AWS

When the Lighthouse AMI is created, it displays in the Amazon Machine Images (AMIs) section of the EC2 Management Console.

Note: If you are launching from AWS Marketplace, you can start at *Instance Type*.

To create a new Lighthouse EC2 instance:

1. Select the Lighthouse AMI.
2. Click **Launch instance from AMI**.

Instance Type

Lighthouse should run on a general purpose instance type, such as M5.

> **Note:** If an instance type that supports "burstable" CPU such as T2 is used, ensure that unlimited CPU is selected, to avoid operational problems caused by CPU throttling.

Key Pair

EC2 requires a key pair to be specified when launching instances.

Network Settings

A security group should be created. Lighthouse requires some ports to be open:

- SSH (TCP/22) – Secure Shell. Access should be limited to just your corporate network.
- HTTPS (TCP/443) – Lighthouse Web UI and REST API. This is used by both web browsers and nodes. For example, call-home enrollment.
- OpenVPN (UDP/1194) – Lighthouse VPN. This is used to communicate with nodes after they are enrolled.
- Other ports may be required to be opened, depending on feature usage. For example
    - SNMP (UDP/161) – SNMP Management
    - OpenVPN (UDP/1195) – Lighthouse Multiple Instance VPN
    - HTTPS (TCP/8443) – Alternate REST API port

Storage

By default, the root volume will be around 53 GiB. This may be sufficient, depending on your intended usage. It is easier to specify more storage now, but more can be added later.

Advanced Details

An initial `lhadmin` password must be set in the `UserData` section.

```
password=topSecretPassword123
```

If the user does not specify the `lhadmin` password in the Advanced Details section they can set the `lhadmin` password using the `ogpasswd utility`.

SET A PASSWORD FOR THE LHADMIN USER ON LIGHTHOUSE

If you are logged into Lighthouse via SSH keys, you will must set `lhadmin` password to login via GUI. Use the "ogpasswd" utility to do this.

```
ogpasswd -u lhadmin -p MySecretPassword
```

FINAL STEPS

When done, the EC2 Linux instance can be shut down and removed or saved for creating future instances.

ADD 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.

Before you add disk space:

- Ensure you take a backup of Lighthouse
- In the case of a multiple instance Lighthouse installation, consider upgrading all instances, not merely the primary instance.

ADD A NEW DISK TO AWS

Launch a 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 Lighthouse:

1. In the AWS web console, go to **Volumes** and create a new 50GB volume in the same availability zone as your LH instance.
2. After 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).

ADD A NEW DISK - 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 make sure that the new disk is specified last, otherwise your disk will appear out of order when you boot the Lighthouse.

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

ADD A NEW DISK - AZURE

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

ADD A NEW DISK - HYPER-V

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 requirements and environment.

9. After you've created the disk, click **Apply** in the VM settings window.
10. Restart the Lighthouse.

ADD A NEW DISK - VIRTUALBOX

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 Lighthouse.

MOUNT 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 must 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/au`. Any other filesystems should be mounted within the filesystem mounted here. The UUID can be obtained by running `blkid`. This will output the UUID's of all the extra hard disks on the system. When referencing the UUID, ensure the entire UUID is enclosed within quote marks like this:

```
"UUID=33464920-f54f-46b6-bd84-12f76eeb92da"
```

else the command will not run correctly.

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 "UUID=33464920-f54f-46b6-bd84-12f76eeb92da"{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
```

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 partition.
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:

   ```
   pvcreate /dev/sdb1
   ```

   > **Note:** Assuming you are now using `/dev/sdb1` that `/dev/xvdb1` will now be mapped to /dev/sdb1 so make sure you use 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".

SETTING UP LIGHTHOUSE

This section describes how to set up Lighthouse to monitor and manage your network.

LOAD LIGHTHOUSE

This section describes the initial setup stages for a new Lighthouse VM, from login, to setting external addresses, to setting the clock.

When you first load Lighthouse, a warning message displays because of the default self-signed certificate. You can ignore this at first load, and later install your own valid certificate using ⚙ **> SERVICES > HTTPS Certificate** to remove the warning message.

LIGHTHOUSE IP ADDRESSES

When the Lighthouse VM is booted and running, it can be reached via the following options:

- The static address, **192.168.0.1**.
- 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.

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 IPv4 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
```

FIRST BOOT OF THE LIGHTHOUSE VM

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

During boot, two screens open.

1. The first 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 displays:

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

202406.02
```

2. The final step in the initial setup displays:

```
To complete initial setup, please set a new root password.

Press ENTER to continue.
```

3. After pressing **Enter**, a prompt displays:

```
Enter new root password:
```

4. Enter a password and press Enter.

> **Note:** It is recommended that 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.

> **Tip:** Keep in mind that non-US-English keyboards are not supported in the graphics console.

5. The confirm prompt displays:

```
Confirm given password
```

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

```
lighthouse login:
```

7. Enter root and press Enter. A password prompt displays:

```
Password:
```

8. Enter the newly-set password and press **Enter**. A standard bash shell prompt displays 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:~#
```

SET THE LIGHTHOUSE HOSTNAME

To set the hostname for a running Lighthouse instance:

1. Select ⚙ **> SYSTEM > Network Settings**.
2. Edit the **Hostname** as required.

   > **Note:** Hostnames must follow the naming conventions:
   >
   > - ASCII alphanumerical characters plus '-'.
   > - Maximum 64 characters.

3. Edit the **SSH Port**.
4. Click **Apply**.

ADD EXTERNAL IP ADDRESSES MANUALLY

Adding a Lighthouse instance’s external IP address or addresses manually to a Lighthouse instance’s configuration is an optional step. In general, these should not be changed except by a network support engineer.

> **Note:** IP addresses can be IPv4, IPv6, or DNS names.

To add external addresses to the configuration for a Lighthouse instance:

1. Select ⚙ **> SYSTEM > Network Settings**.
2. Click ⊕ **Add External Network Address**.

   A dialog displays.
3. Enter the **External Endpoint address**.
4. 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.
5. Click **Add Address**.

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

1. Use the ↓ ↑ icons 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 order configured.

LIGHTHOUSE SSL CERTIFICATE

Lighthouse ships with a private SSL Certificate that encrypts communications between it and the browser. Most browsers display a warning message when first trying to access Lighthouse.

> **Note:** If you plan to use the Lighthouse **Multiple Instance** feature, the certificate is used on all instances. In this case, we recommend using a wildcard certificate.

To examine this certificate or generate a new **Certificate Signing Request**:

1. Select ⚙ **> SERVICES > HTTPS Certificate**.

   The details of the **Current SSL Certificate** appear.

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

SET THE LIGHTHOUSE INTERNAL CLOCK

Lighthouse and Node system times must be in sync. Enrollment can fail if there is a significant difference between the Lighthouse and the node. It is recommended that you use an NTP server to automatically manage the date and time.

If you are using multiple instances, configure the time zone for the secondary instances before you add them as secondary instance. The only way to change the time zone after adding a secondary instance is to use the CLI.

MANUAL CONFIGURATION

To manually select the time zone:

1. Select ⚙ **> SYSTEM > Time Settings**.

   Timezone options are displayed.
2. Select the timezone for the Lighthouse instance from the **Time Zone** drop-down list.
3. Click **Apply**.

AUTOMATIC CONFIGURATION

To set time automatically:

> **Note:** It is highly recommended that you use a NTP Server to automatically manage date and time.

1. Select ⚙ **> SYSTEM > Time Settings**.
2. Click AUTOMATIC to display options.

3. Select **Enabled**.
4. Click ⊕ **Add server**.
5. Enter a working NTP Server address in the **NTP Servers** field.
6. Click **Apply**.

SET UP NETWORKING REQUIREMENTS

This section outlines the basic steps to setup networking requirements for Lighthouse including:

- Lighthouse Session Settings
- MTU of the Lighthouse VPN tunnel
- Network connection
- SNMP Manager Settings
- SNMP Service
- Cellular Health Settings
- Lighthouse MIBs
- Smart Management Fabric

EXAMINE OR MODIFY THE LIGHTHOUSE SESSION SETTINGS

To modify Web and CLI session settings:

1. Select ⚙ **> SYSTEM > Session Settings**.
2. Enter the **Web Session Timeout:**

   This value can be set from 1 to 1440 minutes.
3. Enter the **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.

4. If required, select **Enabled** for **Alternate API Port Status**.

   This port is set to 8443. 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.

EXAMINE OR CHANGE LIGHTHOUSE VPN NETWORK SETTINGS

The VPN network ranges for Lighthouse VPN (LHVPN) and Smart Management Fabric can be set up or modified.

To setup the Lighthouse VPN:

1. Select ⚙ **> SERVICES > Lighthouse VPN**.
2. Under the **VPN NETWORK RANGE** section:
   a. Enter the **Address Space**: The range to be used for the VPN.

      > **Note:** The Address Space must be in the IPv4 Private Address Space, subnet must have enough space for the existing Enrolled Nodes to Lighthouse, cannot overlap with an existing subnet across lighthouse and the base address for the network range.

   b. Enter the **CIDR Subnet Mask**: The Classless Inter-Domain Routing notation for the subnet mask for the VPN network.

   c. *Calculated Node Capacity*: An auto-calculated field with the total number of usable nodes.

   d. Enter the **Tunnel MTU**: The MTU is the size, in bytes, of the largest packet supported by a network layer protocol, including both headers and data.
3. Under the **SMART MANAGEMENT FABRIC NETWORK RANGE** section:

a. Enter the **Address Space**.

b. Enter **CIDR Subnet Mask**.

c. *Calculated Node Capacity*: An auto-calculated field with the total number of usable nodes.

d. *Smart Management Fabric MTU*: an auto-calculated field, the Smart Management Fabric MTU is based on a fixed offset off the LHVPN MTU.

NETWORK CONNECTIONS

VIEW AVAILABLE NETWORK CONNECTIONS

To see the network connections available to Lighthouse:

1. Select ⚙ **> SYSTEM > Network Interfaces**.

   The **NETWORK INTERFACES** page displays.

EDIT A NETWORK INTERFACE

To edit a network interface:

> **Note:** Editing the network settings may break connectivity.

1. Select ⚙ **> SYSTEM > Network Interfaces**.

   The **NETWORK INTERFACES** page displays.

2. Click on the interface **Name** link within the grid.

   The **UPDATE** dialog displays.

3. Make the required changes.

4. Click **Update Interface**.

Note: Do not change the **Connection Type** of default network interfaces. If a default interface is not required, edit the interface and select **Disabled**.
If *default-static* and *default-DHCP* are changed to the same configuration method (that is, if both are set to *Static assignment* or both are set to *DHCP*), neither interface works.

CELLULAR HEALTH REPORTING

Administrative users can control the cellular health reporting settings under ⚙ **> SERVICES > Cell Health Reporting**.

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, the appropriate status is shown in Lighthouse for the node.

ENABLE CELL HEALTH REPORTING

To enable Cell Health Reporting:

1. Select ⚙ **> SERVICES > Cell Health Reporting**.
2. Select **Enabled**.
3. Set the **Signal Quality Ranges** to report. Click and drag the range handles to the range value corresponding to: *Bad*, *Moderate* and *Good*.
4. Set the **Frequency** that Lighthouse will check the signal quality.
5. Set whether to enable the **Cell Connectivity Test** to get a complete cell health report.

> **Note:** This will cause cell data consumption.

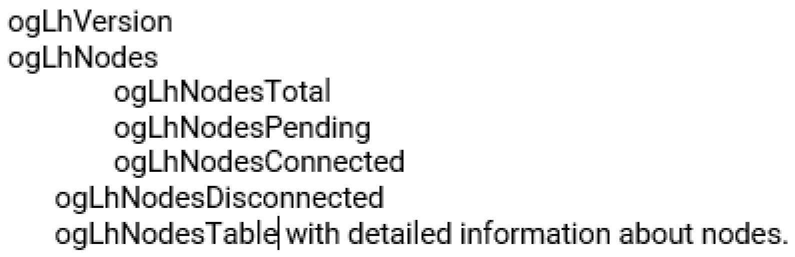
6. Click **Apply**.

LIGHTHOUSE MIBS

Lighthouse *Management Information Bases (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. The MIBs turn the SNMP data into text that is more readable to human readers.

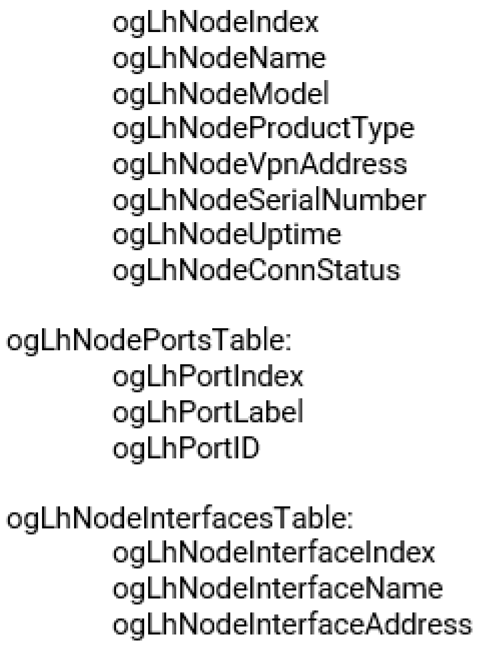
GENERIC INFORMATION ABOUT LIGHTHOUSE VERSION AND NODES COUNT



```
ogLhVersion
ogLhNodes
          ogLhNodesTotal
          ogLhNodesPending
          ogLhNodesConnected
     ogLhNodesDisconnected
     ogLhNodesTable with detailed information about nodes.
```

AVAILABLE INFORMATION FOR AN ENROLLED OPENGEAR NODE

ogLhNodesTable:



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

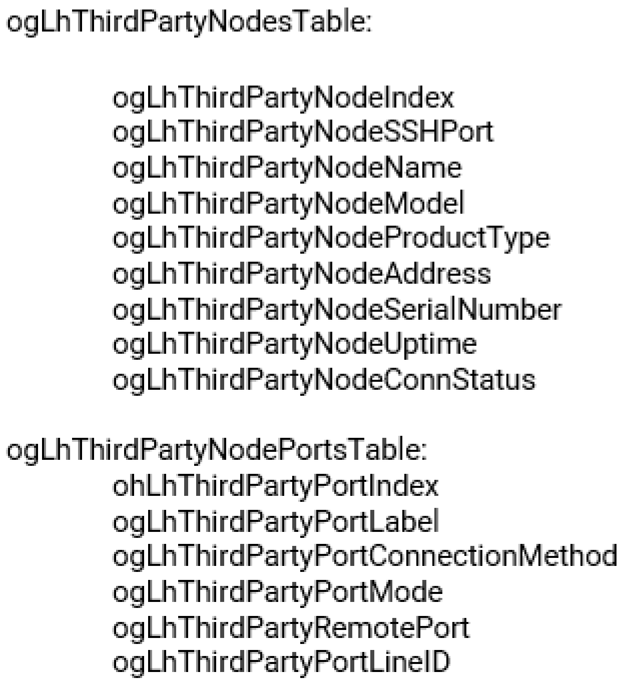
ogLhNodePortsTable:

- ogLhPortIndex
- ogLhPortLabel
- ogLhPortID

ogLhNodeInterfacesTable:

- ogLhNodeInterfaceIndex
- ogLhNodeInterfaceName
- ogLhNodeInterfaceAddress

AVAILABLE INFORMATION FOR AN ENROLLED THIRD-PARTY NODE



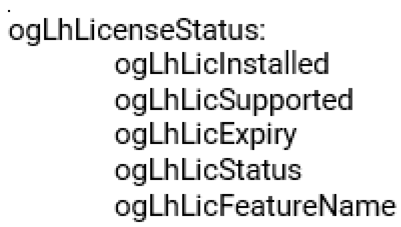
ogLhThirdPartyNodesTable:

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

ogLhThirdPartyNodePortsTable:

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

AVAILABLE LICENSING INFORMATION



```
ogLhLicenseStatus:
        ogLhLicInstalled
        ogLhLicSupported
        ogLhLicExpiry
        ogLhLicStatus
        ogLhLicFeatureName
```

AVAILABLE ENROLLED NODE CELLULAR HEALTH INFORMATION

ogLhNodeCellularHealth

OTHER SNMP COMMANDS

To retrieve Lighthouse specific information use SNMP commands such as:

- `snmpwalk`
- `snmpget`

EXAMPLES OF LIGHTHOUSE MIB QUERIES THAT USE 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:

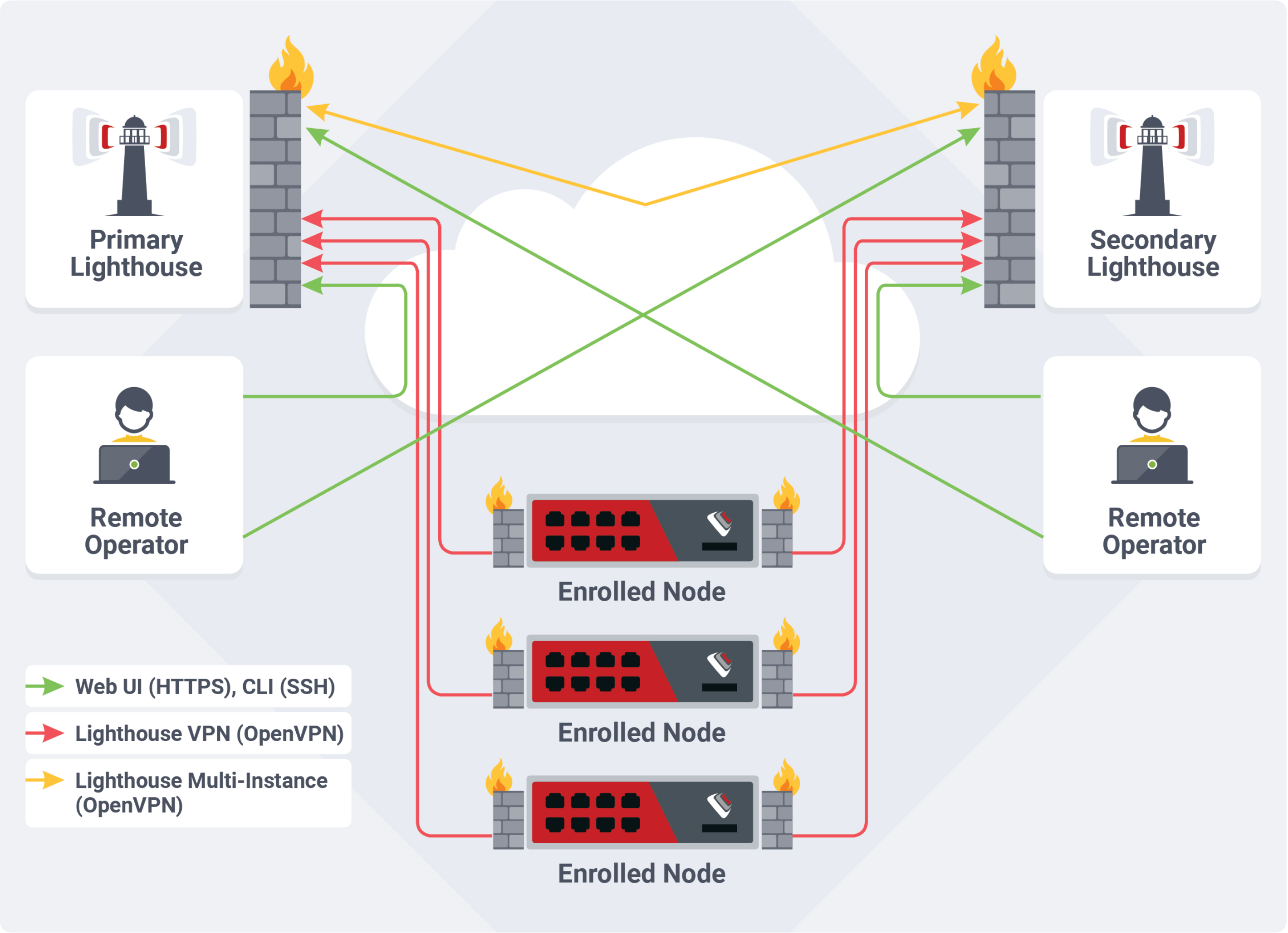
```
snmpwalk -m ALL -v1 -c public 192.168.1.1 ogLhNodeSerialNumber.192.168.128.2

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)
```

SET UP MULTIPLE INSTANCES OF LIGHTHOUSE

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



The multiple instance functionality allows you to set up secondary or dependent instances of Lighthouse that automatically receive updates from a primary Lighthouse instance and maintains connections to all its remote nodes.

Secondary instances are read-only. They may be used to view Lighthouse information specific to that instance using `ogconfig-cli`, and to connect via `pmshell`.

Configuration changes must be performed on the primary instance, which then updates the information displayed on the secondary instance.

SET UP MULTI INSTANCE

Lighthouse supports up to 10 secondary instances for each primary Lighthouse instance.

A Lighthouse with multiple instance support requires multiple separate subnets for Lighthouse VPN connections between:

- each instance and its nodes.
- the primary and secondary instance Lighthouses.

> **Note:** Each subnet must not overlap any subnet in use by another Lighthouse instance.

Before you attempt to set up a multiple instance:

- 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.
- Configure the networking information for each instance (hostname, external endpoints, network addresses, REST API port).
- Configure the time settings of each instance.
- Ensure you have a subscription active on your primary Lighthouse.

STEPS TO SETUP MULTI INSTANCE

To set up multi Instance on the primary Lighthouse:

1. On the primary Lighthouse, select ⚙ **> MULTI INSTANCE > Secondary Lighthouses**.
2. Click ⊕ **Add Secondary Lighthouse**.
3. Enter the **Description** for the Lighthouse.
4. Enter the **Network Address**: accepts either the current IP Address or hostname.
5. Enter the **Network Port** for HTTPS connections.

6. Enter the **Username**.
7. Enter the **Password**.
8. Provide the **VPN NETWORK RANGE** details:
   - Address Space: enter a valid, unused network subnet to use as the secondary lhvpn address range.
   - CIDR Subnet Mask.

   Notes:
   - Lighthouse displays the Calculated Node Capacity based on the values entered reflecting the addressable nodes based on the network.
   - The secondary Lighthouse instance must be able to reach the primary instance on UDP Port 1195.
9. Provide the **SMART MANAGEMENT FABRIC RANGE** details if the secondary lighthouse is to be discoverable on the Smart Management Fabric:
   - Address Space
   - CIDR Subnet Mask

   Note: Lighthouse displays the Calculated Node Capacity based on the values entered reflecting the addressable nodes based on the network.
10. Click **Apply**.

ENABLE ALTERNATE REST API PORTS

If you are planning to use the alternate REST API ports, you must make sure this option is enabled on both the primary and secondary Lighthouse servers, prior to enrollment of secondaries.

Lighthouse will prevent the enrollment of a secondary Lighthouse instance if there is a mismatch in these settings. If this occurs, the message "Lighthouse is using Alternate API port" will be displayed on the secondary Lighthouse page.

To fix the issue, either:

- Enable the Alternate REST API port on both Lighthouse servers, or
- Disable the Alternate REST API port on both Lighthouse servers, then delete the failed Lighthouse Enrollment and try again.

CONFIGURE THE ALTERNATE REST API

The Alternate REST API Port is configured as follows:

1. Select ⚙ **> SYSTEM > Session Settings**.
2. For **Alternate API Port Status**, select **Enabled**.
3. Click **Apply**.

Note: Additional enrollment-only REST API port is 8443.

CONFIGURE SUBNETS FOR A MULTIPLE INSTANCE LIGHTHOUSE

A Lighthouse with multiple instance support requires multiple separate subnets for Lighthouse VPN connections between:

- each instance and its nodes.
- the primary and secondary Lighthouses.

Note: Each subnet must not overlap any subnet in use by another Lighthouse instance.

CONFIGURE THE SUBNETS

To configure the subnet between the primary Lighthouse and its nodes:

1. Select ⚙ **> MULTI INSTANCE > Multi Instance VPN** on the primary Lighthouse.

   The **MULTI INSTANCE VPN** page displays.

2. Enter the **Address Space**.

3. Enter the **CIDR Subnet Mask**.

> **Note:** The *Calculated Address Capacity* is a calculated field and displays the addressable nodes based on the network.

CONFIGURE SECONDARY INSTANCES

To configure the subnet between each secondary Lighthouse and its node:

1. Select ⚙ **> MULTI INSTANCE > Secondary Lighthouses** on the primary Lighthouse.

   The **SECONDARY LIGHTHOUSES** page displays.

2. Click on the name of the secondary Lighthouse to edit.
   The **EDIT SECONDARY LIGHTHOUSE** page displays.

3. Under the **VPN NETWORK RANGE** section:

   a. Update the *Address Space*.

   b. Update the *CIDR Subnet Mask*.

   > **Note:** The *Calculated Node capacity* is a calculated field and displays the addressable nodes based on the network.

4. If the secondary lighthouse is to be discoverable on the Smart Management Fabric , under the **SMART MANAGEMENT FABRIC RANGE** section:

a. Update the *Address Space*.

> **Notes:**
>
> - The *CIDR Subnet Mask* is a calculated field.
> - The *Calculated Node capacity* is a calculated field and displays the addressable nodes based on the network.

CONFIGURE SECONDARY INSTANCE INFORMATION PRIOR TO ENROLLMENT

Other information that is specific to the secondary 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

> **Note:** The instance specific information is available on both the primary and secondary Lighthouses but it is read-only on the secondary Lighthouse.

Configurations of all Lighthouse instances are stored in `lighthouse_configurations`.

These can be viewed via `ogconfig-cli`. The primary instance has a value of `Primary` for its role, and secondary instances have the value `Secondary`.

The following is an example of the `ogconfig-cli` session:

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

ogcfg> print lighthouse_configurations[0].role

lighthouse_configurations[0].role (string): 'Primary'

ogcfg> print lighthouse_configurations[1].role

lighthouse_configurations[0].role (string): 'Secondary'
```

Alternatively, the command `/usr/bin/is_secondary` outputs `n` for a primary Lighthouse or `y` for a secondary Lighthouse.

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
```

UNENROLL A SECONDARY INSTANCE

To unenroll a secondary Lighthouse instance from the primary Lighthouse:

1. Click ⚙ **> MULTI INSTANCE > Secondary Lighthouses**.
2. Select the **Instance Name**.
3. Click 🗑 **Unenroll Instance** in the top-right of the page.

   A confirmation message displays.
4. Click **Confirm**.

PROMOTE 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 set up a secondary Lighthouse if required.

Caution: 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` or run with `sudo` privileges on the secondary instance via console or ssh and run:

```
>sudo promote-secondary-lighthouse
```

Remove all dead connections from node side using the node's web UI. The Promotion tool deletes connection between primary and dependent 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.

- All scheduled firmware upgrades are cancelled in the event of a secondary Lighthouse promotion, and must be rescheduled.
- Firmware files are not replicated among the multiple instance cluster and must be re-uploaded to the new primary after promotion.

ENABLE ALERTING AND MONITORING

Administrative users can configure the Simple Network Management Protocol (SNMP) Manager settings and multiple external servers to export the syslog to via TCP or UDP.

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. 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. The SNMP service starts after it has been configured correctly. If the user does not provide an Engine ID, an auto-generated ID will be used. Lighthouse Health statistics (load/uptime/memory usage, etc.) can be retrieved.

SNMP MANAGER SETTINGS

To enable SNMP Manager:

1. Select ⚙ **> SERVICES > Alerting and Monitoring**.
2. Select **Enabled** to enable the SNMP Manager.
3. For Manager Protocol, select **UDP**, or **UDP over IPv6**, **TCP**, or **TCP over IPv6**.
4. Enter the **Manager Address** to receive SNMP messages.
5. Enter the **Manager Port**, the TCP/UDP port number to send SNMP messages to.
6. Select the SNMP protocol **Version:v1** or **v2c** or **v3**.

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

   For **v1**:

   1. Enter the **SNMP Community** to use for messages.

   For **v2c**:

   1. Select 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**.
      1. If *TRAP* is selected, specify an optional **Engine ID** for sending an SNMP TRAP message

         > **Note:** If left blank, the auto-generated Engine ID from the SNMP Service is used. An Engine ID is not required for an SNMP INFORM message.

   2. Enter the SNMP v3 **Engine ID** and required Configure SNMP Manager Settings.
   3. Enter the **Username** to send the messages as, select the **Authentication Protocol**, either MD5 or SHA, and enter the SNMP user's **Authentication Password**.
   4. Choose a **Privacy Protocol**, either **MD5** or **SHA**.
   5. Enter a **Privacy Password**.

7. Under the SNMP Message Settings section, set the events to trigger a *TRAP/INFORM* notification by checking a combination of:
   - *Node Connection Status*: send an SNMP notification when a node connects or disconnects.
   - *Secondary Replication Status*: send an SNMP notification when problems occurs with database replication on secondary Lighthouse.
   - *Node Cellular Health Status*: to use Cellular Health Status change traps.
8. Click **Apply**.

For information on Structure of notifications for Opengear nodes, refer to OG-LIGHTHOUSE-MIB.mib.

SNMP SERVICE SETTINGS

To enable the SNMP Service:

1. Select ⚙ **> SERVICES > Alerting and Monitoring**.
2. Select the **SNMP SERVICE** tab.

3. Select the TCP/IP Protocol from *UDP* or *TCP*.
4. Enter the **Location**.
5. Enter the **Contact**.
6. Set the SNMP Version the service will support by checking a combination of the following:
   - v1/v2
   - v3
7. If *v1/v2* is checked configure the settings under the *SNMP V1 & V2C* section:
   a. Enter the Read-Only Community.
   b. Enter the Read-Write Community.
8. If *v3* is checked configure the settings under the *SNMP V3* section:
   a. Enter the Engine ID. Override the automatically generated SNMPv3 Engine.

   > **Note:** If not specified, an Engine ID will be automatically generated using Network Interface details.

   b. Select the Security Level: *No Security*, *Authentication*, *Authentication and Encryption*.

   > **Tip:** The SNMP v3 security level *Authentication and Encryption* is recommended.

   For *No Security*:

   i. Enter the Read Only Username

   For *Authentication*:

   i. Enter the Read Only Username
   ii. Select the Authentication Protocol: MD5 or SHA
   iii. Enter and confirm the Authentication Password

   For *Authentication and Encryption*:

   1. Enter the Read Only Username
   2. Select the Authentication Protocol: MD5 or SHA
   3. Enter and confirm the Authentication Password
   4. Select the Privacy Protocol: DES or AES
   5. Enter and confirm the Privacy Password.

9. Click **Apply**.

SYSLOG SERVER MANAGEMENT

VIEW THE CONFIGURED SYSLOG SERVERS

1. Select ⚙ **> SERVICES > Alerting and Monitoring**.
2. Select the **SYSLOG** tab.

ADD A NEW SYSLOG SERVER

1. Select ⚙ **> SERVICES > Alerting and Monitoring**.
2. Select the **SYSLOG** tab.
3. Click the ⊕ **New Syslog Server** button.

   The **CREATE NEW SYSLOG SERVER** dialog displays.
4. Enter the *Server Address*.
5. Select the Protocol: *UDP* or *TCP*.
6. Enter the Port.

   > **Note:** For UDP the default port is 514. For TCP the default port is 601.
7. Click **Create Syslog server**.

MODIFY A SYSLOG SERVER

1. Select ⚙ **> SERVICES > Alerting and Monitoring**.
2. Select the **SYSLOG** tab.
3. Locate and click the *SERVER ADDRESS* of the server to modify from the grid.

   The **UPDATE** dialog displays.
4. Modify the details as required.
5. Click **Update Syslog server**.

DELETE A SYSLOG SERVER

To enable the SNMP Manager:

1. Select ⚙ **> SERVICES > Alerting and Monitoring**.
2. Select the **SYSLOG** tab.
3. Locate and click the *SERVER ADDRESS* of the server to modify from the grid.
4. Click 🗑 **Delete**.

   The **DELETE SYSLOG SERVER** confirmation dialog displays.
5. Click **Delete**.

ENABLE SMART MANAGEMENT FABRIC

Smart Management Fabric represents an advanced functionality designed to offer heightened flexibility and accessibility to network and IT professionals throughout the network fabric. It empowers them by facilitating effective orchestration and management through the management network.

As Smart Management Fabric expands its reachability through the utilization of OSPF and the integration of Opengear nodes and Lighthouse for establishing new paths, it is crucial to acknowledge the potential risk of overexposing the network, which could lead to bypassing Layer 2 or Layer 3 access control measures.

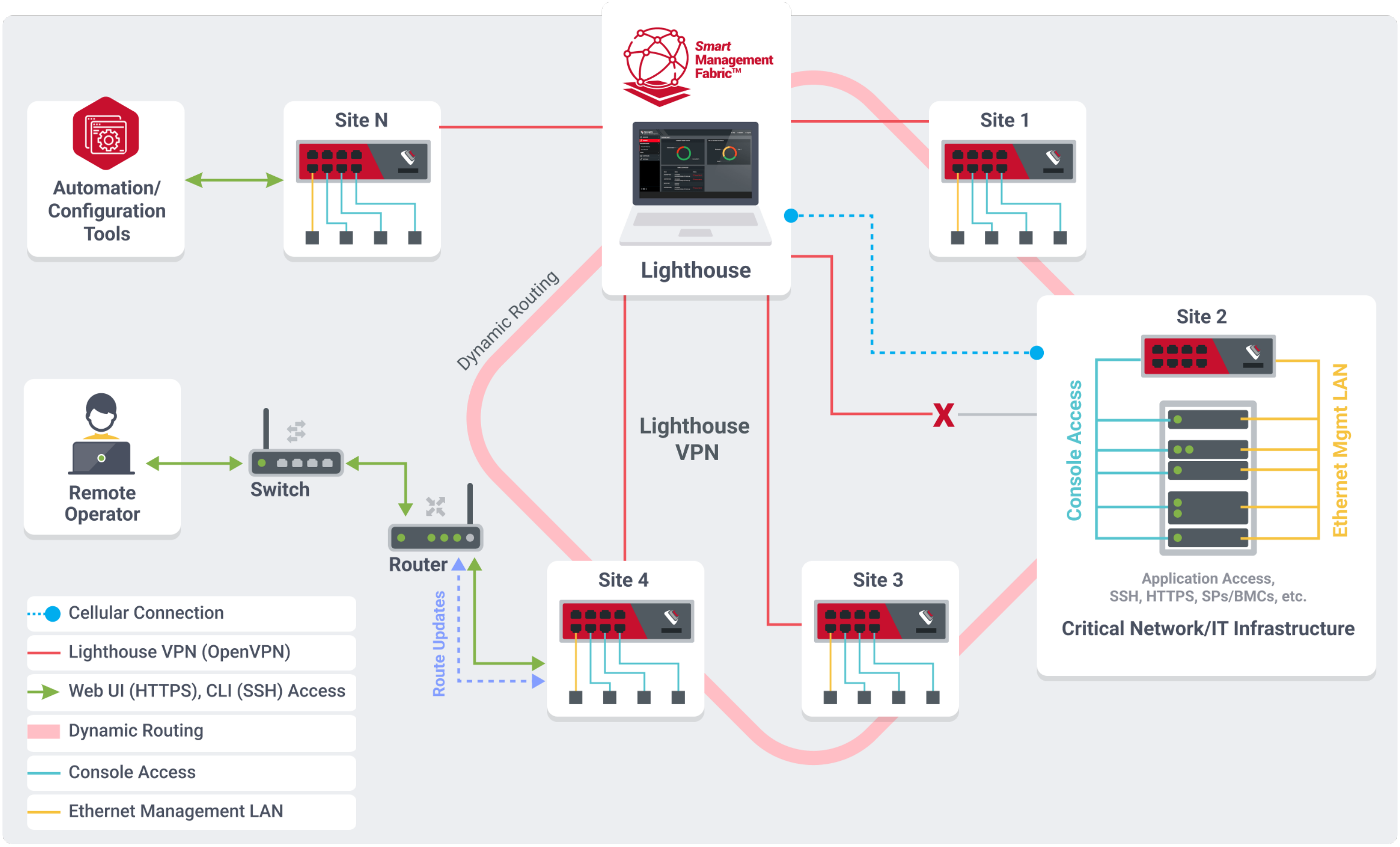
While the communication between the Lighthouse and OpenGear nodes, such as OM, is safeguarded through a VPN connection, and the OSPF configuration is carefully restricted and secured, there exists a potential risk when devices like routers and switches under customer autonomy are configured within the OSPF process without adequate diligence. To address and mitigate these risks, the following strategies are recommended:

- Conduct a meticulous examination of networks participating in OSPF advertisement, with a suggestion to implement passive interfaces.
- Execute comprehensive testing and verification to ensure that no routing occurs among networks not involved in Smart Management Fabric.
- Verify the activation of OSPF authentication to augment network security.

Smart Management Fabric uses dynamic link state routing to allow IP connectivity to IT resources that are on connected IPv4 networks that are downstream from the lighthouse:

- via SSH, https (GUI), SPs/BMCs (iLO, iDRAC, etc.).
- via commonly used automation tools such as RDP, Ansible, Python, vCenter.

To implement Smart Management Fabric an Automation Edition subscription is required, as well as a supported (23.10 firmware and up) Opengear console server such as Operations Manager.



Note: Smart Management Fabric is advanced functionality that utilizes dynamic routing protocols. It is crucial to acknowledge the potential risk of overexposing your network.

Enable Smart Management Fabric

After deploying the Lighthouse, to set up Smart Management Fabric to create an internal network area between Lighthouse and the console servers:

1. Log in to the Lighthouse web UI as a Lighthouse Administrator or the root user.
2. From the menu, select ⚙ **> SERVICES > Smart Management Fabric**.

3. Select **Enabled**.
4. Enter the **Internal Area ID** for the backbone area for the internal network.

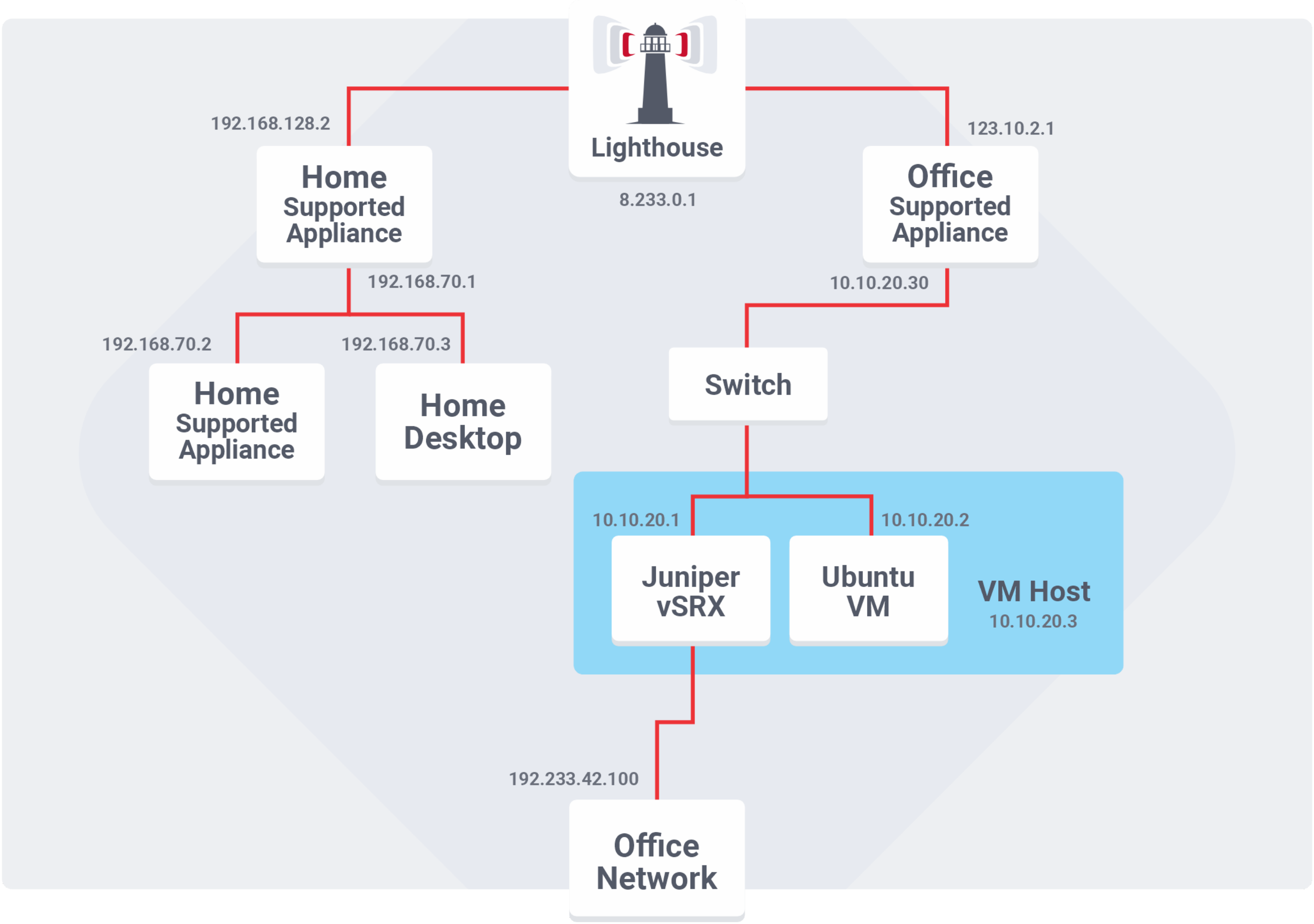
   The area is a logical collection of internal networks, routers, and links with the same area identification.
5. Click **Apply**.

To ensure the Smart Management Fabric stays up to date, these are the following scenarios that would require an additional push of configuration after the initial setup:

- Any changes made to Smart Management Fabric VPN subnets for the Primary or Multi Instance Lighthouses.
- A new Multi-Instance Lighthouse is added that is required to be part of the Smart Management Fabric network.
- Any changes to the Lighthouse VPN for the Primary or Multi Instance Lighthouse that may include:
  - Any subnet changes.
  - Any changes to MTU for a specific node.

SMART MANAGEMENT FABRIC - A SAMPLE DEPLOYMENT

In this sample, the devices on two separate networks, can connect via Lighthouse, when Smart Management Fabric is enabled.



USE SMART MANAGEMENT FABRIC

To set up Lighthouse as the router for a deployment:

1. Configure the Address space.
2. Enable Smart Management Fabric and provide the internal area id.
3. Configure the Smart Management Fabric templates for the nodes.

SMART MANAGEMENT FABRIC AND NETOPS INTERACTIONS

The Smart Management Fabric feature can be used with the following NetOps features with some caveats related to the following:

- IP Access.
- Secure Provisioning.

> **Note:** Automation Gateway is not compatible with Smart Management Fabric.

IP ACCESS

The IP Access Subnet Address is not unique and is present on every IP Access enabled node. This will cause routing issues for the Smart Management Fabric if there are multiple IP Access enabled nodes.

SECURE PROVISIONING

The default Smart Management Fabric subnet range conflicts with the Secure Provisioning subnet range, however either change the Smart Management Fabric subnet range before enabling secure provisioning, or change the Secure Provisioning Subnet range before it is deployed.

SECURE PROVISIONING CONFIGURATION

The Secure Provisioning Subnet is not unique by default so if the default is used on more than one Smart Management Fabric node then it will effectively not be able to route to those Secure Provisioning connected devices.

Customize the Secure Provisioning Subnet by setting a static IPv4 address on the Node’s LAN interface before deploying Secure Provisioning. Secure Provisioning will then inherit the subnet range from a static IPv4 address on the node’s LAN interface address.

MANAGING LIGHTHOUSE

This section describes how to manage Lighthouse to monitor and manage your network.

MANAGE SUBSCRIPTIONS

Lighthouse has a flexible, simplified subscription model that allows you to add extended functionality if required.

> **Note:** Contact sales@opengear.com with a request for the type of subscription, including the number of nodes to which you want to apply the subscription. You will receive an encrypted .zip file named ***subscription.zip***.

VIEW SUBSCRIPTIONS

1. Select ⚙ **> SYSTEM > Subscriptions**.

   The **SUBSCRIPTIONS** page displays.
   The page displays the **SUBSCRIPTIONS** tab providing information on your subscriptions including the type of subscriptions, number of assigned out of the total nodes available, subscription status and the number of days before expiry.

VIEW SUBSCRIPTION ASSIGNMENTS

1. Select ⚙ **> SYSTEM > Subscriptions**.

   The **SUBSCRIPTIONS** page displays.
2. Select the **SUBSCRIPTION ASSIGNMENT** tab.
   The page displays a gird of nodes with the respective subscription assigned.

> **Note:** Node *Filter* and *Search* controls are available.

SUBSCRIPTION TIERS

The subscriptions available are:

- The Enterprise Edition Subscription provides base functionality to enroll, monitor, and manage nodes.
- The Enterprise Automation Edition provides base functionality plus the advanced feature sets:
  - Smart Management Fabric.
  - Connected Resource Gateway.
  - Automation Gateway.
  - Secure Provisioning.

You can apply the extra functionality to selected nodes only, giving you the flexibility to use different subscriptions for different nodes.

Subscription licensing is clearly displayed on the UI. Users can see the active subscriptions, the node count supported, assigned nodes, and time left on the subscription.

ADD A NEW SUBSCRIPTION TO LIGHTHOUSE

1. Select ⚙ **> SYSTEM > Subscriptions**.
2. Click **Add Subscription** for the applicable subscription tier.
   The **NEW SUBSCRIPTION** dialog displays.
3. Click **Choose file** and select the required zip file.
4. Click **Apply**.

ASSIGN SUBSCRIPTIONS TO NODES

Users can reassign nodes to different subscriptions to optimize Automation Edition features on their network.

For example, if the user:

- Previously had two license tiers, but only paid to renew one license tier.
- Purchased an Automation Edition license for the first time, and wants the extra functionality on some of their nodes.
- Wants to swap which nodes have access to the extra functionality available in an Automation Edition license.

ASSIGN A SINGLE NODE TO A SUBSCRIPTION

To assign a single node to a subscription:

1. Select ⚙ **> SYSTEM > Subscriptions**.
2. Select the **SUBSCRIPTION ASSIGNMENT** tab.
3. To assign a subscription to a node click the ⇄ **Assign subscription to node** button in the right-hand column for the required node.
   The **ASSIGN SUBSCRIPTIONS** dialog displays showing details of the node in the left panel and subscription options in the right panel.
4. Select from the available subscription options presented in the right panel.
5. Click **Confirm Assignment**.

ASSIGN MULTIPLE NODES TO A SUBSCRIPTION

To assign multiple nodes to a subscription:

1. Select ⚙ **> SYSTEM > Subscriptions**.
2. Select the **SUBSCRIPTION ASSIGNMENT** tab.
3. Select the node(s) to add to a subscription by selecting the checkbox in the left column.
   The ⇄ **Update Subscriptions** button is displayed above the table.

4. Click ⇥ **Update Subscriptions**.

   The **ASSIGN SUBSCRIPTIONS** dialog displays showing details of the node in the left panel and subscription options in the right panel.

5. Select from the available subscription options presented in the right panel.
6. Click **Confirm Assignment**.

Note: If a node that is assigned to the Automation Edition and has Smart Management Fabric configured is reassigned to an Enterprise Edition, the disable Smart Management Fabric template is automatically pushed to that node and it is no longer a part of the Smart Management Fabric network.

Tip: Another way of removing a node from a subscription is to unenroll the node. The subscription then becomes available to be assigned to another node.

UPDATE A SUBSCRIPTION TO LIGHTHOUSE

1. Select ⚙ **> SYSTEM > Subscriptions**.
2. Click ⇥ **Update Subscription** for the applicable subscription tier.

   The **NEW SUBSCRIPTION** dialog displays.

3. Click **Choose file** and select the required zip file.
4. Click **Apply**.

UPGRADE LIGHTHOUSE OVERVIEW

Lighthouse can be upgraded using a `.lh_upg` image file. Note the following conditions:

- AWS requires `.aws.lh_upg` and Microsoft Azure requires `.azure.lh_upg`. All other platforms use the standard .lh_upg file.

- Incremental upgrades to Lighthouse using `lh_upg` files are only supported from 20.Q3.x and not earlier releases.

> **Note:** Upgrades do not overwrite existing configurations or user files, however it is recommended that you perform a Configuration Backup before you upgrade a Lighthouse.

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

ABOUT THE UPGRADE PROCESS

Lighthouse performs the following high-level steps to upgrade to the new version:

1. Validates the upgrade image, and takes a backup of the current root filesystem and snapshot of data volume.
2. Reboots into the new root filesystem (from the upgrade image).
3. Performs data migrations and system re-configurations.
4. Commits the changes on success, or reverts on failure.
5. For Multiple Instance only, after a successful Primary upgrade: After a 10 minute delay, secondary instances are automatically upgraded in sequence.

Lighthouse uses LVM snapshots to manage storage during upgrade and data migration. If a failure occurs, the snapshot is used to revert the system to its previous state.

In regard to software versions, Lighthouse must be upgraded in succession. For example, to upgrade to a version that is several releases newer than your current release, you must install all the major releases in between to install the newest one.

PREPARE TO UPGRADE LIGHTHOUSE

While upgrading Lighthouse is a simple process, the following tasks are best practices to ensure a successful upgrade:

- Read the release notes, in particular any known issues or special requirements for the new version
- Provide a sufficiently large maintenance window to start and complete the upgrade. Larger networks and more complex configurations will take more time. You can run `q-stat` to check the job queue status.
- Download the correct upgrade image.
  - The current release can be found here: Opengear FTP
  - Archived releases are available here: Archives Opengear FTP
- Verify the image integrity by checking against SHASUMS. For example, on Linux:

  ```
  $ sha1sum lighthouse-24.06.02.lh_upg 0f52c30a212566030a1742b6e3d5bf9316d89abd
  lighthouse-24.06.02.lh_upg
  ```

- Before starting the upgrade, generate a Technical Support Report. This can assist Opengear support with diagnosing upgrade issues.
- Use the df command to display disk usage by filesystem:

  ```
  root@lighthouse:~# df /mnt/data/
  ```

- Ensure that you have enough disk space in `/mnt/data`. If you have less than 20% disk space available, clean up unnecessary files, or add disk space. If disk space is added to the primary instance, add additional space all secondary instances also.
- Take a Configuration Backup on the Primary Lighthouse. This can be used to restore the system in the event of failure during upgrade.

- Do not make any configuration changes during the upgrade process (includes secondary upgrades).
- Consider taking a Virtual Machine backup. If the Lighthouse storage has been increased (by adding additional physical volumes to LVM) then these must be included in the backup, and all storage volumes must be in sync.
- You can also consider running some technical checks before starting the upgrade to check job queues, memory, temporary filesystem (tmpfs), database integrity, and multiple instance status.

UPGRADE LIGHTHOUSE

UPGRADE A PRIMARY LIGHTHOUSE

To upgrade a primary Lighthouse:

1. Select ⚙ **> SYSTEM > System Upgrade**.
2. Select the **Upgrade Method**, either **Fetch image from HTTP/HTTPS Server** or **Upload Image**.
    - To upgrade via **Fetch image from HTTP/HTTPS Server**, enter the URL for the system image in the *Image URL* text-entry field.
    - To upgrade via **Upload Image**, navigate to the directory containing the appropriate upgrade image file and drag and drop the image onto the target page section or click **select file** to open a dialog.

    > **Note:** To upgrade to a version that is several releases newer than your current release, you must install all the major releases in between before the latest one.

3. Click **Perform Upgrade**.

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

> **Caution:** The **Advanced Options** section should only be used as part of an Opengear Support call.

UPGRADE MULTI INSTANCE LIGHTHOUSE

> **Note:** The upgrade must be performed through the Primary Lighthouse instance.

When the primary Lighthouse is updated, any secondary instance Lighthouses are updated after the primary has successfully booted. Secondary Lighthouse upgrades are performed in parallel (not in a queue) to speed up the overall process.

> **Note:** If any Lighthouse fails to successfully upgrade along the way, the upgrade stops.

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

Where there are multiple instances of Lighthouse, when a system upgrade is being performed the status of secondary instances is flagged in the ⚙ **> SYSTEM > System Upgrade** page.

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.

> **Note:** Information about the upgrade progress and status is visible by navigating to the 🗎 **Jobs** page.

UPGRADE LIGHTHOUSE VIA THE CLI

Lighthouse includes a shell-based tool — `sysflash` — that allows a user with administrative privileges to upgrade the system for the instance from the local terminal.

> **Note:** Before you use `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.

UPGRADE VIA LOCAL TERMINAL

To upgrade the system for the Lighthouse instance using the Lighthouse local terminal:

1. Select **Terminal**.
2. At the `[hostname] login:` prompt, enter an administrator username and press **Enter**.
3. At the `Password:` prompt, enter the administrator's password and press **Enter**.
4. To use `sysflash` with a `.lh_upg` file available via an HTTP or HTTP server, at the local terminal bash shell prompt, enter a URL that it is *URL encoded*.

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

5. Press **Enter**.

UPGRADE VIA LOCAL FILE SYSTEM

To upgrade Lighthouse with `sysflash` and a .lh_upg file available via the local file system:

1. Select **Terminal**.
2. At the `[hostname] login:` prompt, enter an administrator username and press **Enter**.
3. At the `Password:` prompt, enter the administrator's password and press **Enter**.

4. At the local terminal bash shell prompt enter:

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

5. Press **Enter**.

> **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.

List the flags by running either of the following at a local terminal bash shell prompt:

`sysflash -h` or

`sysflash --help`

UPGRADE NETOPS MODULES

NetOps Modules are released independently of Lighthouse software or Operations Manager firmware.

Node upgrades may be carried out through the Lighthouse UI.

UPGRADE VIA LIGHTHOUSE UI

1. Log in to the Lighthouse web UI with a user assigned with **Netops** permissions.
2. Select 🖧 **NetOps > NetOps Installation**.
3. Select **Online**.
4. Click ⇄ **Start Online Sync**.

TROUBLESHOOT THE UPGRADE PROCESS

There are two main reason why an upgrade fails:

Reason	Description
Migration failures	Lighthouse will return to the pre-upgrade state. An example of such a failure is malformed data in the database, which does not conform to more stringent schema checks in the new version.
System level failures	This would include environmental issues such as power loss during the upgrade. If such an issue prevented Lighthouse from rolling back to the previous state, Lighthouse may be left in an unusable state.

RECOVER FROM A FAILED UPGRADE

If Lighthouse is in an unusable state (either as a result of upgrade problems, or any other catastrophic failure), you can:

1. Promote a secondary Lighthouse, or
2. Restore a configuration backup, or
3. Manually retry upgrade.

PROMOTE A SECONDARY LIGHTHOUSE

This requires multiple instances of Lighthouse. If the primary Lighthouse instance is unreachable, one of the existing secondary instances can be promoted to become the new primary Lighthouse instance.

The old primary Lighthouse instance should be replaced with a fresh installation of Lighthouse. This can then be enrolled as a secondary to the newly promoted primary Lighthouse

RESTORE A CONFIGURATION BACKUP

Deploy a Lighthouse instance with the version you are upgrading from. Restore the configuration backup taken before the failed upgrade.

MANUALLY RETRY UPGRADE

Manually retry via the CLI if a secondary upgrade fails. Determine the reason for the upgrade failure and resolve the issue before you retry the upgrade. Assistance from Opengear technical support may be required.

Troubleshoot a Failed Secondary Lighthouse Upgrade

> **Note:** The **CLI** interface commands have maintained references to **unbound** instances. Within the Lighthouse user interface, dependent instances have been renamed to Secondary Lighthouses.

To retry the upgrade for a secondary Lighthouse, first list the instance and identify the IDs of the secondary lighthouses that must be retried by running the `retry_dependent_upgrades list` command.

`retry_dependent_upgrades list` command will list all the secondary lighthouses and their current status, as in the example below:

```
root@lighthouse:~# retry_dependent_upgrades list

ID UUID Hostname Firmware Version Fw Status Lighthouse Status

2 lighthouse_configurations-2 lighthouse-2 22.11.2 not updated UpgradeFailed

3 lighthouse_configurations-3 lighthouse-3 24.02.0 updated Enrolled

4 lighthouse_configurations-4 lighthouse-4 22.11.2 not updated UpgradeFailed
```

In the example above, the Lighthouses with IDs 2 and 4 have failed to upgrade (note that the “Enrolled” status is the wanted status). To re-trigger an upgrade for those lighthouses run the following command. Ensure that the lighthouse IDs are comma separated:

```
retry_dependent_upgrades trigger -l 2,4
```

The method above is the recommended way but there is a shortcut to re-trigger all failed secondary upgrades by running the command below:

```
retry_dependent_upgrades trigger --failed
```

Note: Only one secondary upgrades at a time.

You should be able to view the secondary upgrade progress in the primary Lighthouse UI as you would for a normal upgrade.

Contact Support

In the unlikely event that Lighthouse does not automatically recover, or your troubleshooting efforts fail, contact Opengear support for advice with the following information available to share:

- If your instance is usable, download a Technical Support Report (from the Help menu).
- The configuration backup taken before the upgrade process.

CONFIGURATION BACKUP

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

1. Select ⚙ **> SYSTEM > Backup, Restore and Reset**.
2. Specify additional **User Files** you also want to include in the backup.
   This is optional and enter one file or directory per line.
3. Specify whether to **Encrypt Backup**. To enable:
   a. Select **Enabled**.
   b. Enter and confirm a password.

4. Click **Download Backup** and save this file.

   The filename consists of a timestamp and `lh_bak` extension, for example: `lighthouse-20190710100325.lh_bak`

CONFIGURATION RESTORE

To restore the configuration and user files:

1. Select ⚙ **> SYSTEM > Backup, Restore and Reset**.
2. Select **CONFIGURATION RESTORE**.
3. Navigate to the directory that contains the appropriate configuration image file, then drag and drop the file onto the target page section or click **select file** to open a dialog. Supported files: .lh_bak.
4. Enter the **Backup Password** if the configuration was encrypted.
5. Click **Restore Backup**.

   The **CONFIGURATION RESTORE CONFIRMATION** dialog displays.
6. Click **Yes**.

   Lighthouse restores the backup and any included user files, then restarts.

Considerations If Using Multiple Instances

Lighthouse Version	**Consideration**
BEFORE 23.04.1	Existing dependent Lighthouse instances will not be synced with the primary, and must be deleted and re-enrolled again.
23.04.1 AND LATER	The newly restored primary should re-establish contact with the existing dependent instances. The databases may be out of sync at first, but will be re-synced automatically.

SHUT DOWN OR RESTART LIGHTHOUSE

The following sections describe the process to shut down or restart Lighthouse.

This includes finding instance version, returning to factory setting, and shutting down or restarting a running version of Lighthouse.

FIND THE CURRENT LIGHTHOUSE INSTANCE VERSION

You can find the current Lighthouse version in the following ways:

- Displayed under the logo in the top-left of the display.
- Located within the Lighthouse UI in the instance details.
- Via the local Lighthouse shell.

VERSION INFORMATION WITHIN THE LIGHTHOUSE UI

1. Select ☰.

   The information panel expands.

VERSION INFORMATION VIA THE LOCAL SHELL

1. Select ⊡ **Terminal**.
2. At the `[hostname] login:` prompt, enter an administrator username and press **Enter**.
3. At the `Password: prompt`, enter the administrator's password and press **Enter**.
4. At the bash shell prompt, enter `cat /etc/version` and press **Enter**.

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

```
root@lighthouse:~# cat /etc/version 2022.Q1.0
```

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.

OTHER INFORMATION SOURCES

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

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

2022.Q1.0
```

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 2022.Q1.0 \n \l
```

SHUT DOWN A RUNNING LIGHTHOUSE INSTANCE

1. Select **Terminal**.
2. At the `[hostname] login:` prompt, enter an administrator username and press **Enter**.
3. At the `Password: prompt`, enter the administrator password and press **Enter**.
4. Enter the command `shutdown` now and press **Enter**.

   The Lighthouse instance shuts down.

RESTART A RUNNING LIGHTHOUSE INSTANCE

1. Select **Terminal**.
2. At the `[hostname] login:` prompt, enter an administrator username and press **Enter**.
3. At the `Password: prompt`, enter the administrator password and press **Enter**.
4. At the prompt enter one of the following commands:
   - `reboot` and press **Enter**, or
   - `shutdown -r now` and press **Enter**.

   The Lighthouse instance shuts down and reboots.

RETURN A LIGHTHOUSE INSTANCE TO FACTORY SETTINGS

At some stage, you may require a return to the factory settings. You can do this either through the UI or by running a shell script if you have root access.

> **Note:** During this process, the current Lighthouse configuration is overwritten and user files are deleted. If you want, you can create a backup of the configuration and any required user files.

RESTORE FACTORY SETTINGS

To return a Lighthouse instance to its factory settings using the Lighthouse UI:

1. Login to the Lighthouse as root.

   > **Note:** 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 ⚙ **> SYSTEM > Backup, Restore and Reset**.
3. Select the **FACTORY RESET** tab.
4. Read and understand the warning presented.
5. To proceed, type **YES** in the confirmation field.
6. Click **Reset Lighthouse**.

MONITOR JOBS

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

To view current or scheduled jobs:

1. Select 🗎 **Jobs**.

   The JOBS page displays defaulted to the **CURRENT** tab.

2. To display jobs that have completed, select the **ENDED** tab

3. Use the **Filter Jobs** control to refine the jobs displayed.

4. To view the details of a job, select the link in the **Job Type** column.

   The job details dialog displays.

CONFIGURING LIGHTHOUSE

Lighthouse can be customized as required, by the creation of users and user groups, to enable Smart Management Fabric, for authentication, and to enable network traffic mirroring.

MANAGE TEMPLATES

Templates are a centralized way of changing the configuration for enrolled Opengear device nodes by pushing predefined configuration templates to selected nodes.

The following are the types of config templates to create and manage:

- Authentication
- Scripts
- Smart Management Fabric
- Users & Groups

Other templates that can be pushed around:

- NetOps Modules
- Port Logging

Both have predefined templates to deploy as required.

MANAGE AUTHENTICATION TEMPLATES

Only users assigned to the *Lighthouse Administrator* role can access **Node Tools > Config Templates > Authentication Templates** and manage authentication templates.

CREATE A NEW AUTHENTICATION TEMPLATE

> **Note:** Only *Lighthouse Administrators* can create authentication templates.

1. Select **Node tools > Config Templates**.

   The **CONFIG TEMPLATES** page displays.

2. Click ⊕ **New Config Template**.

   The **PICK A CONFIG TYPE** selection options expand.

3. Select **Authentication**.

   The **ADD AUTHENTICATION TEMPLATE** page displays.

4. Enter a **Name** and **Description** for a template in the **Template Details** section.

5. Select **Pre Populate** to populate the template fields with the current Lighthouse remote authentication settings.

6. Select the required **Scheme**:

   - Local Users Only
   - Radius
     a. Enter the Remote authentication server *Address* and *Port*.
     b. If required, click ⊕ **Add Authentication Server** to add a server.
     c. If required, click ⊕ **Add Accounting Server** to add a server.
     d. Enter the *Server Password* and *Confirm Password*.
   - TACACS+
     a. Enter the Remote authentication server *Address* and *Port*.
     b. If required, click ⊕ **Add Authentication Server** to add a server.
     c. Select the **TACACS+ login method** used to authenticate to the server. Defaults to PAP. To use DES encrypted passwords, select *Login*
     d. Enter the *Server Password* and *Confirm Password*.
     e. Enter the TACACS+ service to authenticate with. This determines which set of attributes are returned by the server. Defaults to "raccess"
   - LDAP

a. Enter the Remote authentication server *Address* and *Port*.

b. If required, click ⊕ **Add Authentication Server** to add a server.

c. If required, enter **LDAP Base DN** details. The distinguished name of the search base. For example: `dc=my-company,dc=com`.

d. If required, enter **LDAP Bind DN** details. The distinguished name to bind to the server with. The default is to bind anonymously.

e. If required, enter *Bind DN Password* and *Confirm Password*

f. If required, enter **LDAP Username attribute** details. The LDAP attribute that corresponds to the login name of the user (commonly "sAMAccountName" for Active Directory, and "uid" for OpenLDAP).

g. If required, enter **LDAP group membership attribute** details.

h. Check **Ignore Referrals** to disregard LDAP referrals to other servers (Only applied to OM Devices).

i. Select the **SSL protocol**.

j. Check **Ignore SSL certificate errors** so errors encountered when accessing LDAPS servers will be ignored.

k. Select and upload the CA certificate to validate LDAPS servers.

7. Click **Save Template**.

Notes:

- When an authentication template is pushed to a node, the authentication settings at that node are replaced by the authentication settings defined in the authentication template.
- The authentication templates do not support the full list of settings that the Opengear devices support. However, templates can be applied, and then additional settings configured manually.

EDIT AN AUTHENTICATION TEMPLATE

1. Select **Node tools > Config Templates**.

   The **CONFIG TEMPLATES** page displays.

2. The **AUTHENTICATION** tab is selected.

3. Select the name of the template to be modified.

   The **EDIT AUTHENTICATION TEMPLATE** page displays.

4. Make required changes.

5. Click **Save Template**.

DELETE AN AUTHENTICATION TEMPLATE

1. Select **Node tools > Config Templates**.

   The **CONFIG TEMPLATES** page displays.

2. The **AUTHENTICATION** tab is selected.

3. Select the name of the template to be deleted.

   The **EDIT AUTHENTICATION TEMPLATE** page displays.

4. Click 🗑 **Delete Template**.

   A confirmation dialog displays.

5. Click **Delete**.

MANAGE SCRIPT TEMPLATES

Script Templates allow the user to upload arbitrary shell scripts to be run on a node. For example, a script may set additional configuration settings not available in other templates or store additional files onto the node such as certificates.

The uploaded script must:

- Have a *.sh* extension.
- Be less than *1MB* in size.

Other than those, there are no other restrictions on the script file to be uploaded. When saved, the template stores the size and SHA1 checksum of the script. This can be used to verify the script contents of the template when saved.

To apply script templates, the selected nodes must be running firmware version 4.1.1 or later.

CREATE A NEW SCRIPT TEMPLATE

Note: Only *Lighthouse Administrators* can create script templates.

1. Select **Node tools > Config Templates**.

   The **CONFIG TEMPLATES** page displays.
2. Click ⊕ **New Config Template**.

   The **PICK A CONFIG TYPE** selection options expand.
3. Select **Scripts**.

   The **ADD SCRIPT TEMPLATE** page displays.
4. Enter a **Name** and **Description** for the template.
5. Enter the **Script timeout**.

   This is the time to wait (in minutes) for the execution of the script. After timeout is reached, script will be stopped.
6. Select and upload the script file. Supported files: `.sh`
7. Click **Save Template.**

Note: *Script checksum* and *Script size* are displayed after the template with uploaded script is saved.

EDIT A SCRIPT TEMPLATE

1. Select **Node tools > Config Templates**.

   The **CONFIG TEMPLATES** page displays.

2. Select the **SCRIPT** tab.

3. Select the name of the template you want to modify.

   The **EDIT SCRIPT TEMPLATE** page displays.

4. Make required changes.

5. Click **Save Template**.

DELETE A SCRIPT TEMPLATE

1. Select **Node tools > Config Templates**.

   The **CONFIG TEMPLATES** page displays.

2. Select the **SCRIPT** tab.

3. Select the name of the template you want to delete.

   The **EDIT AUTHENTICATION TEMPLATE** page displays.

4. Click **Delete Template**.

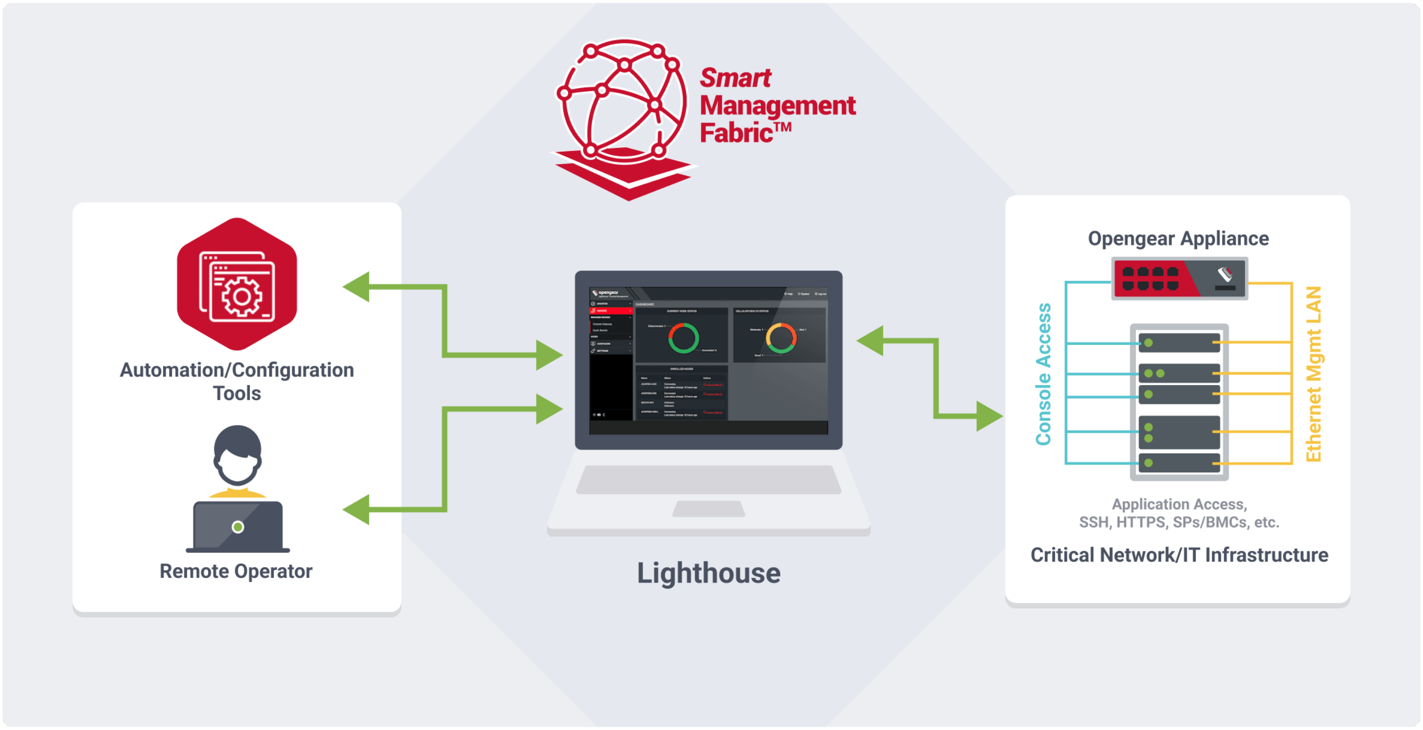
   A confirmation dialog displays.

5. Click **Delete**.

MANAGE SMART MANAGEMENT FABRIC TEMPLATES

Smart Management Fabric allows you to create an enterprise grade Open Shortest Path First (OSPF) solution for network connectivity, configuration and troubleshooting devices connected to supported nodes on Lighthouse, wherever they may be.

Caution: Smart Management Fabric is an advanced feature. Prior to roll out, it is recommended to ensure its compliance with your enterprise security posture.


Smart Management Fabric™
Automation/Configuration Tools
Remote Operator
Lighthouse
Opengear Appliance
Console Access
Ethernet Mgmt LAN
Application Access, SSH, HTTPS, SPs/BMCs, etc.
Critical Network/IT Infrastructure

Smart Management Fabric uses dynamic link state routing to allow IP connectivity to IT resources regardless of whether the connection is either:

- Physically, via USB, serial ports that are configured with IP addresses
- Virtually, via SSH, https (GUI), SPs/BMCs (iLO, iDRAC, etc.)
- Via commonly used automation tools such as RDP, Ansible, Python, vCenter

Smart Management Fabric templates allow OSPF to be run on a node. For example, a template may set network addresses, subnet masks and authentication methods. To apply Smart Management Fabric templates, the selected nodes must be:

- a supported Opengear appliance
- running firmware version 23.10 or later.

Note: A user can override the smart management fabric configuration by using the command line interface tool.

CREATE A SMART MANAGEMENT FABRIC TEMPLATE

1. Select **Node tools > Config Templates**.

   The **CONFIG TEMPLATES** page displays.

2. Click ⊕ **New Config Template**.

   The **PICK A CONFIG TYPE** selection options expand.

3. Select **Smart Management Fabric**.

   The **ADD SMART MANAGEMENT FABRIC TEMPLATE** page displays.

4. Enter the **Name** and **Description** of the template.

5. Check the applicable ***Routing Options***:

   - Check **Redistribute Connected** to redistribute directly connected routes to all routers in the attached Routing Information Protocol (RIP) domain.
   - Check **Redistribute Kernel** to share the static routes on the devices.

6. Check **Enable Masquerading for all interfaces** to override individual masquerading settings on interfaces.

7. Select the **OSPF Configuration Method**. Select one of the following to send link-state advertisements on one of following:

   - Interface - to advertise the OSPF on the whole interface.
   - Network - to advertise the OSPF on a selected network.

8. Select the OPSF Configuration Method.

- Select **Interface** to advertise OSPF on the whole interface.
  1. Click ⊕ **Add Interface**.

     The **ADD INTERFACE** dialog displays.
  2. Enter the *Interface Name*. This must match the name of the interface on the device.
  3. Check the *Masquerade Interface* to show that traffic from this interface appears to come directly from the supported node to other devices in the subnet.
  4. Check the *Passive Interface* if routes are not to be propagated to the network it is on.
  5. Enter the link *Cost* used in OSPF route calculations which indicates the overhead required to send packets across the specified interface.
  6. Select the Authentication Method *MD5*, *Clear Text* or *No Authentication*.

     > **Note:** For *MD5* at least one key is required. Click ⊕ **Add Digest Key** to enter the key details.

- Select **Network** to define which networks should be advertised.
  1. Click ⊕ **Add Network**.
  2. Enter the *Address*, *Subnet Mask*, and *OPSF Area*.
  3. Click ⊕ **Add Interface**.

     The **ADD INTERFACE** dialog displays.
  4. Enter the *Interface Name*. This must match the name of the interface on the device.
  5. Check the *Masquerade Interface* to show that traffic from this interface appears to come directly from the supported node to other devices in the subnet.
  6. Check the *Passive Interface* if routes are not to be propagated to the network it is on.
  7. Enter the link *Cost* used in OSPF route calculations which indicates the overhead required to send packets across the specified interface.
  8. Select the Authentication Method *MD5*, *Clear Text* or *No Authentication*.

> **Note:** For *MD5* at least one key is required. Click ⊕ **Add Digest Key** to enter the key details.

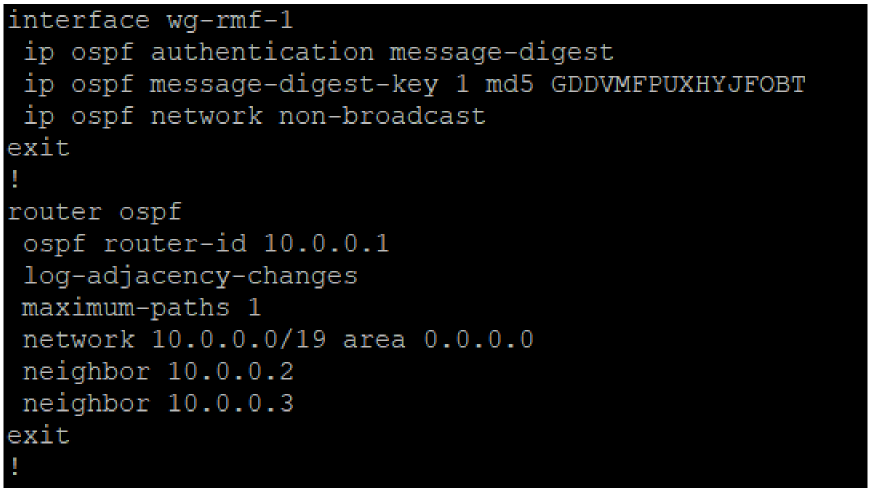
9. Click **Save Template**.

VALIDATE SMART MANAGEMENT FABRIC TEMPLATES

To check that the Smart Management Template is applied and to verify that OSPF is running and configured, login to Lighthouse and Appliances CLI. Run the following commands to validate that the Smart Management Fabric template is properly applied:

```
vtysh -c "show running-config"

vtysh -c " sh ip ospf neigh" -c " sh ip ospf route"
```

The templated OSPF/SMF values display, for example:



```
interface wg-rmf-1
 ip ospf authentication message-digest
 ip ospf message-digest-key 1 md5 GDDVMFPUXHYJFOBT
 ip ospf network non-broadcast
exit
!
router ospf
 ospf router-id 10.0.0.1
 log-adjacency-changes
 maximum-paths 1
 network 10.0.0.0/19 area 0.0.0.0
 neighbor 10.0.0.2
 neighbor 10.0.0.3
exit
!
```

EDIT A SMART MANAGEMENT FABRIC TEMPLATE

1. Select **Node tools > Config Templates**.

   The **CONFIG TEMPLATES** page displays.

2. Select the **SMART MANAGEMENT FABRIC** tab.
3. Select the name of the template you want to modify.

   The **EDIT SMART MANAGEMENT FABRIC TEMPLATE** page displays.
4. Make the required changes.
5. Click **Update Template**.

Caution: Firewall rules are added to the nodes to allow OSPF traffic.

Note: You cannot edit the system *Disable Smart Management Fabric* template.

DELETE A SMART MANAGEMENT FABRIC TEMPLATE

1. Select **Node tools > Config Templates**.

   The **CONFIG TEMPLATES** page displays.
2. Select the **SMART MANAGEMENT FABRIC** tab.
3. Select the name of the template you want to delete.

   The **EDIT SMART MANAGEMENT FABRIC TEMPLATE** page displays.
4. Click **Delete Template**.

   A confirmation dialog displays.
5. Click **Delete**.

Note: You cannot delete the system *Disable Smart Management Fabric* template.

MANAGE USER AND GROUP TEMPLATES

To navigate to the page:

1. Select **Node Tools > Config Templates**.
2. Select the **USERS AND GROUPS** tab.

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 of which they are a member.

> **Note:** Each template must contain at least one group.

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.

CREATE A NEW USER AND GROUP TEMPLATE

1. Select **Node tools > Config Templates**.

   The **CONFIG TEMPLATES** page displays.
2. Click ⊕ **New Config Template**.

   The **PICK A CONFIG TYPE** selection options expand.
3. Select **Users & Groups**.
   The **ADD USERS AND GROUPS TEMPLATE** page displays.
4. Enter a *Name* and *Description* for the template.
5. Set the group list:

   > **Note:** Groups provided in the list replace any user defined *groups* on the node.

   a. Click ⊕ **Add Group**.

      The **ADD GROUP** dialog displays.

   b. Enter the **Group Name**.

   c. Enter the **Description**.

   d. Select the **Role**; either *Node Administrator* or *Node User*.

   e. For the role *Node User*, check the *Restrict accessible Serial Ports* to enter the *Permitted Serial Ports Range*.

      Ranges use the format start-finish eg., `1,3-5,8`.

   f. Click **Add Group**.

6. Set the user list:

   > **Note:** Users provided in this list replace any user defined *users* on the node.

   a. Click ⊕ **Add User**.

      The **ADD USER** dialog displays.

   b. Enter the **User Name**.

   c. Enter the **Description**.

   d. Enter the **Password** and **Passsword Confirmation**.

   e. Check the Group Name(s) to select the *Group Membership*

   f. Click **Add User**.

7. 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 remain on the node. To push users, the selected nodes must be running firmware version 4.3.0 or later.

EDIT A USERS AND GROUPS TEMPLATE

EDIT A USERS AND GROUP TEMPLATE

1. Select **Node tools > Config Templates**.

   The **CONFIG TEMPLATES** page displays.

2. Select the **USERS AND GROUPS** tab.
3. Select the name of the template you want to modify.

   The **EDIT USERS AND GROUPS TEMPLATE** page displays.

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

DELETE A GROUP FROM A TEMPLATE

1. Select **Node tools > Config Templates**.

   The **CONFIG TEMPLATES** page displays.

2. Select the **USERS AND GROUPS** tab.
3. Select the name of the template you want to delete.

   The **EDIT USERS AND GROUPS TEMPLATE** page displays.

4. Select the group name from the **SET GROUP LIST** section.

   The **UPDATE GROUP** dialog displays.

5. Click 🗑 **Delete**.
6. Click **Save Template**.

DELETE A USER FROM A TEMPLATE

1. Select **Node tools > Config Templates**.

   The **CONFIG TEMPLATES** page displays.

2. Select the **USERS AND GROUPS** tab.
3. Select the name of the template you want to modify.

   The **EDIT USERS AND GROUPS TEMPLATE** page displays.
4. Select the username from the **SET USER LIST** section.

   The **UPDATE USER** dialog displays.
5. Click 🗑 **Delete**.
6. Click **Save Template**.

DELETE A USER AND GROUP TEMPLATE

1. Select **Node tools > Config Templates**.

   The **CONFIG TEMPLATES** page displays.
2. Select the **USERS AND GROUPS** tab.
3. Select the name of the template you want to delete.

   The **EDIT USERS AND GROUPS TEMPLATE** page displays.
4. Click 🗑 **Delete Template**.

   A confirmation dialog displays.
5. Click **Delete**.

DELETE USERS OR GROUPS FROM A TEMPLATE

> **Note:** To save the template, the system requires a minimum of either one *Group* or one *User* added.

DELETE A GROUP FROM A TEMPLATE

1. To modify a template by deleting a group, select **Node Tools > Config Templates** from the main menu.

   The **CONFIG TEMPLATE** dashboard displays.
2. Select the **USERS AND GROUPS** tab.
3. Select the template you want to modify.

   The **EDIT USERS AND GROUPS TEMPLATE** page displays.
4. Select the group you want to delete from the **SET GROUP LIST** section by clicking the name of the group.

   The **UPDATE GROUP** dialog displays.
5. Click the **Delete** button.

   The group is deleted.
6. Click **Save Template** to save the changes.

DELETE A USER FROM A TEMPLATE

1. To modify a template by deleting a group, select **Node Tools > Config Templates** from the main menu.

   The **CONFIG TEMPLATE** dashboard displays.
2. Select the **USERS AND GROUPS** tab.
3. Select the user you want to delete from the **SET USER LIST** section by clicking the username.

   The **UPDATE GROUP** dialog displays
4. Select the user to be deleted.

   The **UPDATE USER** dialog displays
5. Click the **Delete** button.

   The user is deleted.
6. Click **Save Template** to save the changes.

PUSH TEMPLATES

Users with **Lighthouse Administrator** privileges (that is, users with the Lighthouse Administrator role or users who are members of groups with the Lighthouse Administrator role) can access **Node TOOLS > Config Templates** and push templates affecting nodes in node filters linked to their role.

The following types of templates can be applied to selected nodes:

- Authentication
- Users and Groups
- Script
- Netops Module Activation
- Port Logging
- Smart Management Fabric

Lighthouse Administrators can manually apply the Netops modules of Secure Provisioning, Automation Gateway, or IP Access to the suitable Opengear Console Server node.

PUSH A TEMPLATE

The process to push templates consists of four stages, each one a step in the overall wizard:

1. Select a template.
2. Select the target nodes.
3. Run the Pre-flight Test. This test run simulates what happens if the template is pushed to the selected nodes.
4. Push the template.

SELECT A TEMPLATE

1. Select **Node Tools > Config Templates**.

   The **CONFIG TEMPLATES** page displays.

2. Select a template config type tab:

   - AUTHENTICATION
   - NETOPS MODULES
   - PORT LOGGING
   - SCRIPT
   - SMART MANAGEMENT FABRIC
   - USERS AND GROUPS

3. For the required template click the **Push template** icon on the right of the row.

   The **PUSHING TEMPLATE** page displays.

SELECT THE TARGET NODES

1. Select the target nodes by clicking the check boxes next to them.

   > **Note:** Third-party nodes are not supported for template execution.

RUN PRE-FLIGHT TEST

1. Click **Run Pre-Flight Test**.

   After all nodes finish pre-flight, the Pre-flight Status goes green and the **Push Configuration** button becomes active.

PUSH THE TEMPLATE

1. Click **Push Configuration**.
2. Confirm the *TEMPLATE RUN STATUS* per node as displayed in the column.
3. Confirm the *PUSH STATUS* per node as displayed in the column.
4. Click **Exit to Config Templates**.

MANUALLY APPLY NETOPS MODULES VIA TEMPLATE

1. Select **Node Tools > Config Templates**.

   The **CONFIG TEMPLATES** page displays.
2. Select the **NETOPS MODULES** tab.

   The *Secure Provisioning*, *IP Access*, and *Automation Gateway* templates display.
3. For the required template click the *Push template* icon on the right of the row.

   The **PUSHING TEMPLATE** page displays.
4. Select the required nodes by clicking the check boxes next to them.

   > **Note:** Only applicable Console Servers for the module display during this node selection step.
5. Click **Run Pre-Flight Test**.

   The test status is displayed in the PREFLIGHTSTATUS column for the node.
6. When pre-flight is complete, click **Push Configuration**.

CONFIGURATION AND TECHNICAL SUPPORT REPORTS

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

The support technician may ask for this report if you contact Opengear Technical Support.

GENERATE A SUPPORT REPORT VIA THE LIGHTHOUSE INTERFACE

1. Select ? to expand the help menu.
2. Select **Generate Technical Support Report**.

   The help menu updates to display the following message:

   > Report is being generated.
   >
   > Download will start automatically.

   The generation completes and the help menu updates to display the following message:

   > 👍 Report Downloaded Successfully.

GENERATE A SUPPORT REPORT VIA THE LOCAL TERMINAL

1. Select >_ **Terminal**.

   A local terminal window displays.
2. At the [hostname] login: prompt, enter an administrator username and press **Enter**.
3. At the password: prompt, enter the administrator password and press **Enter**.
4. At the bash shell prompt, enter: `support-report -z > /tmp/support.zip` and press **Enter**.

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 zip 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.

Following are two options for copying the file from Lighthouse:

1. 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
   ```

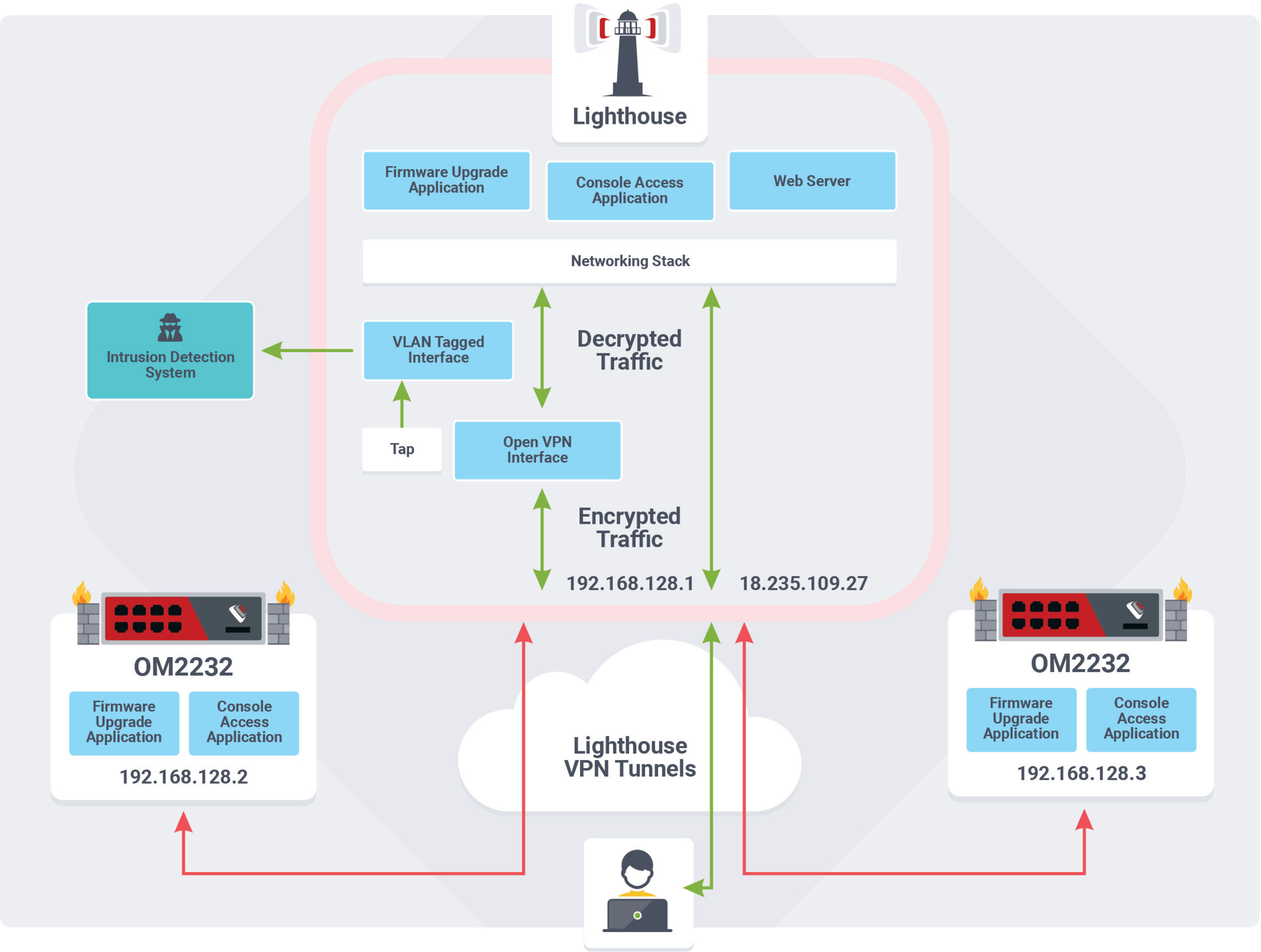
   Note: For Windows users, `WinSCP` on Win10 also works.

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

   ```
   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.
   ```

CONFIGURE LIGHTHOUSE FOR NETWORK TRAFFIC MIRRORING

Lighthouse can be integrated with a user's enterprise Intrusion Detection System (IDS) for real-time detection of security events in their network infrastructure.



Lighthouse allows users to configure the network traffic tap feature via the command line interface. The `traffic_mirroring` command:

- Mirrors all network traffic over the encrypted OpenVPN tunnel between Lighthouse and the Opengear supported appliances, and forwards all network traffic as decrypted packets to a configurable endpoint. The endpoint is expected to be a “gateway” IP address of an external device that is routable from Lighthouse.

- Preserves the original UDP and TCP/IP header information while mirroring (so that the IDS can reassemble TCP streams and inspect the payload).

> **Note:** Traffic mirroring is only supported on TCP and UDP.

- Provides an option to add a configurable VLAN tag to the Ethernet header.
- Works with multiple instances.

Only users with `sudo` access on the primary Lighthouse CLI (for example, via the admin group) can enable or disable traffic mirroring.

Users must ensure that:

- the traffic is routed to the required destination in their enterprise network.
- their firewall rules allow traffic mirroring.

> **Note:** All functionality is available only via the Lighthouse CLI. There is no UI or REST API interface for network traffic mirroring feature. For detailed CLI usage see `traffic_mirroring --help`.

CONFIGURE NETWORK TRAFFIC MIRRORING FOR MULTIPLE INSTANCES

You can configure network traffic mirroring for multiple instances of Lighthouse. It can mirror traffic between Lighthouses, and between a node and a dependent Lighthouse.

If new dependent Lighthouses are added to a network that is mirroring traffic, they must be re-configured for network traffic mirroring.

Note: All CLI configuration, including enabling and disabling, must be run on the primary Lighthouse. A dependent Lighthouse can only run the `--test` and `--status` arguments.

Users can specify different settings for each Lighthouse. For example:

- A dependent Lighthouse can have a different VLAN ID (or no VLAN ID), and a different destination IP.
- A dependent Lighthouse can be set to only mirror node traffic, and not multi-instance traffic. This is useful because the primary is already mirroring that traffic.
- You can enable or disable network traffic mirroring per instance.

Note: All newly enrolled secondary Lighthouse instances have network traffic mirroring disabled by default.

TROUBLESHOOT NETWORK TRAFFIC MIRRORING

It is possible that there may be momentary periods of up to a few seconds where traffic is not being mirrored. For example mirroring outages of a few seconds can occur during:

- Configuration if changes are being made to the VPN subnet or firewall.
- The Lighthouse boot process.

To ensure that traffic monitoring is uninterrupted, avoid rapid changes to configuration and repeated reboots of Lighthouse.

WORKING WITH LIGHTHOUSE NODES

After Lighthouse is installed and configured, enroll a small set of nodes, then create a set of tags and node filters that allow nodes access to be filtered to the correct subset of users.

When these nodes are installed, test access to the Web UI and serial ports for the node.

Node Filters, Port Filters and Resource Filters can be associated with a group, this restricts the group to only have access to those resources which match the filter. Note that these filters intersect for ports and nodes, if you give a user access to a node but access to the first port on every node, they can only access the first port on that node, and they won't be able to access the first port on any other nodes. If no filter is assigned, and the user has permissions to access the object in question ("Connected Resource Gateway" for resources and "Node and Devices (Base) for Nodes and Ports), then the user has access to all objects. Those permissions are disabled by default when you create a new role.

> **Note:** You can leverage the Connect Resource Gateway to proxy via SSH, HTTP or HTTPS to configure your third-party nodes. For more information about configuring your third-party nodes, see Set up Lighthouse as a Connected Resource Gateway (CRG).

ENROLLMENT BUNDLES

An Enrollment Bundle 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. This is useful for larger roll outs where many nodes are deployed with similar configuration and responsibilities. If relevant Node Filters 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 you to run a set of templates on a node, after it is enrolled.

ASSIGN SUBSCRIPTIONS TO A BUNDLE

- Every bundle is associated with a subscription.
- If you only have one subscription, it is selected automatically.
- If you have both an Enterprise Edition and Automation Edition subscription, then you must select which subscription to use for nodes in the bundle. Automation Edition provides all the capabilities of Enterprise Edition, plus the ability to use Automation Gateway, Secure Provision, Smart Management Fabric and Connected Resource Gateway.

CREATE AN ENROLLMENT BUNDLE

1. Select **Node Tools > Enrollment Bundles**.
2. Click the ⊕ **Add Enrollment Bundle** button.

   The **New Enrollment Bundle Details** page displays.
3. Enter a **Name** and **Token** for the bundle in the respective fields.
4. (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.
5. Under **Subscription Selection** select either: **Lighthouse Enterprise Edition** or **Lighthouse Enterprise Automation**.
   Where only one subscription is available, this is selected for you
6. Under the **BUNDLE TAGS** section, select / create the tags to apply to any nodes that enroll using this enrollment bundle.
   a. Click ⊕ **Add Tag**.
   b. Select the tag and the value from the drop-down lists or enter a new tag key and value in the fields.

7. Under the **BUNDLE TEMPLATES** section, select the templates to apply to any nodes that enroll using this Enrollment bundle.

   To add a template:

   a. Click ⊕ **Add Templates**.

      The **CHOOSE TEMPLATES** dialog displays.

   b. Select the templates to add.

   c. Click **Add**.

   To remove a template:

   a. Click the ✕ icon for the individual template.

   Note: The templates in the table are executed in the order they appear. The templates can be reordered using the ↓ ↑ icons.

   Caution: Template push operations stop if one template fails.

8. Under the **NETOPS MODULES** section, select the NetOps modules to automatically activate for any supported nodes.

   To add a bundle:

   a. Click ⊕ **Add Module**.

      The **CHOOSE MODULE** dialog displays.

   b. Select the required **NetOps Module** from the list.

   c. Click **Add**.

   To remove a bundle:

   a. Click the ✕ icon for the individual bundle.

Note: The modules in the table are executed in the order they appear. The modules can be reordered by using the ↓ ↑ icons.

Caution: Module push operations stop if one template fails.

9. Click **Apply**.

STRUCTURE OF AN ENROLLMENT BUNDLE FILE

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
```

MANAGE THE ENROLLMENT SETTINGS

1. Select **Node Tools > Enrollment Settings**.

   The **ENROLLMENT SETTINGS** page displays.

2. Enter the *Enrollment Token* for nodes to request enrollment.
3. Select the *Default Subscription* for call home node enrollments.

   The available nodes are displayed against the subscriptions.

4. Optionally click **Download** to download the manifest `.og` file.
5. Click **Apply**.

ENROLL NODES

Enrolling nodes is the process of connecting nodes to Lighthouse to make them available for access, monitoring, and management. A node is a device that can be enrolled with Lighthouse, allowing it to be accessed, managed, and monitored.

You can enroll nodes in the following ways:

- From the Lighthouse Web UI.
- From the Node Web UI.
- USB drive.
- Mass Enrollment using Zero Touch Provisioning (ZTP).
- Automatic enrollment of appliances as managed through the preference settings in the Lighthouse Service Portal (LSP). The Lighthouse Service Portal is part of the Opengear Customer Portal.

Note: OPERATIONS MANAGER support may be partial for earlier releases, which may currently involve *mass node enrollment using ZTP* and *enrollment via USB drive.* However, all template types are supported.

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

Lighthouse uses OpenVPN tunnels secured with certificate authentication to connect the Lighthouse instance and remote nodes. For the connections to work properly, the clocks/times between the Lighthouse instance and each remote node server must be synchronized. During the enrollment process when a new remote node is being added, if that node is not using NTP (Network Time Protocol) to synchronize its time, the node checks the HTTP Date header sent by Lighthouse in the enrollment request.

The remote node then sets its own local system clock to match the time shown in that HTTP Date header from Lighthouse. This ensures that the new remote node has its time matched to the Lighthouse before the VPN tunnel is established, preventing potential time sync issues between the tunnel endpoints.

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 by selecting **Node Tools > Enrollment Settings** from the menu and individual tokens set per Enrollment bundle.

Enrollment is a two-step process:

1. After enrollment begins, nodes receive their Enrollment package, and establish a VPN connection to Lighthouse.
2. The node is now in the *Pending* state and must 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.

ENROLLMENT VIA THE LIGHTHOUSE WEB UI

Note: Enrollment via Lighthouse Web UI only works if the node is reachable from Lighthouse.

To enroll a node:

1. Select ⊕ **Enroll Node** in the header pane.

   The **ENROLL NODES** dialog displays.

2. From the **Product Type** drop-down, select the type of node to enrol.

   You can select from:

   - Opengear device
   - Generic third party device
   - Avocent ACS6000
   - Avocent ACS8000
   - Avocent ACS Classic
   - Cisco 2900 Series
   - Digi Passport

3. Enter the **Network Address**.

4. Enter the **Username**, and **Password** of the node being enrolled.

   Note: 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.

5. Select the **Subscription Type**. Each type shows the number of available subscriptions:

   - Enterprise Edition.
   - Automation Edition.

6. If required, you can also select to **Auto-approve enrollment** and/or **Run pre-enrollment connectivity check**.
7. Click **Enroll Node**.
   When enrolled, the details for the console server are removed from the **Nodes > PENDING** tab and added to the **Nodes > ENROLLED** tab.

ENROLLMENT VIA THE NODE WEB UI

> **Note:** If a node is behind a firewall, Lighthouse cannot initiate enrollment.

1. Select **Nodes**.
2. For the enrolled node, click 💻 Web Access UI from the right column.
3. Enter credentials and login.
4. Select **Serial & Network > LIGHTHOUSE**.
5. Enter the **Server Address** of Lighthouse (which can be hostname, FQDN, or IP address).
6. Optionally, enter the **Server Port**.
7. Enter the **Enrollment Bundle** (if a specific bundle is being used), and the **Enrollment Token** (either the global token or the bundle-specific token).
8. Select **Apply Settings**.
   The enrollment process begins.
   If Auto Approve is not enabled, when enrolled, the node displays in the Lighthouse Web UI under the **Nodes > PENDING** tab, with a **Status** of *Approval*.
9. Click the ✓ **Approve Node** icon in the right hand column.

> **Note:** Step 9 is only applicable if Auto Approve is not Enabled in the Enrollment Bundle.

ENROLL NODES VIA OM, ACM, CM, AND IM WEB UI

Nodes can be enrolled from other UIs such as the **Operations Manager** (OM).

ENROLL VIA OM, CM8XXX WEB UI

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. More details are available in the related Opengear appliance manuals.

ENROLL VIA ACM AND IM WEB UI

On the Web UI, select **Serial & Network > Lighthouse** to open the **Request Enrollment with Lighthouse Server** page.

ENROLLMENT VIA USB DRIVE

An unconfigured device can be enrolled using a USB drive loaded with an enrollment bundle.

DOWNLOAD THE ENROLLMENT BUNDLE

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

   A list of existing **Enrollment Bundles** displays.

2. On the row of the required bundle, click the ⤓ icon to download .

   Depending on the browser’s configuration, a `manifest.og` file either downloads to the local system or displays a dialog asking to specify where to download the file.

ENROLLMENT VIA USB

1. Copy `manifest.og` to the root directory on a USB drive.
2. Plug the USB drive into an unconfigured and powered-down device.

3. Power the device up.
   On first boot, the device searches for the file — `manifest.og` — on any USB drives attached to the device and configures the device based on its contents.

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.

MANAGE NODES

After a node is enrolled, you can connect to it directly, either to monitor it, or to run commands on it, multiple nodes can be selected. Lighthouse allows you to manage all nodes. **Nodes** displays the status and number of all the ports on Lighthouse, which you have permissions to view/edit.

VIEW ALL NODES

1. Select **Nodes**.

   The NODES page displays defaulted to the **ENROLLED** tab.

2. To display nodes with an enrollment that has been initiated, select the **PENDING** tab.

FILTER NODES

1. Select **Nodes**.

   The NODES page displays.
   The following options are available:

   - Filter Nodes: create and save a new filter or select / edit an existing filter.
   - Enter a search term to filter the results. The AND operator is used if more than one search term is entered. To enter a multi word search phrase, enclose them in double quotes.

ASSIGN OR REMOVE A NODE TAG

Note: The following steps apply to enrolled nodes only.

1. Select **Nodes**.

   The NODES page displays.

2. Click the **Edit Node Tags** icon.

   The **ADD/EDIT TAGS** dialog displays.

3. Complete one of the following actions:

   - Create a new tag:

     a. Click ⊕ **Create new tag**.

        A new row displays.

     b. Select a tag name.

     c. Select a tag value

   - Remove an assigned tag:

     a. Select ✕ **Remove tag** for the specific tag.

        The tag is removed from the list.

4. Click **Apply**.

APPROVE PENDING NODES

1. Select **Nodes**.

2. Select the **PENDING** tab.

3. Perform one of the following actions:

   - Individually: use the ✓ icon located in the right hand column to apply the action.
   - Multiple: use the *'checkboxes'* to multi-select nodes and select ✓ **Approve Selected**.

   A Nodes approved message displays.

UNENROLL NODES

FOR A NODE THAT HAS BEEN ENROLLED

1. Select ☰ **Nodes**.
2. Use the *'checkboxes'* to select the node(s).
3. Select ✕ **Unenroll Selected**.

   A confirmation dialog displays.
4. Click **Remove**.

FOR A NODE THAT IS PENDING APPROVAL

1. Select ☰ **Nodes**.
2. Select the **PENDING** tab.
3. Perform one of the following actions:
   - Individually: use the ✕ icon located in the right hand column to apply the action.
   - Multiple: use the *'checkboxes'* to multi-select nodes and select ✕ **Unenroll Selected**.

   A confirmation dialog displays.
4. Click **Remove**.

CONNECT TO THE WEB-MANAGEMENT INTERFACE OF A NODE

After a node is enrolled, you can connect to the web-management interface for the enrolled node.

CONNECT FROM THE DASHBOARD

1. Select ⌂ **Dashboard**.

   The **NODES** grid displays defaulted to the **DISCONNECTED** tab.

2. Select the **CONNECTED** tab.
3. You can click the 💻 **Access Web UI** icon on the right of a connected node to access the Web UI of the node.

   The web-based login for that node loads.
4. Enter credentials and login.

Tip: A message displays at the bottom of the browser and provides a link to return to Lighthouse.

Note: The appearance of the Web UI depends on which device you have added.

CONNECT FROM THE NODES PAGE

You can access the WEB UI from the **ENROLLED** grid. For the required node, select the 💻 icon from the right hand column.

1. Select 🖴 **Nodes**.

   The **NODES** page displays defaulted to the **ENROLLED** tab.
2. Select the **CONNECTED** tab.
3. You can click the 💻 **Access Web UI** icon on the right of a connected node to access the Web UI of the node.

   The web-based login for that node loads.
4. Enter credentials and login.

CONNECT TO A NODE’S SERIAL PORTS

You can connect to the serial ports for the enrolled node via the Console Gateway option on the user interface if required. This allows the user to directly connect to any devices that are connected to the serial ports of the selected node.

FROM THE NODES PAGE

1. Select **Nodes**.
2. Use the **Filter Nodes** options to limit the displayed nodes.
3. Select the *node name* to display the **Node Details** page.

   The **PORTS** tab displays all the configured and unconfigured ports on the node.
4. Use the **Web Terminal Access** or SSH **SSH** icons on a particular port to access it.

FROM THE PORTS PAGE

1. Select **Ports**.

   A page displays all the ports on the connected nodes.
2. Find the port by using the **Filter Ports** or **Filter Nodes** options to restrict the listed resources within enrolled nodes.
3. Use the **Web Terminal Access** or SSH **SSH** icons on a particular port to access it.

CONNECT VIA SSH AND CHANGING THE DELIMITER CHARACTER

When connecting with SSH, 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:

1. Select **> SERVICES > Console Gateway**.
2. Enter a delimited character in the **Console Gateway Port Delimiter**.

   The carat, **^**, is the most common alternative.

   > **Note:** The auto-generated link default delimiter is '`:`'. URI syntax `ssh://user-name:console-server-name:port-number@lighthouse-ip-address`

3. Select the **Console Gateway SSH Address** to choose an address from which to SSH.

   > **Note:** 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 ⚙ **> SYSTEM> Network Settings** page.

EXAMPLE SSH 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#
```

CONNECT VIA THE WEB TERMINAL

To provide easy console port access, Lighthouse includes an HTML5 Web Terminal. The HTML5 Web Terminal includes native cut, copy, and paste support.

1. Locate the particular port from the **Nodes** or the **Ports** page.
2. Click the >_ **Web Terminal Access**.

   The **WEB TERMINAL** page displays.
3. Use `alt-x` to return to navigation.

SELECT 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 NODES

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"
```

To select all nodes:

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

RUN COMMANDS ON SELECTED NODES

When nodes are selected, the commands to be run for each can be given. These are run on each managed node in parallel. Any command that 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
```

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.

UPGRADE NODES VIA THE UI

When you are required to upgrade nodes to the latest firmware, for example, for security fixes, use the Lighthouse UI to upgrade up to 5000 connected nodes per task. You can upgrade nodes either immediately or at a scheduled time, outside normal business hours.

From the main menu, select **Node Tools > Firmware Upgrade**.

Completed jobs with the nodes selected can be duplicated so as to allow easy sequential upgrades. Nodes that failed to upgrade can be re-scheduled.

On opening the **FIRMWARE UPGRADE** page there are two tabs accessible, plus a Schedule Upgrade button, these are:

- Upgrade Tasks: A filtered dashboard where you can view scheduled, in progress and completed tasks and see their status.
- File Manager: An area that allows upgrade files to be uploaded and a table that displays previously uploaded files.
- Node Firmware Upgrade scheduling wizard: This is where you can set up and schedule firmware upgrades. The wizard is accessed by clicking on the + button in the Upgrade Tasks tab.

Completed jobs with the nodes selected can be duplicated so as to allow easy sequential upgrades. Nodes that failed to upgrade can be re-scheduled.

FIRMWARE FILES

Before you upgrade, ensure that you have access to the required firmware file.

UPLOAD A FIRMWARE FILE

Note: Only one file may be uploaded at a time using the upload tool. If multiple files are selected and placed in the drag and drop field, only the last file that was selected is uploaded.

1. Navigate to **Node Tools > Firmware Upgrade**.
2. Select the **File Manager** tab in the **Firmware Upgrade** page.
3. Complete one of the following actions to select the file:
   - Click ‘select file’ to select the file.
   - Drag and drop the file into the upload area.

   The file starts uploading and progress is displayed in the panel.

Click the ✕ button to cancel in-progress uploads .

If you close the website or if the HTTPS connection to Lighthouse is closed, the upload is cancelled.

DELETE A FIRMWARE FILE

To delete a firmware file that is no longer required:

1. Navigate to **Node Tools > Firmware Upgrade**.
2. Select the **File Manager** tab on the **Firmware Upgrade UI** page.
3. Click the **Delete** button next to the firmware file you want to delete.
4. Click the **Remove** button on the **Confirm Node Firmware Deletion** message box.

Note: If the selected firmware file is required by an ongoing or upcoming firmware upgrade task, an error message displays and the file is not deleted.

NODE UPGRADE TASKS

To upgrade a node you must set up a task.

CREATE AN UPGRADE TASK

The Node Upgrade UI is available in the Web UI under **Node Tools > Node Firmware Upgrade**.

Use the **UPGRADE TASKS** tab to:

- Upgrade a node
- Schedule a node upgrade
- Upload a new firmware file

To upgrade a node use the task information table:

1. Click the **Schedule Upgrade button** (**+**) to schedule an upgrade (or start one immediately).
   The **Node Firmware Upgrade** wizard displays.
2. Enter a name for the upgrade task.
3. Click the **Select File** button in the **SELECT FIRMWARE** pane.
   The Select Firmware list displays.
4. Select the firmware or upload new firmware for the upgrade task. Click **Select Firmware** button.
   Note that in the next step you must select the nodes to be upgraded with the firmware.
   The list of available and compatible nodes display in the **Select Nodes** pane. Note the messages in the **Select Firmware** pane. You can also use the **Filter Nodes** list to find the exact nodes you want to upgrade.

Note: Compatible nodes that have already been scheduled for an upgrade cannot be selected, these are visible in the list but appear greyed-out. Nodes that are not compatible with the firmware file are not listed.

5. Select the node/s. Click **Next - Schedule Upgrade** button.
   The page displays the **Start Time** pane and the nodes scheduled for upgrade.
6. Select the **Start time** as either **Immediately after creation** for immediate start, or, **Set time**.
   For Set Time, enter the Scheduled time Date and Time.

Note: The scheduled time is always in UTC.

7. Use the **Allow/Disallow** toggle to choose whether nodes can upgrade in failover mode or not.
8. Click **Next – Review and Confirm** to go to the review screen. Check the schedule details are correct.

   Note: To change schedule details, click Back – Schedule Upgrade, all local data is preserved while you change parameters on previous screens.

9. Select **Confirm**, and type **Yes** at the prompt, then click **Confirm** to create the task.
   If successful, a message displays that the task is created.

CANCEL AN UPGRADE TASK

You can cancel a task that has not yet completed the upgrade.

To cancel an upgrade task:

1. Select **Node Tools > Firmware Upgrade**.
2. In the task list, select the task name you want to cancel.

The Task Details screen displays.

3. Click the X button (top-right) to cancel the task.

> **Note:** You cannot cancel upgrade tasks that are already completed.

Limitations:

- Do not cancel an upgrade job just as it is about to begin, for example, 10 seconds before or after the start time.
- If an upgrade is cancelled while in progress, only nodes that have not yet upgraded are cancelled.

COPY A SCHEDULED TASK

You can copy a scheduled task to create a new upgrade task. The new upgrade task uses the nodes that were selected for the original task you are copying, for example to add or remove nodes from the list. You can also select different firmware or use the same firmware for the list of nodes.

> **Note:** For a task to be copied, the task must have already run or been cancelled, and display an Upgrade Status of Completed.

1. Navigate to the Node Firmware Upgrade Home screen.
   The Node Upgrade UI is available in the Web UI under **Node Tools > Firmware Upgrade**.
2. Select the task you want you copy, by clicking the task name in the task table.
3. Click the **Copy** button to copy the task.
   A new task is created with the same nodes.

DELETE AN UPGRADE TASK

Upgrades cannot be permanently deleted as they can offer a valuable insight into the health of the nodes when problem-solving and provide the version path that they have traversed over their lifetime.

If the number of jobs is becoming unmanageable, or jobs must be deleted for security measures, the support team can advise on how to remove/clear them.

RETRY AN UPGRADE TASK

If an upgrade task fails, you can retry the node upgrades, provided the firmware file is still available.

1. Navigate to the Node Firmware Upgrade Home screen.
   The Node Upgrade UI is available in the Web UI under **Node Tools> Firmware Upgrade**.
2. Select the task you want to retry, by clicking the task name in the task table.
   You can filter the task list by using the **Completed with Errors** filter.
3. In the task detail screen, click the **Repeat task** button.

   > **Note:** If the relative Firmware file has been deleted, the **Repeat task** button is disabled.

   The **Firmware Upgrade** page displays.

NODE UPGRADE RUNTIME BEHAVIOUR

The following describes the behavior of the Node Upgrade tool while performing routine upgrade tasks.

Note: If the connection to the node is interrupted during the upgrade, the upgrade may be cancelled and will fail unless the upgrade was in the final stages and had no requirement for further interaction with Lighthouse. In this scenario Lighthouse may still report the node upgrade as failed if it was unable to confirm that the upgrade succeeded due to the node being disconnected during the validation period. Failure to upgrade one node does not affect other nodes in the upgrade job.

PROMOTE A SECONDARY INSTANCE TO PRIMARY

- All scheduled upgrades are cancelled when a secondary node is promoted to be the new primary node.
- Firmware files are not replicated among the multiple instance cluster and must be re-uploaded to the new primary after promotion.

SKIP VERSIONS

If a node is scheduled to be upgraded from 21.03 directly to 22.01 (skipping 21.Q4), it will upgrade the node even if it has been manually upgraded to 21.04 before the scheduled upgrade starts.

Note: Lighthouse does not check or validate the version jumps for nodes, so there is a risk that the upgrade could fail if major versions are being skipped. Skipping versions is not recommended or supported, however, it is not disallowed.

TIME ZONES

Node upgrades can only be initiated or scheduled by an operator with administrator credentials while logged in at the primary lighthouse. The scheduling of the node upgrade is based on the time zone of the primary lighthouse.

If the time zone of the primary lighthouse is changed before a scheduled upgrade starts, the schedule time is based on the new time zone. This may result in jobs not running at all, being skipped, ignored, or otherwise running at unpredictable times.

Note: It is recommended that you avoid changing the system time of lighthouse, or its time zone, while jobs are scheduled.

OFFLINE NODES

If a node is offline or otherwise unreachable at the time of upgrade, the node is skipped.

If the node is offline there is a one minute buffer before the scheduled upgrade is skipped and Lighthouse reports the node with a failed to upgrade status.

LIGHTHOUSE AVAILABILITY AND STABILITY

Lighthouse must be online and fully booted (preferably for at least a few minutes) before the upgrade starts. It is good practice to have Lighthouse online for a few hours before a node upgrade. This ensures that all the nodes that will be upgraded have re-established their connection and allows time to troubleshoot any issues.

Do not attempt to change major Lighthouse settings, especially those involving the network or time zone, when an upgrade is underway or imminent.

Do not conduct multiple major operations on nodes simultaneously, for example, do not apply templates to the node while it is being upgraded. Do not login to a node and change settings moments before an upgrade occurs.

Note: If Lighthouse is offline when a scheduled upgrade is due to start, the upgrade is not run.

OPERATIONS NOT SUPPORTED

There are several operations that are not supported as part of the Lighthouse Node upgrade process.

UNENROLL NODES AT UPGRADE

Unenrolling nodes as they are being upgraded is not supported as this could result in unexpected behaviour.

DOWNGRADE VERSIONS

Lighthouse does not allow downgrading of nodes, nor does it allow upgrading to an identical version. The node upgrade will skip nodes that are at the upgrade version or later, for example, if upgrading from version 21.03 to 21.04, it will ignore any nodes that are already at 21.04 or 22.01.

BACK UP NODES

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

> **Note:** Node backup requires node firmware 4.6 or later.

To set up node backup:

1. Select ⚙ **> SERVICES > Node Backup**.
2. Select **Enabled** to turn on this service.
3. Under the **Storage** section, select the **Number of stored backups** you want to keep.
4. In **Retention period after unenrollment**, select *None, Days* or *Forever*.
5. Enter the **Location** you want to store the backup files.

   The default is `/mnt/nvram/`.

6. Click **Validate** to make sure the location exists and has enough space to store them.
7. Under **SCHEDULING** configure the *Start* time and *cadence* of the backups.
   a. For the **Start** time, choose either **Immediately** or choose **Set Time** to open editable **Date** and **Time** fields.
   b. Choose how often you want to **Repeat** the backup by selecting either **One Time Only** or **Interval** and configuring the settings.
8. Click **Apply**.

> **Note:** You can modify these options by returning to ⚙ **> SERVICES > Node Backup** at any time.

MANAGE PORTS

Lighthouse allows you to manage all ports connected to nodes. **Ports** displays the status and number of all the ports on Lighthouse, which you have permissions to view/edit.

VIEW PORTS FOR ALL NODES

1. Select **Ports**.

   The Ports page displays.

VIEW PORT INFORMATION FOR A NODE

1. Select **Ports**.
2. Select the **NODE NAME** for a associated with the port.

   The ports management page displays.

FILTER PORTS

1. Select **Ports**.

   The PORTS page displays.

2. The following options are available:

   - Filter Ports: create and save a new filter or select / edit an existing filter.
   - Filter Nodes: create and save a new filter or select / edit an existing filter.
   - Predefined filters: *Show All*, *Configured Ports*, *Unconfigured Ports*
   - Enter a search term to filter the results. The AND operator is used if more than one search term is entered. To enter a multi word search phrase, enclose them in double quotes.

CREATE, ASSIGN OR REMOVE A TAG

1. Select **Ports**.

   > **Note:** The following steps can also be performed by accessing port information from the **Nodes** page.

   The Ports page displays.

2. Click the **Add/Edit Tags** icon.

   The **ADD/EDIT TAGS** dialog displays.

3. Complete one of the following actions:

   - Create a new tag

     a. Click ⊕ **Create new tag**.

        A new row displays.

     b. Enter a tag name.

   - Assign an existing tag

   a. Click **Select existing tag**.

      A new row displays with the option to select an existing tag.

   b. Select the tag name.

- Remove an assigned tag

   a. Select ✕ **Remove tag**.

      The tag is removed from the list.

4. Click **Update Tags**.
   The CONFIRM CHANGES dialog displays.
5. Click **Confirm**.

ACCESS PORT LOGS

The node must be configured to display logs, and the user will have *Logging* > *Port Logging* access granted in the permissions set. Either Full Access or Read Only.

ACCESS PORT LOGS FROM THE NODES PAGE

1. Select **Nodes**.
2. Select the **NODE NAME** for a node that is enrolled and configured.
   The ports management page displays.
3. Select the **LOGS** tab.
4. Select the **PORT** from the list.
5. Filter the list as required:
   - Specify the date range.
   - Search for a log with a particular text using **Filter Log Contents**.
6. You can also download the displayed logs, or open a new window to view the logs in more detail.

ACCESS PORT LOGS FROM THE PORTS PAGE

1. Select 🖧 **Ports**.
2. Select the **NODE NAME** for a node that is enrolled and configured.

   The ports management page displays.
3. Select the **LOGS** tab.
4. Select the **PORT** from the list.
5. Filter the list as required:
   - Specify the date range.
   - Search for a log with a particular text using **Filter Log Contents**.
6. You can also download the displayed logs, or open a new window to view the logs in more detail.

Tip: You can also access the logs via the web terminal and SSH access for a configured device. To view the logs in more detail download the displayed logs, or open a new window.

MANAGE RESOURCES

Lighthouse allows users to manage resources connected to enrolled nodes that have implemented Smart Management Fabric so that resources can avail of client-less network access.

The Smart Management Fabric network can be leveraged to allow users to manually add resources that exist on those networks. Users can specify three proxy methods through Lighthouse to provide client-less network access via:

- HTTP
- HTTPS
- SSH

Notes:

- To proxy via HTTP or HTTPS to a resource, Lighthouse must be accessed through a Domain.
- Resources can only be used in a Lighthouse that has an Automation Edition subscription. However users can delete and view a resource if they downgrade to an Enterprise Edition subscription.

Operations to manage resources are on the **Resources** page. As a reminder the *Connected Resource Gateway* permissions must be set to *Full Access* as found under the Advanced Features in the Operation Permission table.

VIEW RESOURCES

1. Select **Resources**.

   The RESOURCES page displays.

FILTER RESOURCES

1. Select **Resources**.

   The RESOURCES page displays.
2. The following options are available:
   - Filter Resources: create and save a new filter or select / edit an existing filter.
   - Connectivity: *Any Status*, *Connected*, *Disconnected*.
     - *Connected*: at least one proxy protocol can be connected.
     - *Disconnected*: all proxy protocols are disconnected.
   - Enter a search term to filter the results. The AND operator is used if more than one search term is entered. To enter a multi word search phrase, enclose them in double quotes.

CREATE, ASSIGN OR REMOVE A RESOURCE TAG

1. Select **Resources**.

   The RESOURCES page displays.

2. Select the name of resource from the grid.

   The **EDIT RESOURCE** dialog displays.

3. Under the TAGS section of the dialog, complete one of the following actions:

   - Create a new tag

     a. Click ⊕ **Create new tag**.

        A new row displays.

     b. Enter a tag name.

   - Assign an existing tag

     a. Click **Select existing tag**.

        A new row displays with the option to select an existing tag.

     b. Select the tag name.

   - Remove an assigned tag

     a. Select ✕ **Remove tag**.

        The tag is removed from the list.

4. Click **Update Resource**.

CONNECT TO A RESOURCE

1. Select **Resources**.

   The RESOURCES page displays.

2. For a connected resource select an icon in the actions column to access via:

- HTTP
- HTTPS
- SSH

Note: For SSH connections Lighthouse will not know the username for the target resource. When selecting SSH, a modal will appear to update the Target Resource Username. You may leave it blank and the user will be prompted to enter the Username when using SSH. This field will default to the Username of the current login session.

For an IP connection there are three different states that the connection can be within. The state can be seen when hovering over each one of the proxy access button for each resource.

Connection State	Description
Pending	When a resource is added for the first time, the state will come back as pending.
Connected	When Lighthouse can validate the connectivity via that specific protocol on the specified port, the hover state displays a timestamp of when the last time it was validated.
Failure	When Lighthouse can not validate the connectivity via that specific protocol on the specified port.

LOGGING

Connected Resource Gateway logging can be found in a file located in `/var/log` called `crg.log`. This file captures the following information:

- If a Lighthouse user is granted access to a resource via HTTP, HTTPS or SSH.
- If a Lighthouse user is rejected access to a resource due to Connected Resource Gateway permission being set to Deny.

- Unauthenticated users that try to access a resource are sent to an NGINX error page, which is captured in the existing NGINX access logs.

ADD A NEW RESOURCE

> **Note:** A user with role permissions including *Full Access* set for *Advanced Features > Connected Resource Gateway* can add a new resource.

To add a new resource:

1. Select **Resources**.

   The **RESOURCES** page displays.
2. Click the ⊕ **New Resource** button.

   The **NEW RESOURCE** dialog displays.
3. Enter a **Name** and **Network Address**.
4. Select the service from *HTTP*, *HTTPS* or *SSH* and the port number or that service. For example, 80 is the default port for the HTTP service.
5. Under the TAGS section create, assign or remove a resource tag.
6. Click **Add Resource**.

BULK ADDITION OF RESOURCES

The Lighthouse UI *does not* allow bulk addition of resources and is performed one at a time through the UI. Users can use the REST APIs to add multiple resources into the catalog. Refer to the `/smf_resources/bulk POST` request.

EDIT A RESOURCE

1. Select **Resources**.

   The **RESOURCES** page displays.

2. Click the **Resource Name** of the resource.

   The **EDIT RESOURCE** dialog displays.

3. Edit the fields as required.

4. Under the TAGS section create, assign or remove a resource tag.

5. Click **Update Resource**.

BULK UPDATE OF RESOURCES

The Lighthouse UI *does not* allow bulk updating of resources and is performed one at a time through the UI. Users can use the REST APIs to update multiple resources into the catalog. Refer to the `/smf_resources/bulk PUT` request.

DELETE A RESOURCE

1. Select **Resources**.

   The **RESOURCES** page displays.

2. Click the **Resource Name** of the resource.

   The **EDIT RESOURCE** dialog displays.

3. Click the **Delete** button.

   A confirmation message displays.

4. Click **Delete**.

   The resource is deleted and a success notification displays.

BULK DELETION OF RESOURCES

1. Select **Resources**.

   The **RESOURCES** page displays.

2. Select the checkbox for each resource to delete.

3. Click the 🗑 **Delete Selected** button.

   A confirmation message displays.

4. Click **Delete**.

   The resource(s) is deleted and a success notification displays.

SET UP LIGHTHOUSE AS A CONNECTED RESOURCE GATEWAY (CRG)

Notes: Opengear is transitioning from Automation Gateway (AG) to Connected Resource Gateway (CRG) for access, control, and management of connected resources:

- CRG will be the primary solution moving forward, offering improved scalability, security, and continued enhancements.
- New users should begin with CRG as the recommended option, while existing AG users are encouraged to migrate to CRG.
- AG discovered resources are not automatically accessible on CRG.
- To migrate to CRG, you must set up CRG as if you are setting up a new implementation.

Prerequisites:

- The external DNS configuration must have the entries to point to the Lighthouse:
  - To register in the DNS, ensure that you use the address format: *.crg.your1stlighthouseaddress.com *.crg.your2ndlighthouseaddress.com.
  - You only require DNS if you want to proxy the GUI for http/https.
- Smart Management Fabric. If this is already set up, you can start at step 7.

1. **Optional:** Create and upload the Lighthouse SSL Certificate and ensure that it is valid for all sub-domains.

> **Note:** For Connected Resource Gateway (CRG), add the following subdomain patterns to your certificate:
> *.crg.your1stlighthouseaddress.com
> *.crg.your2ndlighthouseaddress.com

2. **Optional:** To add any resources via the domain name for the resource (rather than IP address), ensure that Lighthouse is configured to use an appropriate DNS server. If Lighthouse:
   - gets its primary IP address via DHCP, configure the DHCP server to specify the DNS server to use.
   - uses a static IP address, edit the configuration for that connection on the **Interfaces** page and ensure that a DNS server address is configured.

   > **Note:** If you only want to add resources via IP address, you can skip this step.

3. Enable Smart Management Fabric.
4. Create a Smart Management Fabric Template for enrolling nodes.
   You can apply this template either:
   - on enrollment via bundle.
   - at any time by pushing a template.

   > **Notes:**
   > - If you are completing these steps for third-party nodes, then you do not require a SMF template and can skip this step.
   > - If you have existing nodes, and you push the SMF template, then steps 5 and 6 are not required.

5. Create an Enrollment Bundle and ensure that the Smart Management Template is linked to the bundle under the **Bundle Templates** section.

6. Enroll the node.

   The node initially shows as enrolled with no templates applied.

   The linked template is pushed and applied to the node.

7. Add a resource.

   Notes:

   - Lighthouse attempts to discover the configured routes, to ensure they exist, every 60 seconds. During this time, the following toast message may appear: '*The provided address is not an SMF discovered subnet*.' This message also appears if there is no route in SMF to your device; check that you added a node near that network and configured SMF on that node (via template) correctly.
   - Lighthouse now polls to check connectivity to the resource. When established, the appropriate HTTP, HTTPS and SSH icons are enabled.

8. Click the appropriate icon to connect to the resource.

MANAGING FILTERS AND TAGS

To provide clear and customized access to nodes, Lighthouse uses search expressions called filters which allow properties and user-supplied tags, consisting of a name and value, to be compiled into a search expression.

These filters can be applied to Nodes, Ports, and Resources.

Lighthouse also supports the management of customized tags that can be applied to Nodes, Ports, and Resources. These tags can be applied within the custom filters.

MANAGE FILTERS

VIEW FILTERS

To view a list of filters available:

1. Select ⚙ **> FILTERS AND TAGS > Filters**.

   The **FILTERS** page displays defaulted to the **NODE FILTERS** tab.

2. Select the:

   - **NODE FILTERS** tab to display a grid of available filters that can be applied when viewing the **Nodes** page.

   - **PORT FILTERS** tab to display a grid of available filters that can be applied when viewing the **Ports** page.

   - **RESOURCE FILTERS** tab to display a grid of available filters that can be applied when viewing the **Resources** page.

MANAGE NODE FILTERS

CREATE A NODE FILTER

CREATE FROM THE FILTER MANAGEMENT PAGE

1. Select ⚙ **> FILTERS AND TAGS > Filters**.
2. Click the ⊕ **New Filter** button.

   The **PICK A FILTER TYPE** selection displays.
3. Select **Node Filter**.

   The **NEW NODE FILTER** dialog displays.
4. Enter a **Node Filter Name**.
5. For each filter criteria:
   a. Select the *Field to search* from.
   b. Select the *Operator*.
   c. *Enter the Value matched on.*
6. To enter additional search parameters, click ⊕ **Add Criteria**.
7. The option to select the Boolean operator (AND / OR) is displayed. The default selected option is **AND**.
   1. Select **AND** if the filter must match all values.
   2. Select **OR** if one or the other values must match.
8. Enter the additional details in the *Field to search*, *Operator* and *Value* fields.

   > **Note:** Click the ✕ icon to remove additional search parameters as necessary.
9. Click **Add Filter**.

CREATE FROM THE NODE FILTER CONTROL

1. Select **Nodes**.
2. Click the **Filter Nodes** control.
3. Click the **Select Node Filter** control.
4. Select **New Node Filter**.

   The filter criteria section displays.
5. Enter the criteria.
6. To enter additional search parameters, click ⊕ **Add Criteria**.
7. Either:
   1. Apply without saving.
      1. Click **Apply**.

         The filter control updates to show **Filter Nodes: New Node Filter *** indicating a filter has been applied but has not been saved.
   2. Save the filter.
      1. Click the **Save node filter** icon.

         The **SAVE NODE FILTER** dialog displays.
      2. Enter the **Node Filter Name**.
      3. Review / update the filter criteria.
      4. Click **Save and Apply**.

         The filter control updates to show **Filter Nodes:** <*Name of Filter*> indicating a filter has been applied.

Note: This is available on the Nodes > PENDING tab as well.

EDIT A NODE FILTER

EDIT FROM THE FILTER MANAGEMENT PAGE

1. Select ⚙ > **FILTERS AND TAGS > Filters**.
2. Click the **Name** of a node filter.

   The **EDIT NODE FILTER** dialog displays.
3. Update the name and/or the filter criteria.
4. Click **Update Filter**.

EDIT FROM THE NODE FILTER CONTROL

1. Select ▤ **Nodes**.
2. Click the **Filter Nodes** control.
3. Click the **Select Node Filter** control.
4. Select the filter to edit.
5. Click the ✎ **Edit node filter** icon.

   The **EDIT NODE FILTER** dialog displays.
6. Update the name and/or the filter criteria.
7. Click **Save and Apply**.

   The filter control updates to show **Filter Nodes:** <*Name of Filter*> indicating a filter has been applied.

DELETE A NODE FILTER

DELETE FROM THE FILTER MANAGEMENT PAGE

1. Select ⚙ **> FILTERS AND TAGS > Filters**.
2. Click the **Name** of a node filter.

   The **EDIT NODE FILTER** dialog displays.
3. Click 🗑 **Delete**.

   A message to confirm the action displays.
4. Click **Delete**.

   The filter is deleted and a success notification displays.

DELETE FROM THE NODE FILTER CONTROL

1. Select 🖴 **Nodes**.
2. Click the **Filter Nodes** control.
3. Click the **Select Node Filter** control.
4. Select the filter to edit.
5. Click the ✎ **Edit node filter** icon.

   The **EDIT NODE FILTER** dialog displays.
6. Click 🗑 **Delete**.

   A message to confirm the action displays.
7. Click **Delete**.

   The filter is deleted and a success notification displays.

MANAGE PORT FILTERS

CREATE A PORT FILTER

CREATE FROM THE FILTER MANAGEMENT PAGE

1. Select ⚙ **> FILTERS AND TAGS > Filters**.
2. Click the ⊕ **New Filter** button.

   The **PICK A FILTER TYPE** selection displays.
3. Select **Port Filter**.

   The **NEW PORT FILTER** dialog displays.
4. Enter a **Name** .
5. For each filter criteria:
   a. Select the *Field to search* from.

   b. Select the *Operator*.

   c. *Enter the Value matched on.*
6. To enter additional search parameters, click ⊕ **Add Criteria**.
7. The option to select the Boolean operator (AND / OR) is displayed. The default selected option is **AND**.
   1. Select **AND** if the filter must match all values.
   2. Select **OR** if one or the other values must match.
8. Enter the additional details in the *Field to search*, *Operator* and *Value* fields.

   > **Note:** Click the ✕ icon to remove additional search parameters as necessary.
9. Click **Add Filter**.

CREATE FROM THE PORT FILTER CONTROL

1. Select **Ports**.
2. Click the **Filter Ports** control.
3. Click the **Select Port Filter** control.
4. Select **New Port Filter**.

   The filter criteria section displays.
5. Enter the criteria.
6. To enter additional search parameters, click ⊕ **Add Criteria**.
7. Either:
   1. Apply without saving.
      a. Click **Apply**.

         The filter control is updates to show **Filter Ports: New Port Filter *** indicating a filter has been applied but has not been saved.
   2. Save the filter.
      a. Click the **Save port filter** icon.

         The **SAVE PORT FILTER** dialog displays.

      b. Enter the **Port Filter Name**.

      c. Review / update the filter criteria.

      d. Click **Save and Apply**.

         The filter control updates to show **Filter Ports:** <*Name of Filter*> indicating a filter has been applied.

EDIT A PORT FILTER

EDIT FROM THE FILTER MANAGEMENT PAGE

1. Select ⚙ **> FILTERS AND TAGS > Filters**.
2. Select the **PORT FILTERS** tab.
3. Click the **Name** of a port filter.

   The **EDIT PORT FILTER** dialog displays.
4. Update the name and/or the filter criteria.
5. Click **Update Filter**.

EDIT FROM THE PORT FILTER CONTROL

1. Select **Ports**.
2. Click the **Filter Ports** control.
3. Click the **Select Port Filter** control.
4. Select the filter to edit.
5. Click the ✎ **Edit port filter** icon.

   The **EDIT PORT FILTER** dialog displays.
6. Update the name and/or the filter criteria.
7. Click **Save and Apply**.

   The filter control updates to show **Filter Nodes:** <*Name of Filter*> indicating a filter has been applied.

DELETE A PORT FILTER

DELETE FROM THE FILTER MANAGEMENT PAGE

1. Select ⚙ **> FILTERS AND TAGS > Filters**.
2. Select the **PORT FILTERS** tab.
3. Click the **Name** of a port filter.

   The **EDIT PORT FILTER** dialog displays.
4. Click 🗑 **Delete**.

   A message to confirm the action displays.
5. Click **Delete**.

   The filter is deleted and a success notification displays.

DELETE FROM THE PORT FILTER CONTROL

1. Select **Ports**.
2. Click the **Filter Ports** control.
3. Click the **Select Port Filter** control.
4. Select the filter to edit.
5. Click the ✏ **Edit port filter** icon.

   The **EDIT PORT FILTER** dialog displays.
6. Click 🗑 **Delete**.

   A message to confirm the action displays.
7. Click **Delete**.

   The filter is deleted and a success notification displays.

MANAGE RESOURCE FILTERS

CREATE A RESOURCE FILTER

CREATE FROM THE FILTER MANAGEMENT PAGE

1. Select ⚙ **> FILTERS AND TAGS > Filters**.
2. Click the ⊕ **New Filter** button.

   The **PICK A FILTER TYPE** selection displays.
3. Select **Resource Filter**.

   The **NEW RESOURCE FILTER** dialog displays.
4. Enter a **Name** .
5. For each filter criteria:
   a. Select the *Field to search* from.
   b. Select the *Operator*.
   c. *Enter the Value matched on.*
6. To enter additional search parameters, click ⊕ **Add Criteria**.
7. The option to select the Boolean operator (AND / OR) is displayed. The default selected option is **AND**.
   1. Select **AND** if the filter must match all values.
   2. Select **OR** if one or the other values must match.
8. Enter the additional details in the *Field to search*, *Operator* and *Value* fields.

   > **Note:** Click the ✕ icon to remove additional search parameters as necessary.
9. Click **Add Filter**.

CREATE FROM THE RESOURCE FILTER CONTROL

1. Select **Resources**.
2. Click the **Filter Resources** control.
3. Click the **Select Resource Filter** control.
4. Select **New Resource Filter**.

   The filter criteria section displays.
5. Enter the criteria.
6. To enter additional search parameters, click ⊕ **Add Criteria**.
7. Either:
   1. Apply without saving.
      1. Click **Apply**.

         The filter control is updates to show **Filter Resources: New Resource Filter *** indicating a filter has been applied but has not been saved.
   2. Save the filter.
      1. Click the **Save resource filter** icon.

         The **SAVE RESOURCE FILTER** dialog displays.
      2. Enter the **Resource Filter Name**.
      3. Review / update the filter criteria.
      4. Click **Save and Apply**.

         The filter control updates to show **Filter Resources:** <*Name of Filter*> indicating a filter has been applied.

EDIT A RESOURCE FILTER

EDIT FROM THE FILTER MANAGEMENT PAGE

1. Select ⚙ **> FILTERS AND TAGS > Filters**.
2. Select the **RESOURCE FILTERS** tab.
3. Click the **Name** of a resource filter.

   The **EDIT RESOURCE FILTER** dialog displays.
4. Update the name and/or the filter criteria.
5. Click **Update Filter**.

EDIT FROM THE RESOURCE FILTER CONTROL

1. Select 🖳 **Resources**.
2. Click the **Filter Resources** control.
3. Click the **Select Resource Filter** control.
4. Select the filter to edit.
5. Click the ✎ **Edit resource filter** icon.

   The **EDIT RESOURCE FILTER** dialog displays.
6. Update the name and/or the filter criteria.
7. Click **Save and Apply**.

   The filter control updates to show **Filter Resources:** <*Name of Filter*> indicating a filter has been applied.

DELETE A RESOURCE FILTER

DELETE FROM THE FILTER MANAGEMENT PAGE

1. Select ⚙ **> FILTERS AND TAGS > Filters**.
2. Select the **RESOURCE FILTERS** tab.
3. Click the **Name** of a resource filter.

   The **EDIT RESOURCE FILTER** dialog displays.
4. Click 🗑 **Delete**.

   A message to confirm the action displays.
5. Click **Delete**.

   The filter is deleted and a success notification displays.

DELETE FROM THE RESOURCE FILTER CONTROL

1. Select **Resources**.
2. Click the **Filter Resources** control.
3. Click the **Select Resource Filter** control.
4. Select the filter to edit.
5. Click the ✎ **Edit resource filter** icon.

   The **EDIT RESOURCE FILTER** dialog displays.
6. Click 🗑 **Delete**.

   A message to confirm the action displays.
7. Click **Delete**.

   The filter is deleted and a success notification displays.

MANAGE TAGS

Lighthouse has two types of tags

- Node Tags: can be assigned to a node, and consist of a name and a value
- Resource Tags: can be assigned to Ports and Resources, and consist of only a name.

Both types of tags can be used in filters. Lighthouse has Node Filters, Port Filters and Resource Filters. These filters can be used to filter results on the Nodes, Ports and Resources pages, and control access to these objects by assigning the filter to a user group.

VIEW TAGS

The **Tags** page displays the status and number of all the resources on Lighthouse, which you have permissions to view/edit.

To view a list of tags available:

1. Select ⚙ **> FILTERS AND TAGS > Tags**.
   - The NODE TAGS tab displays a grid of available node tags.
   - The RESOURCE TAGS tab displays a grid of available resource tags.

MANAGE NODE TAGS

CREATE A NEW NODE TAG

To create a node tag from the Tags Management Page:

1. Select ⚙ **> FILTERS AND TAGS > Tags**.
2. Click the ⊕ **New Tag** button.

   The **PICK A TAG TYPE** selection displays.
3. Select **Node Tag**.

   The **NEW NODE TAG** dialog displays.

4. Enter a **Name**.
5. Enter a **Value**.
6. For multiple values, click the **Add Value** button and enter.
7. Click **Add Tag**.

EDIT A NODE TAG

To edit a node tag from the Tags Management Page:

1. Select ⚙ **> FILTERS AND TAGS > Tags**.
2. Click the **Name** of a node tag.

   The **EDIT NODE TAG** dialog displays.
3. Update the **Name** and/or **Values**.
4. To add additional values, click the **Add Value** button and enter.
5. To remove any values click the ✕ icon.
6. Click **Update Tag**.

DELETE A NODE TAG

To delete a node tag from the Tags Management Page:

1. Select ⚙ **> FILTERS AND TAGS > Tags**.
2. Click the **Name** of a node tag.

   The **EDIT NODE TAG** dialog displays.
3. Click 🗑 **Delete**.

   A confirm message displays.
4. Click **Delete**.

   The tag is deleted and a success notification displays.

MANAGE RESOURCE TAGS

CREATE A NEW RESOURCE TAG

To create a resource tag from the Tags Management Page:

1. Select ⚙ **> FILTERS AND TAGS > Tags**.
2. Select **RESOURCE TAGS** tab.
3. Click the ⊕ **New Tag** button.

   The **PICK A TAG TYPE** selection displays.
4. Select **Resource Tag**.

   The **NEW RESOURCE TAG** dialog displays.
5. Enter a **Name**.

   > **Note:** Add more than one name at a time by separating each name with a comma.
6. Click the **Add Tag** button.

EDIT A RESOURCE TAG

To edit a resource tag from the Tags Management Page:

1. Select ⚙ **> FILTERS AND TAGS > Tags**.
2. Select the **RESOURCE TAGS** tab.
3. Click the **Name** of a resource tag.

   The **EDIT RESOURCE TAG** dialog displays.
4. Update **Name**.
5. Click **Update Tag**.

DELETE A RESOURCE TAG

To delete a resource tag from the Tags Management Page:

1. Select ⚙ **> FILTERS AND TAGS > Tags**.
2. Select the **RESOURCE TAGS** tab.
3. Click the **Name** of a resource tag.

   The **EDIT RESOURCE TAG** dialog displays.
4. Click 🗑 **Delete**.

   A confirm message displays.
5. Click **Delete**.

   The tag is deleted and a success notification displays.

MANAGING LIGHTHOUSE USERS

Lighthouse supports locally defined users, and remote users who are authenticated and authorized by Authentication Authorization Accounting (AAA) systems such as LDAP, Radius, and TACACs+. 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.

ROLE DESCRIPTION

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.

The predefined system roles are:

Role	Description
LighthouseAdmin	The *Lighthouse Administrator* role is assigned to groups whose members are required to manage and maintain Lighthouse. Members have access to all data on the Lighthouse system and create and manage custom groups with custom permission sets.
NodeAdmin	The *Node Administrator* role is assigned to groups that are required to manage and maintain a set of Nodes. Each group with the Node Administrator role must have an associated Node Filters which is evaluated to define the set of nodes that the group members have access to.
NodeUser	The *Node User* role is assigned to groups that require access a set of nodes. Each group with the Node User role must have an associated Node Filters which is evaluated to define the set of nodes that the group members have access to. Optionally, access to the resources can be limited by associating the saved *Resource Filter* with the Node User role.
Reporter	The *Reporter* role is assigned to groups that require just read-only access across the system permission sets.

MANAGE USER GROUPS

User groups are used to grant permissions to users that are assigned to them. To add permissions to a user group, you must create a role and then assign that role to the group.

For certain actions, such as accessing a port or a resource, the access of users in the group can be limited by associating filters with the group. Users only have access to objects that are matched by their associated filters.

- A user may be a member of multiple groups, the permissions from those groups add to get the permissions of the user.
- 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.

CREATE A USER GROUP

1. Select **> USERS & ACCOUNTS > Groups and Roles**.

   The **GROUPS AND ROLES** page displays.
2. Click ⊕ **Add User Group** to the right of the page filter control.

   The **NEW GROUP** page displays.
3. Select **Enabled** to enable group.
4. Enter a group **Name** and **Description**.

   > **Note:** *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.
5. Under the **ACCESS CONTROLS** section:

   a. Select a filter from the *By Node Filter* control to restrict access to nodes that match the selected filter. Not selecting a node filter gives users access to all nodes.
   b. Select a filter from the *By Port Filter* control to restrict access to ports that match the selected filter. Not selecting a port filter gives users access to all ports.
   c. Select a filter from the *By Resource Filter* control to restrict access to resources that match the selected filter. Not selecting a resource filter gives users access to all resources.

6. Under the **ROLES** section, manage the roles assigned to the group.
   - To add a role:
     a. Click ⊕ **Add Role**.
        The **ADD ROLES** dialog displays.
     b. With the checkboxes, *check* the roles to add.
     c. Select the *role name* to view the details of the role and the *operation permissions* associated with the role. Click **< Back** to return to the ADD ROLES dialog.
     d. Click **Add**.
   - To remove a role:
     a. Click the ✕ icon for the role.
        The role is removed.
   - Each role has specific operation permissions associated with it and **CLI (Command Line Interface)** access levels for **Console Shell Access Level**, **Shell Access**, and **PM Shell Access**.
   - Click View details to see the information for each group

1. Review the **PERMISSIONS SUMMARY** section.
   This section displays how CLI permissions are derived based on the selected roles.
2. Review the **OPERATION PERMISSIONS** section.
   This section displays how operation permissions are derived based on the selected roles.
3. Click **Apply**.

AVAILABLE ROLES:

Role	Description
Lighthouse Administrator	Members of groups with this role have *Full* access to all nodes resources. The following applies to the group filters: • The filters are set to *All Nodes* and *Port Filter* set to *All Ports*. **This cannot be changed**. • Conversely if a group’s node filter is *All Nodes* and *Port Filter* is *All Ports* you can not set the group's role *Lighthouse Administrator*. **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.
NodeAdmin	Has no shell access. Has *Read Only* access to Netops Modules, all Nodes & Configuration Operations, Cell Health, Node Filters, 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.

CREATE A USER GROUP FROM AN EXISTING GROUP

1. Select 👥 **> USERS & ACCOUNTS > Groups and Roles**.

   The **GROUPS AND ROLES** page displays.

2. From the **USER GROUPS** tab, select the name of the group to use as a template.

   The **VIEW USER GROUP** page displays.
3. Click **Use as Template**.

   The **NEW GROUP** page displays populating the form with values from the group selected with the exception of the name.
4. Enter the new group **Name**.
5. Review and modify as required.
6. Click **Apply**.

EDIT A USER GROUP

1. Select **> USERS & ACCOUNTS > Groups and Roles**.

   The **GROUPS AND ROLES** page displays.
2. Select the **ROLES** tab.
3. Select the name of the role to use as a template.

   The **VIEW ROLE** page displays.
4. Click **Edit**.

   The **EDITING** role page displays.
5. Review and modify the details as required.
6. Click **Apply**.

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** is disabled - it must be enabled to take effect for remote AAA users.

DELETE A GROUP

1. Select **> USERS & ACCOUNTS > Groups and Roles**.

   The **GROUPS AND ROLES** page displays.

2. From the **USER GROUPS** tab, select the name of the group you want to delete.

   The **VIEW USER GROUP** page displays.

3. Click **Delete Group**.

   The **CONFIRM GROUP DELETION** dialog displays.

4. Click **Delete**.

MANAGE ROLES

A user is added to a user group for which one or many roles are assigned. Roles are used to configure the level of permissions to the CLI and the set of features to which a user has access.

CLI PERMISSIONS REFERENCE

Permission	Options	Description
Console Shell Access Level	Admin | Standard | Disabled	Ability to connect to nodes' command lines via Lighthouse's SSH. **Standard** allows access the node's console with the same username as the Lighthouse user, if the user has an existing user account of the same name on the node. The user will be asked for the password of their account on the node whilst connecting. **Admin** will access the node's console as the node's root user. Lighthouse will authenticate the connection to the node using it's own ssh key, the user will not have to enter a password.
Shell Access	Enabled | Disabled	Ability to access the Lighthouse command line as an administrator.

Permission	Options	Description
PM Shell Access	Enabled | Disabled	Ability to connect to serial ports via SSH.

OPERATION PERMISSIONS REFERENCE

Feature Set	Feature	Description
Actions	Events	Ability to enable or disable if events are used to generate notifications.
	Subscriptions	Ability to manage third-party access to events.
Logging	Port Logging	Currently allows access to delete port logs through the API. Other port logging functionality is currently available through the CLI on Lighthouse. Any additional API port logging functionality will be accessible via this permission.
	Syslog	Allows managing the system syslog settings through the /system/logging endpoint, currently this functionality is not available via the UI. There is another permission (Services > Syslog) which allows the user to setup remote syslog servers for Lighthouse to send logs to.
Netops	Netops Modules	Allows configuring Netops modules, this includes the ability to use each module, set each module to always deploy, and redeploy modules. Installing and updating the modules is handled under the Services > NetOps permission.
Advanced Features	Smart Management Fabric	Allows enabling Smart Management Fabric on Lighthouse and setting the internal area ID. Enabling this permission requires “Multiple Instance”.

Feature Set	Feature	Description
		Enabling Smart Management Fabric on the nodes requires the Nodes and Configuration > Template Push permission. Configuring the Smart Management Fabric Network Range on the Lighthouse VPN, requires “Full Access” on Services > LHVPN.
	Connected Resource Gateway	Allows managing resources through Connected Resource Gateway. To read or change tags on resources, you require the appropriate level on *Tags*. To read or change resource filters, you require the appropriate level on *Filters*.
Nodes and Configuration	Nodes and Devices (Base)	Access to dashboard, nodes, ports, node enrollment and node web UI. **Read Only** will allow you to view nodes and ports, make searches for ports, and view node and port filters. **Full Access** will allow you to do Lighthouse driven node enrollments, and approve nodes that are in a pending state. You can also unenroll nodes.
	Nodes and Devices (Advanced)	Extends Nodes and Devices (Base) permissions. **Read Only** allows access to cell health information, and node connection information. **Full Access** allows for changing the subscription associated with a node
	Nodes and Firmware Management	Ability to manage node firmware uploads and schedule node upgrades.
	Template Push	Ability to push templates to nodes and manage templates.

Feature Set	Feature	Description
Service Settings	LHVPN	
	Cell Health	
	Console Gateway	
	Date & Time	
	HTTPS	
	Netops	Ability to install Netops modules and modify local Netops repositories.
	Node Backup	
	Session Settings	
	SNMP	
	SSH	
	Syslog	
Filters and Tags	Bundles	Ability to manage bundles.
	Filters	Allows for the management and use of filters.
	Tags	Allows for the management and use of tags.

Feature Set	Feature	Description
System	Admin and Subscriptions	Ability to manage access settings for Lighthouse and manage subscription details.
	Backup and Restore	
	Jobs	
	Multi-instance	Ability to manage multi-instance settings and control state of instances
	Network Interfaces	Ability to manage network interface settings
	System Upgrade and Reset	
User and Permissions	Authentication	Ability to manage authentication settings including methods of authentication, policy and restrictions.
	Group and Roles	Ability to create and edit roles and groups, but not the ability to assign them to users.
	Users	Ability to view and manage users, including creation and removal of users.

CREATE A ROLE

1. Select **> USERS & ACCOUNTS > Groups and Roles**.

   The **GROUPS AND ROLES** page displays.

2. Select the **ROLES** tab.

3. Click ⊕ **Add User Role** on right of the page filter control.

   The **NEW ROLE** page displays.

4. Enter a role **Name** and **Description**.

5. Modify the **CLI PERMISSIONS**. Select *Enabled* or *Disabled* for each of the following:

   - *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. Modify the **OPERATION PERMISSIONS** to specify the access level for each feature (FULL ACCESS | READ ONLY | DENY).

   a. Click ⌄ to expand the FEATURE SET

   b. For the FEATURE select the access level to apply for the role.

   > **Note:** The default access level is set to DENY.

7. Click **Apply**.

CREATE A NEW ROLE FROM AN EXISTING ROLE

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

1. Select 👥 **> USERS & ACCOUNTS > Groups and Roles**.

   The **GROUPS and ROLES** dashboard displays defaulted to the **USER GROUPS** tab.

2. Click the **ROLES** tab.

3. Select the role you want to copy from the list.

4. Click **Use as template**.

   The **Create Role** tab displays with all the settings of the existing role.

5. Make changes if necessary.
6. Click **Save Role** to create the new role.

CREATE A ROLE FROM AN EXISTING ROLE

1. Select > **USERS & ACCOUNTS > Groups and Roles**.

   The **GROUPS AND ROLES** page displays.
2. Select the **ROLES** tab.
3. Select the name of the role to use as a template.

   The **VIEW ROLE** page displays.
4. Click **Use as Template**.

   The **NEW ROLE** page displays populating the form with values from the role selected with the exception of the name.
5. Enter the new role **Name**.
6. Review and modify as required.
7. Click **Apply**.

EDIT A ROLE

1. Select > **USERS & ACCOUNTS > Groups and Roles**.

   The **GROUPS AND ROLES** page displays.
2. Select the **ROLES** tab.
3. Select the name of the role you want to edit.

   The **VIEW ROLE** page displays.
4. Click ✎ **Edit**.

   The **EDITING** role page displays.

5. Review and modify the details as required.
6. Click **Apply**.

DELETE A ROLE

1. Select 👥 **> USERS & ACCOUNTS > Groups and Roles**.

   The **GROUPS AND ROLES** page displays.
2. Select the **ROLES** tab.
3. Select the name of the role you want to delete.

   The **VIEW ROLE** page displays.
4. Click 🗑 **Delete Role**.

   The **CONFIRM ROLE DELETION** dialog displays.
5. Click **Delete**.

MANAGE USERS

You can create new users, edit existing users, delete users, and alter groups and permissions. Users can be either local users or remote users, in both instances you must understand how users must be authenticated.

AUTHENTICATION MODES

Authentication	Password And Group Source	Notes
Local	Authentication: Local only; Groups: Local	All users must exist locally before they can log in.
[AAA] - Radius	Authentication: the username/password provided by the user is ONLY tested	If there is a local user with the same username as the [AAA] user and that

Authentication	Password And Group Source	Notes
- Tacacs+ - Ldap	against the [AAA] server. Groups: Union of the user’s local groups and their [AAA] groups. If the user didn’t exist locally and successfully authenticated via [AAA], the user is also added to the `netgrp` group.	user tries to login with the local password, login will be denied UNLESS the local password is the SAME as the remote password, that is, the remote password is used to login. If the [AAA] server is unreachable, the only user that can authenticate locally is `root.`
Local[AAA] - LocalRadius - LocalTacacs+ - LocalLdap	Authentication: The username/password provided by the user is first tested locally and if local authentication fails then the [AAA] server is used. Groups: Union of the user’s local groups and their [AAA] groups. If the user didn’t exist locally and successfully authenticated via [AAA], then the user is also added to the netgrp group.	Basically, the user can log in with either their local password (if the user exists locally) or their [AAA] password (if the user exists in the [AAA] server). The main point is that the username/password is tested locally first and if it fails, [AAA] auth is attempted with the same username and password.
[AAA]Local - RadiusLocal - Tacacs+Local - LdapLocal	Authentication: The username/password provided by the user is first tested by the [AAA] server and if [AAA] authentication fails then the credentials are tested locally. Groups: Union of the user’s local groups and their [AAA] groups. If the user didn’t exist locally and successfully authenticated via [AAA], then the user is also added to the netgrp group.	Basically, the user can log in with either their local password (if the user exists locally) or their [AAA] password (if the user exists in the [AAA] server). The main point is that the username/password is tested by [AAA] first and if it fails, local auth is attempted with the same username and password.
[AAA]DownLocal - RadiusDownLocal	Authentication: Local authentication is ONLY used if the [AAA] server is unreachable. Otherwise [AAA] authen-	This should behave exactly the same as the [AAA] mode until the [AAA] server is unreachable at which point,

Authentication	Password And Group Source	Notes
- Tacacs+DownLocal - LdapDownLocal	tication is always used. Groups: Union of the user's local groups and their [AAA] groups. If the user didn't exist locally and successfully authenticated via [AAA], then the user is also added to the netgrp group.	local authentication is attempted.

> **Note:** The root user can be authenticated by AAA but it will always try local auth for the root user first.

CREATE A LOCAL USER

1. Select 👥 **> Local Users**.

   The **LOCAL USERS** page displays.
2. Click ⊕ **Add User**.

   The **NEW USER** page displays.
3. Select the **User Status**.
4. Enter a **Username**.

   > **Note:** Username must only contain lowercase letters, numbers and _ . -
5. Enter a **Description**.
6. Enable or disable the Remote Password Only feature.
   - Select *Enabled* for **Remote Password Only** to use external AAA server for password.
   - Select *Disabled* for **Remote Password Only** to create password for the user.
     a. Enter the **Password**.
     b. Re-enter to **Confirm Password**.

7. To add a SSH Key:
   a. Click ⊕ **Add SSH Authentication Key**.

      The **ADD SSH AUTHENTICATION KEY** dialog displays.

   b. Enter the SSH public key for the user.
   c. Click **Add Key**.
8. Under the **GROUPS** section, manage the user groups the user will be assigned.
   a. Click ⊕ **Add Group**,

      The **ADD GROUPS** dialog displays.

   b. With the checkboxes, *check* groups to add the user to.
   c. Select the *role name* to view the details of the role and the *operation permissions* associated with the role. Click **< Back** to return to the ADD ROLES dialog.
   d. Click **Add**.
9. Review the **PERMISSIONS SUMMARY** section.

   This section displays how permissions are derived based on the selected roles for both CLI and OPERATION permissions.
10. Click **Apply**.

CREATE USER EVENT LOGS

When a new user is created, an entry is added to the syslog that indicates the name of the new user, 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. To access the syslog from Lighthouse, click ? **> Generate Technical Support Report**.

EDIT A USER

1. Select 👥 **> USERS & ACCOUNTS > Local Users**.

   The **LOCAL USERS** page displays.
2. Select the name of the user to edit.

   The **EDIT USER** page displays.
3. Review and modify the details as required.
4. Click **Apply**.

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

DISABLE A USER

DISABLE A ROOT USER

> **Caution:** Make sure that another user exists that is in a group that has the **Lighthouse Administrator** role.

1. Select 👥 **> USERS & ACCOUNTS > Local Users**.

   The **LOCAL USERS** page displays.
2. Click the ⊘ **Disable User** icon button in the *Actions* section for the root user.

   A confirmation dialog displays with the following message:

   ```
   Warning
   This may prevent configuration of the system. Make sure another user exists
   that is in a group that has the "Lighthouse Administrator" role, before
   proceeding with this action.
   ```
3. Click **Confirm**.

To re- enable the root user, login as another user with Lighthouse Administrator role and enable access for root user from Actions section on Users page.

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.

DISABLE A SINGLE USER

1. Select 👥 **> USERS & ACCOUNTS > Local Users**.

   The **LOCAL USERS** page displays.
2. Click the ⊘ **Disable User** icon in the *Actions* section for the user.

   A confirmation dialog displays.
3. Click **Confirm**.

DISABLE MULTIPLE USERS

1. Select 👥 **> USERS & ACCOUNTS > Local Users**.

   The **LOCAL USERS** page displays.

2. Check the users you want to disable.
3. Click ⊘ **Disable User** from the menu above the user grid.

   A confirmation dialog displays.
4. Click **Confirm**.

ENABLE A USER

ENABLE A ROOT USER

> **Note:** To re-enable the root user, login as another user with Lighthouse Administrator role and enable access for root user from Actions section on Users page.

1. Select **> USERS & ACCOUNTS > Local Users**.

   The **LOCAL USERS** page displays.
2. Click the ⏻ Disable User icon in the *Actions* section for the root user.

   A confirmation dialog displays.
3. Click **Confirm**.

ENABLE A SINGLE USER

1. Select **> USERS & ACCOUNTS > Local Users**.

   The **LOCAL USERS** page displays.
2. Click the ⏻ Enable User in the *Actions* section for the user.

   A confirmation dialog displays.
3. Click **Confirm**.

DELETE A USER

TO DELETE A SINGLE USER

1. Select 👥 **> USERS & ACCOUNTS > Local Users**.

   The **LOCAL USERS** page displays.
2. Click the 🗑 Delete User icon in the *Actions* section of the user.

   A confirmation dialog displays.
3. Click **Confirm**.

   The user is permanently deleted.

DELETE MULTIPLE USERS

1. Select 👥 **> USERS & ACCOUNTS > Local Users**.

   The **LOCAL USERS** page displays.
2. Check the users you want to disable.
3. Click 🗑 **Delete Users** from the menu above the user grid.

   A confirmation dialog displays.
4. Click **Confirm**.

   The users are permanently deleted.

UNLOCK A USER

For a user that fails the login restrictions policy and has become locked out, a notification displays with the following message:

```
Error
Your account has been locked out. Please try again later or contact your
administrator.
```

An administrator can unlock the account with the following steps:

1. Select **> USERS & ACCOUNTS > Local Users**.

   The **LOCAL USERS** page displays.

2. Click the Unlock User icon in the **Actions** section for the locked user.

   The **UNLOCK USER** confirmation dialog displays.

3. Click **Confirm**.

EXPIRE A USER PASSWORD

EXPIRE A USER PASSWORD

You can set a user password to expire. The next time this user logs in, the user will be required to change the password.

1. Select **> USERS & ACCOUNTS > Local Users**.

   The **LOCAL USERS** page displays.

2. Click the Expire Password icon in the **Actions** section for the locked user.

   The **EXPIRE PASSWORD** confirmation dialog displays.

3. Click **Confirm**.

RESTORE A USER PASSWORD

1. Select **> USERS & ACCOUNTS > Local Users**.

   The **LOCAL USERS** page displays.

2. Click the Restore Password icon in the **Actions** section for the user with a password set to expired.

   The **RESTORE PASSWORD** confirmation dialog displays.

3. Click **Confirm**.

MANAGE LOCAL AUTHENTICATION POLICY

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.

Lighthouse Administrators can set Password Policies to ensure that users set secure passwords.

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

SET THE PASSWORD POLICY

1. Select 👥 **> USERS & ACCOUNTS > Local Authentication Policy**.

   The **LOCAL AUTHENTICATION POLICY** page displays.

2. Select **Enabled** to enable the password policy.
3. Modify the **PASSWORD REQUIREMENTS**. Check to enable one or more of the following options:
   - Minimum password length. Enter a value 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. Select *Always* or *Days* and set the number of days between reuse.

- Set password expiry. Set the number of days until passwords expire. At next login, the user must reset the password.

4. Click **Apply**.

SET THE LOGIN RESTRICTIONS

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

> **Caution:** Enabling login restrictions can cause the system to be inaccessible in an emergency.

1. Select 👥 **> USERS & ACCOUNTS > Local Authentication Policy**.

   The **LOCAL AUTHENTICATION POLICY** page displays.
2. Select the **LOGIN RESTRICTIONS** tab.
3. Select **Enabled** to enable the login restriction policy.
4. Enter a value for **Maximum attempts** to set the number of attempts a user can enter an incorrect password before being locked out.
5. Enter a value for **Lockout period** to set the number of minutes until a user can try to login again after reaching maximum incorrect login attempts.
6. Click **Apply**.

CONFIGURE AUTHENTICATION

Lighthouse supports three Authentication Authorization and Accounting (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.

LDAP CONFIGURATION

1. Select **> USERS & ACCOUNTS > Remote Authentication**.

   The **REMOTE AUTHENTICATION** page displays.

2. From the *Scheme* options, select **LDAP**.
3. Select the required **Mode**:
   - LDAPDownLocal
   - LDAP Mode
   - LocalLDAP
   - LDAPLocal
4. Enter the **Address** and optionally the **Port** of the remote authentication server to query. The port setting defaults to LDAP/LDAPS standard ports if not entered.

   Note: Click ⊕ **Add Authentication Server** to add multiple servers. The LDAP subsystem queries them in a round-robin fashion.

5. Enter 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`.
6. Enter 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.
7. Enter and confirm the *Bind DN Password* for the binding user.

8. Enter the **LDAP username** attribute. This depends on the underlying LDAP system.
   Use **sAMAccountName** for Active Directory systems, and **uid** for OpenLDAP based systems.
9. Enter the **LDAP group membership** attribute.
   This is only required for Active Directory and is generally **memberOf**.
10. If required, 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.
11. Under the **SSL** section, choose the required **Server protocol.**
    a. *LDAP over SSL preferred*: this will attempt LDAPS before trying LDAP without SSL
    b. *LDAP (no SSL) only*: non-SSL LDAP is always used
    c. *LDAP over SSL only*: LDAP over SSL is always used
12. Check **Ignore SSL certificate errors** to ignore any SSL certificate errors encountered when accessing LDAPS servers.
    If this option is checked, a certificate file uploaded will not be used.
13. To **UPLOAD CERTIFICATE FILE** to validate LDAPS servers, navigate to the directory containing the appropriate upgrade image file and drag and drop the image onto the target page section or click **select file** to open a dialog.

    > **Note:** Supported files: .crt, .cer, .ca-bundle, .p7b, .p7c, .p7s, .pem, .txt

14. Click **Apply**.

RADIUS CONFIGURATION

1. Select 👥 **> USERS & ACCOUNTS > Remote Authentication**.
   The **REMOTE AUTHENTICATION** page displays.
2. From the *Scheme* options, select **Radius**.

3. Choose the required **Mode**:
   - RADIUSDownLocal
   - RADIUS Mode
   - LocalRADIUS
   - RADIUSLocal
4. Enter the *Address* and optionally the *Port* of the **Remote Authentication Server** to query. The default port is 1812.

   > **Note:** Click ⊕ **Add Authentication Server** to add multiple servers.

5. Enter the *Address* and optionally the *Port* of the **Remote Accounting Server** to send accounting information to. The default port is 1813.

   > **Note:** Click ⊕ **Add Accounting Server** to add multiple servers.

6. Enter and confirm the *Server Password* also known as the RADIUS Secret.
7. Click **Apply**.

To provide group membership, RADIUS must 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.

TACACS+ CONFIGURATION

1. Select > **USERS & ACCOUNTS > Remote Authentication**.

   The **REMOTE AUTHENTICATION** page displays.

2. From the *TACACS+* options, select **Radius**

3. Choose the required **Mode**:

   - TACACS+DownLocal:
   - TACACS+Mode
   - LocalTACACS+
   - TACACS+Local

4. Enter the *Address* and optionally the *Port* of the **Remote Authentication Server** to query. The default port is 49.

   > **Note:** Click ⊕ **Add Authentication Server** to add multiple servers. The TACACS+ subsystem queries them in a round-robin fashion.

5. Select the **Login Method**to set the method used to authenticate to the server. Defaults to PAP. To use DES encrypted passwords, select **Login**.

6. Enter and confirm the *Server Password*, also known as the TACACS+ Secret.

7. Enter the **TACACS+ service**. This determines which set of attributes are returned by the server. Defaults to "raccess".

8. Click **Apply**.

To provide group membership, TACACS+ must be configured to provide a list of group names. The 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.

CONFIGURE SSH AUTHENTICATION

SSH authentication security settings for all users across all Lighthouses is managed under **> USERS & ACCOUNTS > SSH Authentication**.

To configure SSH Authentication:

1. Select **> USERS & ACCOUNTS > SSH Authentication**.

   The **SSH AUTHENTICATION SETTINGS** page displays.

2. Select *Enabled* for **SSH Password Authentication** to allow using a password for SSH connections for all users, including the `root` user.

   > **Note:** With *Lighthouse version 24.06.0 and later*, by default any AWS Lighthouse instance created will have this setting disabled. This setting affects every user account.
   >
   > - If disabled, Lighthouse users must use a SSH key to authenticate SSH connections to Lighthouse. This can be set under **> USERS & ACCOUNTS > Local Users** and then configuring per user.
   > - If enabled, then SSH Authentication will be authenticated via their password.

3. Select *Enabled* for **SSH connection as Root User** to allow the `root` user to SSH into Lighthouse.

> **Note:** With *Lighthouse version 24.06.0 and later*, by default any AWS Lighthouse instance created will have this setting disabled. This setting only affects the `root` user.
>
> - If disabled the `root` user will not be able to use SSH to connect to a Lighthouse Instance.
> - If enabled, the `root` user must still be enabled as well and is done by navigating to **> USERS & ACCOUNTS > Local Users** and select **Enabled** for the `root` user.

4. Click **Apply**.

SAML CONFIG 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 2024.06 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
- Auth0

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

Common values you will require 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).

GENERIC IDP SETUP

This section describes how to integrate Lighthouse with your generic Identity Provider (IdP) Application.

In case Lighthouse's supported IdPs do not include your identity provider, use the Generic IdP setup. This has been made as general as possible to meet expectations of all IdPs in the market today.

Notes:

- 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.
- The `{provider}` in the steps must exactly match one of our provider strings that is, `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
   - create a separate application integration per Lighthouse if you only access each via a single address .
6. If your IdP has the option, set the initiator to the `Service Provider`, set your IdP to sign the **Assertion** for SAML.

GENERIC IDP SAML ATTRIBUTE

You must also 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, that is, 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, that is, 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 (that is, populate `LH_Groups` to include its value).

LIGHTHOUSE IDP SETUP

You must 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 must also 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
```

EXAMPLES OF SPECIFIC IDP SETUPS

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

OKTA EXAMPLE - CREATE AN APPLICATION

You are required to create an application that Okta will be doing authentication on behalf of.

> **Note:** You must know 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: for example, Lighthouse and click **Next**.

3. For the **Single sign on URL** enter: `https://{main lighthouse address}/api/v3.7/sessions/saml/sso/okta`
    a. Select: Use this for Recipient URL and Destination URL
    b. Fill out the **Other Requestable SSO URLs** with the SSO URLs for every Lighthouse address that you want to be able to sign in with, that is, IP addresses and DNS address for both your primary and secondary Lighthouses.

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

1. Open your Okta application.
2. Go to More Actions > SAML Metadata. This is the metadata xml file that you require 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.

For example:

```
saml-idp-metadata -p {root password} creaUser groupste -m /path/to/okta_
metadata.xml -P okta -n "My Okta display name" -l {LH id number}
```

GROUPS SETUP

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

If you do not already have your own setup in Lighthouse:

1. Login to Lighthouse as a local user (or any non-SAML user) i.e. root.
2. Create the User groups with the Roles and permission that you require.
3. In Okta go to **Directory > Groups**.
4. Click **Add Group**.
5. Enter the Group name that matches a Group name on lighthouse.
6. Open your new group.
7. Go to **Manage Apps**.
8. Search for your lighthouse app and click **Assign**.
9. Click **Done**.
10. Go to **Manage People**.
11. Search for and click on the users you want 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 EXAMPLE - CREATE AN APPLICATION

You are required to create an application that Onelogin will be doing authentication on behalf of.

1. Go to **Applications > Add App >** Search for and choose `SAML Custom Connector (Advanced)`.
2. Name your connector, that is, 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. Check `Include in SAML assertion`.

d. Check `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 require 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., For example `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

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

If you do not already have your own **User Groups** setup in Lighthouse:

1. Login to Lighthouse as a local user (or any non-SAML user) for example, root.
2. Create the **User Groups** with the **Roles** and **Permissions** that you require.
3. In Onelogin Go to **Users > Roles**.
4. Click **New Role**.
   a. Set the Role's name to match the lighthouse group you want it to map to.
   b. Select your Lighthouse app to associate the role with.
   c. Click **Save**.
5. Open the newly created 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. Click **Save.**
   a. If you used a mapping then go to **Users > Mappings and run Reapply All Mappings**.
9. Click **Done**

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

AZURE EXAMPLE - ACTIVE DIRECTORY

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

To 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, for example, Lighthouse, then click **Create**.
7. Click **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:

- Name: `LH_Groups`
- 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. Navigate to **3. SAML Signing Certificate** and 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 as follows:
   For example, `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

After this initial setup, you will be able to login as a SAML user. If you do not already have your own User groups setup in Lighthouse, you can set them up as follows:

1. Login to Lighthouse as a local user (or any non-SAML user) i.e. `root`.
2. Create the User groups with the Roles and permission required.

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 required.
6. Go to **Azure Active Directory**.
   a. Go to **Enterprise applications**.
   b. Open your App, that is, Lighthouse.
   c. Go to **Users and groups**.
   d. Click **Add user/group**.
   e. 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.

CONFIGURE AUTH0 FOR IDP

Lighthouse can be added as an **Enterprise application** to AUTH0. This example uses “App roles” to grant users permissions.

CREATE AN APPLICATION (ENTERPRISE APPLICATIONS)

1. Go to **Auth0**.
2. Go to **Applications > Application**.
3. Click **Create application**.
   a. Select **Regular Web Application.**
   b. Name the application, for example, Lighthouse.
4. Go to **Settings** tab.
   a. Select **SAML**.
   b. Set Application Login URI to:

   ```
   https://{main lighthouse address}/api/v3.7/sessions/saml/sp_init/auth0
   ```

   c. In **Allowed Callback URLs** add each address for each Lighthouse that you want to allow users to sign-in via (that is, IP, hostname, dns for both primary and secondary).

   ```
   https://{primary lighthouse address}/api/v3.7/sessions/saml/sso/auth0
   https://{primary lighthouse IP address}/api/v3.7/sessions/saml/sso/auth0
   https://{secondary lighthouse address}/api/v3.7/sessions/saml/sso/auth0
   https://{secondary lighthouse IP address}/api/v3.7/sessions/saml/sso/auth0
   ```

   d. Click **Save**.

5. Go to the **Addons** tab:
   a. Click **SAML2**.
   b. Go to the **SAML settings** tab.
   c. Set the Settings json to:

```
{
"audience": "lighthouse-auth0",
"mappings": {
"roles": "LH_Groups"
},
"passthroughClaimsWithNoMapping": true,
"mapUnknownClaimsAsIs": true,
"nameIdentifierFormat": "urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress",
"nameIdentifierProbes": [
"http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress"
]
}
```

   d. Click either **Enable** or **Save**.
6. Go to **Auth Pipeline**.
7. Go to **Rules**.
8. Click **Create**.
   a. Select empty rule template.
   b. Name it appropriately, for example, Map roles to SAML user property.
   c. Set the script to:

```
function mapSamlAttributes(user, context, callback) {
user.roles = (context.authorization || {}).roles;
callback(null, user, context);
}
```

9. Click **Save**.

CONFIGURE AUTH0 METADATA FOR IDP

To download IdP metadata:

1. Go to **Auth0**.
2. Go to **Application > Applications**.
3. Open your Lighthouse application.
4. Go to **Addons**.
5. Go to **SAML**.
6. Go to **Usage**.
7. Click the **Identity Provider Metadata** download.

   A file named `metadata.xml` downloads to your preferred directory

Configure each Lighthouse to use Auth0 IdP with the metadata file:

1. Copy the downloaded metadata.XML file to your primary Lighthouse.
2. Use the `saml-idp-metadata` file on your primary Lighthouse, configure each of your Lighthouses to use your IdP with this command:

   ```
   saml-idp-metadata -p {root password} create -m /path/to/metadata.xml
   -P auth0 -n "My Auth0 display name" -l {LH id number}
   ```

CONFIGURE AUTH0 FOR IDP

After you have added Lighthouse as an **Enterprise application** to AUTH0, you must use the App roles feature to grant users permissions to use IdP.

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

1. If you do not already have your own Usergroups setup in Lighthouse:
   a. Login to Lighthouse as a local user (or any non-SAML user) for example, root.
   b. Create the User groups with the required Roles and permission.
2. Go to **Auth0**.
3. Go to **User Management**.
4. Go to **Roles**.
5. Click **Create Role**.
6. Enter the Role name that matches a Lighthouse group name.
7. Open your Role.
8. Go to the **Users** tab.
9. Click **Add Users**.
   a. Assign the role to the appropriate users.

   The assigned users are now able to login to Lighthouse with the permission levels granted by the Auth0 Role/Lighthouse group.

LIMITATIONS OF SAML CONFIGURATION

IDP METADATA CERTIFICATE EXPIRY

The Identity Provider (IdP) metadata XML file that you exported to configure Lighthouse contains a certificate that is used to authenticate that the 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 must 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.

MAKE CHANGES TO USER PERMISSIONS

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 want to quickly restrict a user's access, consider altering the permissions of or deleting that user's user groups on Lighthouse. You can also set a low Web Session Timeout.

SAML SSO USER GROUPS

The **LH_Groups** attribute can be used to place SSO 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).

SAML SSO USERS

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

SAML SSO SUPPORT

Feature	Supported
Web Terminal to access the serial port through Lighthouse.	✓

Automation Gateway features through **NetOps > Automation Gateway**	✓
Web Terminal to login to the Lighthouse CLI via the Lighthouse Web UI.	✗
Connected Resource Gateway connect to a resource via SSH.	✗
SSH link on a port accessed through **Ports** from the Lighthouse web interface. For SAML users, the SSH link will not be visible.	✗

ADVANCED FUNCTIONALITY

NetOps modules leverage Opengear's remote presence and proximity to infrastructure by allowing software applications to run directly on top of the out-of-band management fabric. The NeSountOps modules enable automation of the configuration and operation of network infrastructure in data center and remote edge locations.

WHAT IS A NETOPS MODULE

NetOps Modules are software components that enable automation of specific operational scenarios (for example, network device provisioning), deployed as Docker containers. The modules become active on Lighthouse when a module license is installed and can then be manually or automatically activated on nodes.

LICENSING

NetOps modules require a Lighthouse with appropriate licensing and modules to be uploaded and active. You will also require at least one node enrolled and able to use these modules.

Lighthouse Edition	Secure Provisioning*	IP Access*	Automation Gateway
Lighthouse Enterprise		✓	
Lighthouse Enterprise: Automation Edition	✓	✓	✓

* The Smart Management Fabric feature can be used with these NetOps modules with some caveats.

MANAGE NETOPS MODULES

The management of NetOps modules is accessed from the 🖧 NetOps menu.

NETOPS PLATFORM SECURITY

The NetOps Automation platform uses a combination of advanced hardware and software security features to provide end-to-end security and resilience against network breach, fault or failure.

All communications between Lighthouse and nodes are tunneled inside Lighthouse VPN, using strong ciphers and automated certificate authentication and revocation.

Nodes such as the OM and CM8100 contain a TPM chip, which verifies the authenticity of the OM system software, its configuration, and NetOps Module code and data – any unauthorized tampering will render the appliance inoperable. These nodes are physically hardened to resist tampering – attempts to physically to access storage media will render the appliance inoperable

With a built-in cellular capability nodes with the TPM chip provide a secure WAN uplink in the event of network outage, DOS, or during initial network turn-up at a remote location.

CHANGE 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 must alter two 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.

NETOPS MODULE INSTALLATION

PERFORM NETOPS INSTALLATION FROM AN ONLINE REPOSITORY

1. Select **NetOps > NetOps Installation**.

   The **NETOPS INSTALLATION** page displays defaulted to the **ONLINE** tab.
2. Enter the *Repository address* the container registry that hosts NetOps modules.
3. Click ⇄ **Start Online Sync** to start the process.

   A progress message displays:

   ```
   NetOps online sync in progress
   ```

   The sync completes and the following message displays:

   ```
   NetOps online sync completed
   ```

PERFORM NETOPS INSTALLATION FROM AN OFFLINE REPOSITORY

If Lighthouse is deployed on a network where outbound access to the Docker Hub repository is not permitted or not available, use the NetOps offline installer. The offline installer file can be downloaded from this link: offline installer file.

1. Select **NetOps > NetOps Installation**.

   The **NETOPS INSTALLATION** page displays.
2. Select the **OFFLINE** tab.

3. Drag and drop or select a file directly to upload.

   > **Note:** Supported files: `.tar.gz`

4. Click **Upload** to start the process.

PERFORM NETOPS INSTALLATION FROM AN OFFLINE REPOSITORY VIA CLI

1. Copy the offline installer to Lighthouse using scp, WinScp or similar, into the **/mnt/nvram** directory.
2. Log in to Lighthouse shell CLI as a Lighthouse Administrator and run:

   ```
   gzip -d </mnt/nvram/netops_modules_*.tar.gz | nom update && rm
   /mnt/nvram/netops_modules_*.tar.gz
   ```

3. Deploy the upgrade to nodes:
   a. Log in to the Lighthouse web UI as a Lighthouse Administrator or the root user.
   b. Select **NetOps > Manage NetOps Modules**.

      The **MANAGE NETOPS MODULES** page displays.
   c. Confirm the module version.
   d. Click **Redeploy**.

      A progress message displays:

      ```
      Module activation is in progress
      Module activation may take several minutes
      ```

      The deployment completes and the following message displays:

      ```
      Success
      Module synchronization complete
      ```

INSTALL MODULES FROM THE CLI

1. Log in to the Lighthouse CLI as root.

   > **Note:** Optionally, log in as a Lighthouse Administrator and become root with: `sudo -i`

2. Run the following procedure replacing *root* and *default* with a Lighthouse Administrator or root credentials:

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

ACTIVATE A NETOPS MODULE

The NetOps module license is installed on Lighthouse and contains a preset number of available node activations. Each supported node that is activated for the NetOps module provisioning, consumes an available activation. Lighthouse itself does not consume an activation.

Installing the NetOps module license automatically activates the module on Lighthouse, at which point the NetOps Automation platform deploys the central module software components, including the module's UI, to Lighthouse.

The process of automatically or manually activating a NetOps Module on a node deploys the remote module software components to that node, securely over Lighthouse VPN.

PREPARATION

1. Install the Lighthouse VM and ensure it is running.
2. Login to the Lighthouse web UI as root or a Lighthouse Administrator.
3. Install the Lighthouse node subscription (SKU OGLH) under ⚙ **> SYSTEM > Subscriptions**.

PROCEDURE

1. Ensure you have a valid NetOps Module licence installed under ⚙ **> SYSTEM > Subscriptions**.

   > **Note:** The Enterprise Edition license includes the IP Access module.
   > The Automation Edition license includes IP Access, Secure Provisioning, and Automation Gateway.

2. It will take a few minutes for the module to activate on Lighthouse. View the progress under **NetOps > Manage NetOps Modules**.

Nodes may now be automatically activated for the module as they enroll, or manually activated after enrollment.

'ALWAYS ACTIVATE' MODE ON ALL NODES (AUTOMATIC)

In the Always Activate mode, NetOps automatically activates the NetOps Module on all nodes, provided a license is present and activations are available. All nodes are activated for the module as they enroll.

PROCEDURE

1. Select 🖧 **NetOps > Manage NetOps Modules**.

   The **MANAGE NETOPS MODULES** page displays.

2. Under **Always Activate**, select **ENABLED** as required for the module:
   - SECURE PROVISIONING
   - IP ACCESS
   - AUTOMATION GATEWAY
3. Click **Apply**.

To activate a new node, enroll the node into Lighthouse.

ACTIVATE NETOPS ON SELECTED NODES (AUTOMATIC)

You may selectively activate the module on a subset of nodes using Enrollment Bundles. Only nodes enroling using one of these bundles will be automatically activated.

PROCEDURE

4. Select **NetOps > Manage NetOps Modules**.

   The **MANAGE NETOPS MODULES** page displays.
5. Under **Always Activate**, select **DISABLED** as required for the module:
   - SECURE PROVISIONING
   - IP ACCESS
   - AUTOMATION GATEWAY
6. Click **Apply**.
7. Select **Node Tools > Enrollment Bundles**.

   The **ENROLLMENT BUNDLES** dashboard displays.
8. Click **Add Enrollment Bundle** to add a new bundle (you may also edit an existing bundle). The **NEW ENROLLMENT BUNDLE** page displays.
9. Enter the details and choose whether or not to **Auto-Approve** enrollment.

10. Under the **NETOPS SECTION** select **Add Module**.

    The **CHOOSE MODULE** dialog is displayed.

11. Select the module from the drop-down.

12. Click **Apply**.

> **Note:** Lighthouse-initiated manual enrollment (for example, clicking the **Add Node** button in the Lighthouse web UI) does not support bundles, you must use a node-initiated enrollment method.

ACTIVATE THE NETOPS MODULE ON NODES (MANUAL)

To activate nodes manually after enrollment, the process is performed across three phases:

- Select Nodes
- Pre-Flight Test
- Push

SELECT NODES

1. Ensure the **Always Activate** option is **DISABLED** and applied for each module on the **NetOps > Manage NetOps Modules** dashboard.

2. Select **Node Tools > Config Templates**.

   The **CONFIG TEMPLATES** dashboard is displayed defaulted to the **AUTHENTICATION** tab.

3. Select the **NETOPS MODULES** tab.

4. Select the **Push Template** from the actions available on the module.

   The **PUSHING TEMPLATE** page displays.

5. Check the **NODE ID** to push the template to.

PRE-FLIGHT TEST

6. Click **Run Pre-Flight Test**.

   The page displays the current **Pre-Flight Status** per node.

PUSH

7. On successful completion of the *Pre-Flight Test*, select **Push Configuration**.

   The page displays the current **Push Status** per node.

> **Note:** Under the NetOps *licensing* arrangement, when a node is activated for the NetOps Module, the activation is consumed by and locked to that node. Unenrolling the node returns the activation to the available pool. Under NetOps *subscription* arrangements, it would be counted towards the node assignment of the subscription.

DEACTIVATE (REMOVE) A NETOPS MODULE

DEACTIVATE AND REMOVE A NETOPS MODULE VIA THE API

Use the following API call to Lighthouse:

```
DELETE /api/v3/netops/modules/module_id/nodes/node_id
```

where:

- `module_id` is one of dop (Secure Provisioning), IP Access, or ag (Automation Gateway).
- `node_id` is the internal node ID, for example, nodes-1.

DEACTIVATE AND REMOVE NETOPS MODULES VIA CLI

1. Log in to the Lighthouse CLI shell as root or a Lighthouse Administrator user.
2. Determine the node's node ID by running:

   ```
   node-info --list
   ```

3. Update the highlighted fields with your username, password, the node ID and modules to deactivate, then run the following:

   ```
   LH_USERNAME="root"
   LH_PASSWORD="default"
   NODE_ID="nodes-1"
   MODULES_TO_DEACTIVATE="dop ag sdi"

   token=$(curl -k -L -d '{"username":"'$LH_USERNAME'","password":"'$LH_
   PASSWORD'"}' https://127.0.0.1/api/v1/sessions/ | cut -f10 -d'"')

   for module_id in $MODULES_TO_DEACTIVATE; do
    curl -k -L -X DELETE -H "Authorization: Token $token"
   https://127.0.0.1/api/v3/netops/modules/${module_id}/nodes/$NODE_ID
   done
   ```

SECURE PROVISIONING

Secure Provisioning for NetOps is a configuration storage, distribution and provisioning system. It does not generate, test or validate device configuration. Instead, it is focused on provisioning remote resources with user-supplied configuration and device OS images, automatically, remotely and securely, no matter where those devices are and no matter what the state of the network is.

The Secure Provisioning license is installed on Lighthouse and contains a preset number of available node activations. Each node activated for Secure Provisioning consumes an available activation; Lighthouse itself does not consume an activation.

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

The Secure Provisioning module leverages these technologies:

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

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

SECURE PROVISIONING CONFIGURATION MANAGEMENT

Secure Provisioning always applies device configuration in its entirety and does not support applying config patches or deltas to a provisioned device (for example, adding a few lines to running config, to enable a specific feature).

STATELESS FILE MANAGEMENT

Secure Provisioning supports a DevOps-style approach which collapses initial provisioning, disaster recovery and ongoing maintenance workflows into the one workflow:

Using this approach, the config patch is applied in Lighthouse to the central configuration template via git, which renders the configuration file in its entirety and pushes to the OM node. The device is factory reset and pulls the new configuration as if it were being provisioned for the first time.

Pros	Cons
Eliminates config drift.	Requires a longer maintenance window as the device is reset and reboots.
Enforces config reproducibility.	Patches cannot be applied to running configuration.
Central audit trail of all configuration changes.	
Disaster recovery becomes as simple as resetting all devices to reprovision.	

STATEFUL DEVICE MANAGEMENT GATEWAY

The NetOps Automation platform provides a management fabric from remote devices to your central management network via Lighthouse VPN and/or the cellular WWAN.

There are many tools and protocols purpose-built for stateful configuration management, such as Cisco NSO and SolarWinds NCM, and NETCONF and gRPC (OpenConfig).

NetOps can be leveraged by these tools as a secure, resilient management path, both extending their reach to the out-of-band management network, and ensuring reachability during outages.

HOW SECURE PROVISIONING WORKS

The Secure Provisioning feature centrally orchestrates the distribution of resource configuration files and firmware images, and the node provisioning (ZTP) services required to deliver the files to resources.

Secure Provisioning is configured by defining the resources to provision resources with, and defining how these resources should be distributed around your network.

- **Device Resource Bundles** contain the files required to provision one or many resources:
  - *Configuration File*, *Script File* and/or *Image Files*.
  - Each Resource Bundle has a defined *Device Type*.
  - When a Resource Bundle is distributed to a node, any ZTP request matching the Device Type are provisioned with the bundled resources.
  - This may be restricted to specific devices by specifying one or more device *MAC Addresses* (range and reverse match supported) or *Serial Numbers* (not supported by all vendors).
- **Resource Distribution** policies are defined by **Node Inventory Lists**:
  - *Static Node Inventory List* - a predefined, static list of nodes to distribute to.
  - *Dynamic Node Inventory List* - evaluates a Node Filters each time resources are distributed.

> **Tip:** The Dynamic Node Inventory List allows you automatically tag certain nodes with Enrollment Bundles, for example, by region or site class, to help automate resource distribution to newly enrolled nodes in that region.

Device Resource Bundle and Resource Distribution configuration are supplied to Lighthouse using the web UI or CLI (git) method. The Web UI configuration method creates an underlying YAML configuration the same as created using the git method, it is effectively a front end to the git method.

A *git push* to the Lighthouse repository, or clicking the UI *Push Now/Push Resources* button triggers a resource push:

- A git post-commit hook triggers an Ansible playbook on Lighthouse.
- The playbook copies resources down to nodes, securely over Lighthouse VPN.
- The playbook start or restarts ZTP services on nodes.

SUPPORT FOR SECURE PROVISIONING

Opengear OM2200 and OM1200 nodes may be activated as Secure Provisioning nodes.

Opengear ACM7000 or IM7200 nodes may also be activated as provisioning nodes, however not all features are available and there are some caveats to be aware of.

Features that are not available of ACM7000/IM7200 nodes:

- Secure boot and physical tamper resistance.
- Encryption of device resource files at rest.
- Centralized ZTP status logging.
- Device configuration templating.
- Ordered provisioning.
- Post-provisioning scripts.

Other ACM7000/IM7200 caveats:

- Secure Provisioning takes control of node DHCP, NTP, DNS services and overwrites system configuration.
- Secure Provisioning overwrites node Management LAN configuration.

VENDOR RESOURCES SUPPORTED BY SECURE PROVISIONING

Secure Provisioning is vendor-neutral, with support for a broad range of network devices from multiple vendors.

The ZTP process used to provision devices is not standardized, and each vendor OS implements ZTP differently – for example, using differing DHCP options, or requiring an intermediary script to load files.

With Secure Provisioning, you upload configuration and/or firmware image files to create Resource Bundles, then select the vendor profile for that Resource Bundle. This automatically generates the vendor-appropriate ZTP configuration, simplifying the delivery of resources to target devices.

Secure Provisioning currently has built-in support for provisioning devices from these vendors:

- Cisco (IOS, IOS XR, IOS XE, NX-OS)
- Juniper
- Arista
- HPE/Aruba
- Huawei
- Cumulus
- Pica8
- Opengear

Advanced users may add support for additional devices using custom DHCP configuration.

CONNECT TO A TARGET DEVICE

1. Connect a supported resource's management NIC directly to the node.
2. If the node has a built-in Ethernet switch, connect the device to any switch port.

3. Otherwise, connect the device directly to the node's NET2 Ethernet, or via an intermediary management switch.
4. Power on the resource.
5. Ensure the resource is in ZTP mode, this typically requires the device to have its configuration erased/reset to factory defaults.

LOCAL NETWORK SERVICES PROVIDED BY NODES

In addition to zero touch provisioning (ZTP) services, the local node runs local services required to act as a bootstrap management LAN and secure WAN for resources from day zero onwards.

When responding to a BOOTP/DHCP provisioning request from a device, the Operations Manager node hands out its own local address as a:

- Default Gateway
- DNS Server
- NTP Server
- SYSLOG Server

DEFAULT GATEWAY

Devices trying to reach to destinations on the central LAN that Lighthouse resides on are securely routed over Lighthouse VPN. This allows devices to reach, for example, central NMS for monitoring, and central configuration systems for final service provisioning.

Requests to other remote destinations are masqueraded behind and routed out the node's built-in cellular WWAN, allowing devices to reach cloud provisioning services.

Note that device requests are masqueraded to Lighthouse's central IP and will appear to be originating from Lighthouse to hosts on the central LAN.

All traffic between remote node network and the central Lighthouse network is securely tunneled inside Lighthouse VPN.

DNS SERVER

DNS lookups from devices are securely proxied through Lighthouse VPN to the central DNS server(s) used by Lighthouse, allowing devices to resolve central hosts from day one.

NTP SERVER

The NTP Server allows devices to set accurate time on first boot, for example, for certificate verification and generation. By default, the node's NTP service uses its local hardware clock as time source.

SYSLOG SERVER

The Syslog Server relays messages to a central LogZilla instance (this is an optional extra module). This allows log collection from day zero, and analysis of the device ZTP process itself.

SECURE PROVISIONING CONFIGURATION

All system configuration is performed via Lighthouse. The configuration necessary to provision a device consists two elements.

The basic steps to configure Secure Provisioning are:

- Create Device Resource Bundles and upload resource files (for example, configuration files or scripts, firmware images) to Lighthouse.
- Define Node Inventories to distribute the resources to specific nodes, where they will become available for devices to request for provisioning.

DEVICE RESOURCE BUNDLE

A Device Resource Bundle contains the resource files, such as, a configuration file and OS upgrade image that are loaded via ZTP (DHCP + TFTP/HTTP) onto the resource. This may be a full, final configuration, or a baseline configuration to allow the resource to become managed by an upstream configuration service.

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

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

NODE INVENTORY

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

Resource Bundles may be distributed using one of two methods:

- Push to a static list of nodes, selected individually by node ID
- Push to a dynamic list of nodes, linked to a Lighthouse Node Filters of nodes

Note: You may combine distribution methods.

CREATE A DEVICE CONFIGURATION

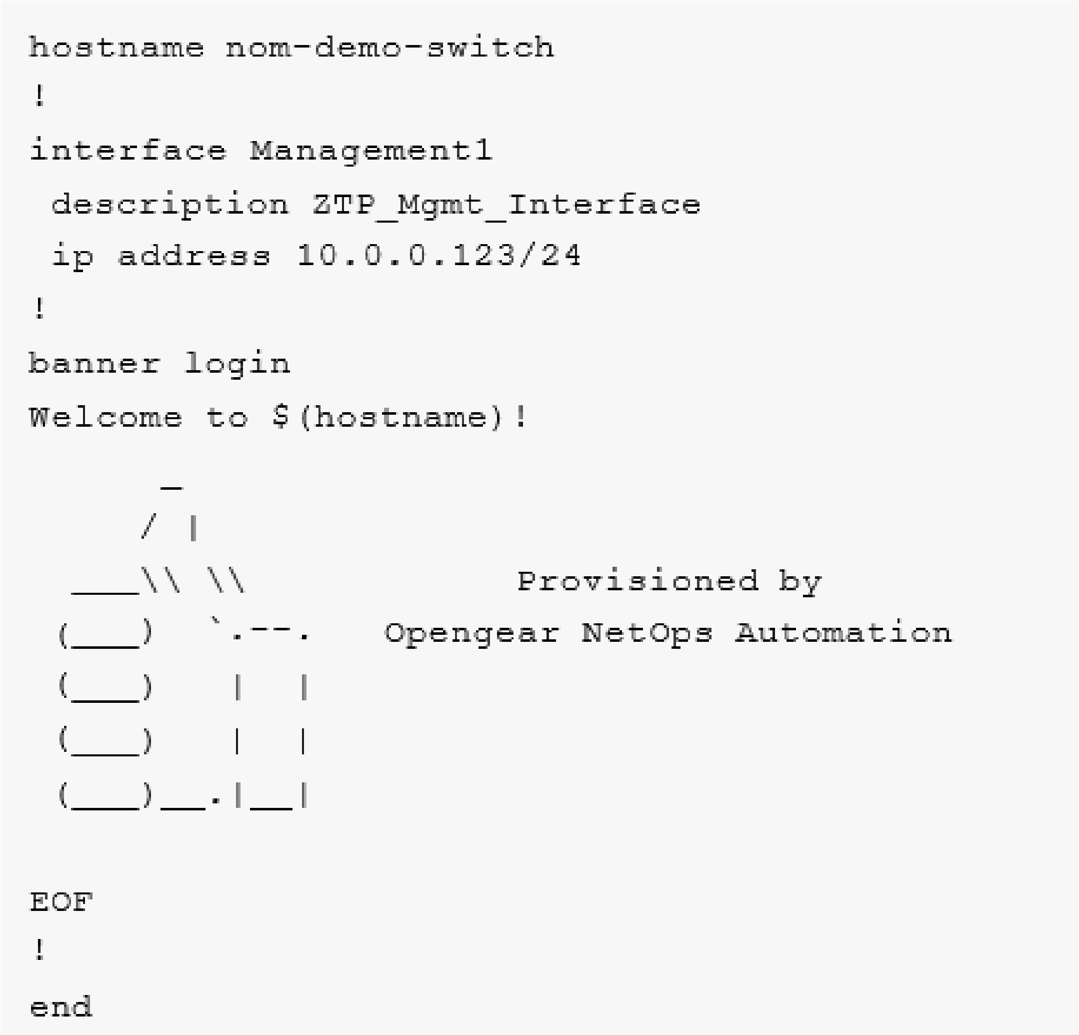
To provision a resource, you must supply device resources. Device resources consist of an initial configuration file for the device to install, and optionally an operating system image for the device to upgrade itself with.

Device resource file formats are specific to the target vendor. Secure Provisioning for NetOps Automation provisions these files, but does not generate them.

EXAMPLE CONFIGURATION FILES

- A trivial Arista initial configuration file may look like:

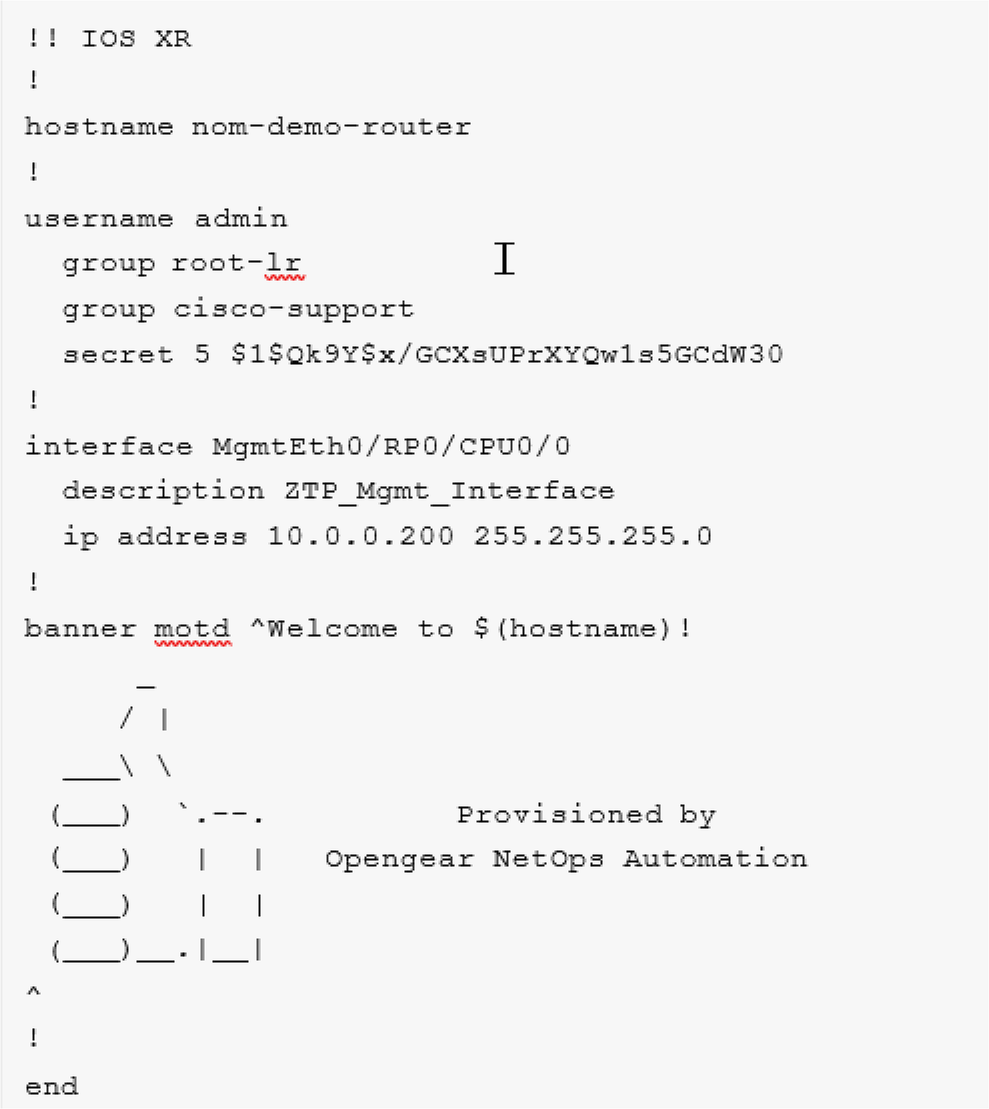
  `demo_arista.cfg`



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

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

EOF
!
end
```

- Example 2: trivial Cisco IOS XR initial configuration may look like:

  `Cisco IOS XR initial configuration`



```
!! IOS XR
!
hostname nom-demo-router
!
username admin
  group root-lr
  group cisco-support
  secret 5 $1$Qk9Y$x/GCXsUPrXYQw1s5GCdW30
!
interface MgmtEth0/RP0/CPU0/0
  description ZTP_Mgmt_Interface
  ip address 10.0.0.200 255.255.255.0
!
banner motd ^Welcome to $(hostname)!
      _
     / |
  ___\ \
 (___)  `.--.          Provisioned by
 (___)   |  |   Opengear NetOps Automation
 (___)   |  |
 (___)__.|__|
^
!
end
```

CLI BASED WORKFLOW

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

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

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

Note that changes pushed to nodes via the Lighthouse UI or API will override those made by direct repository access, therefore UI- based or CLI-based workflows should be considered mutually exclusive modes of operation.

CREATE CONFIGURATION YAML

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

> **Tip:** The web UI provides a convenient way to start provisioning resources without requiring to be fully familiar with YAML or git. The generated YAML files that controls resource bundling and distributions is located on Lighthouse, inside the central-dop container. You can view it by running the following command:
>
> ```
> sudo docker exec -it central-dop cat /srv/central-ui/root/config.yml
> | less
> ```

Use a YAML file to bundle device resources, and control the distribution of device resources from Lighthouse to the nodes.

Procedure

1. Create a new directory or folder of your choosing, for example: *nom-prov*.
2. Inside the *nom-prov* directory, create a new directory or folder called: *downloads*.

3. In the *nom-prov* directory, create a file with the .yml or.yaml extension, using the following format:

NOM-PROV.YML

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

node_inventory:
 MyNodes:
  static:
  - nodes-1

deployment:
 MyNodes:
 - demo_arista
```

Note: Indentation is meaningful in YAML, and you must use space characters not tabs to indent.

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

4. Choose an identifier for each resource bundle item, for example *demo_arista*.
5. For each item, you must provide the **device_type**, as well as well as one or more resources, i.e. *config_file* or *image_file.*

- **device_type** matches this device resource item to all devices from the specified vendor – it may be one of the following: Cisco, Cisco_xe, Cisco_xr, Cisco_nx, Juniper, Arista, Aruba, Huawei, Cumulus, Pica, Opengear.
- **config_file** is the initial configuration file for the device to load via ZTP, as present in the *downloads* directory.
- **image_file** is the initial software image for the device to load via ZTP, as present in the downloads directory.

> **Note:** HPE/Aruba devices do not support the image upgrade via ZTP.
>
> The Cisco Autoinstall process does not support image upgrade via ZTP, to automate image upgrade you must supply a TCL script file rather than a configuration file.

- **mac_address** optionally target this bundle at the listed MAC address(es), which may be specified in full, using a wildcard (for example, 00:10:FA:C2:BF:*), or negated to exclude from the match (for example, !01:23:45:67:89:AB).
- **serial_number** optionally target this bundle at the listed serial number(s).

Device resource items are then assigned to nodes using the **deployment** and optionally the **node_inventory** lists. See *Node Inventory* for an overview of available distribution methods.

The **node_inventory list** defines groups of nodes.

Choose an identifier for each inventory, for example:. *branchinventory_* or *labinventory_* .

DEFINE A STATIC INVENTORY:

1. Create a list named **static**.
2. List nodes by node ID, for example, *nodes-1*.
3. You can view node IDs by running the following command on Lighthouse: `node-info --all`

DEFINE A DYNAMIC INVENTORY:

1. Create a Lighthouse Node Filters.
2. Create a key named **smartgroup** with a value of the Node Filters name, this Node Filters search is dynamically evaluated to a list of nodes each time resources are pushed

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

- Deployment identifiers correspond to **node_inventory** identifiers, for example, *branchinventory_*.
- Assign device resources by listing device resource items, for example, *demo_arista*.
- You may have multiple device resources per deployment.

A more comprehensive YAML file may look like:

`more-devices.yml`

```
device_resources:
 access_switch:
  device_type: juniper
  config_file: 'jn-switch35.config'
  image_file: 'jinstall-ex-4200-13.2R1.1-domestic-signed.tgz'
  mac_address:
  - '00:00:0c:15:c0:*'
  - '!00:00:0c:15:c0:99'
 branch_router:
  serial_number:
  - 'SAD15300D4W'
  - 'FOC1749N1BD'
  - 'AVJ18163A52'
  config_file: 'branch_xr.cfg'
  device_type: cisco_xr
 demo_arista:
  device_type: arista
  config_file: 'demo_arista.cfg'
  image_file: 'arista_eos.swi'

node_inventory:
 branch_inventory:
  static:
  - nodes-1
  - nodes-2
  - nodes-10
 lab_inventory:
  smartgroup: LabNodes

deployment:
 lab_inventory:
 - demo_arista
 - access_switch
 branch_inventory:
 - branch_router
 - access_switch
```

HOW UI FIELDS CORRESPOND TO THE YAML FILE (EXAMPLE)

The following example YAML file contains line-by-line comments (blue text) denoting the UI page or field above each corresponding YAML element:

```
# CONFIGURE NODES > Secure Provisioning > Device Resources
device_resources:
 # Device Resource Details > Name
```

```
access_switch:
 # Device Resource Details > Device Type
 device_type: juniper
 # Device Resource Details > Configuration File
 config_file: 'jn-switch35.config'
 # Device Resource Details > Image File
 image_file: 'jinstall-ex-4200-13.2R1.1-domestic-signed.tgz'
 # Device Resource Details > MAC Addresses
 mac_address:
 - '00:00:0c:15:c0:*'
 - '!00:00:0c:15:c0:99'
# Device Resource Details > Provision After
 provision_after:
 - branch_router

# Device Resource Details > Name
 branch_router:
 # Device Resource Details > Device Type
  device_type: cisco_xr
  # Device Resource Details > Serial Numbers
  serial_number:
  - 'SAD15300D4W'
  - 'FOC1749N1BD'
  - 'AVJ18163A52'
 # Device Resource Details > Configuration File
  config_file: 'branch_xr.cfg'

# Device Resource Details > Name
 demo_arista:
  # Device Resource Details > Device Type
```

```
   device_type: arista
   # Device Resource Details > Configuration File
   config_file: 'demo_arista.cfg.j2'
   # Device Resource Details > Image File
   image_file: 'arista_eos.swi'
   # Device Resource Details > Post-Provisioning Script
   post_provision_script: arista_fixups_over_ssh.py
   post_provision_script_timeout: 900
# CONFIGURE NODES > Secure Provisioning > Resource Distribution
node_inventory:
 # Static Node Inventory List > Inventory Details > Name
 BranchInventory:
  # Inventory Details > Select Nodes
  static:
  - nodes-1
  - nodes-2
  - nodes-10

 # Dynamic Node Inventory List > Inventory Details > Name
 LabInventory:
  # Inventory Details > Node Filters
  smartgroup: LabNodes

# CONFIGURE NODES > Secure Provisioning > Resource Distribution (mostly!)
deployment:

  # CONFIGURE NODES > Secure Provisioning > Resource Distribution        Inventory
Details > Resource Push
  LabInventory:
  - demo_arista
  - branch_router
  - access_switch
  # CONFIGURE NODES > Secure Provisioning > Resource Distribution > Inventory Details > Resource Distribution
  BranchInventory:
  - branch_router
  - access_switch
```

UPLOAD CONFIGURATION AND RESOURCES

1. Assemble device resources on your PC or laptop in preparation for upload.
2. Locate the *nom-prov* directory created in the previous section.
3. Copy device resources into *nom-prov/downloads*.

Your locally assembled files will now look similar to that below:

```
.
└── nom-prov
    ├── nom-prov.yml
    └── downloads
        ├── arista_eos.swi
        └── demo_arista.cfg
```

You must now choose how you will upload files to the central Secure Provisioning repository, using Secure Copy (`scp`) or `git`.

OPTION A. SECURE COPY METHOD

Secure copy the entire *nom-prov* directory to Lighthouse port 2222, to the **/srv/central-auto/** directory and authenticating as root, for example, using the `scp` command:

```
cd nom-prov

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

.. where 192.168.0.1 is the IP address of Lighthouse.

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

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

OPTION B. GIT METHOD

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

Example commands to initialize the repository for the first time:

```
ssh-copy-id root@192.168.0.1

cd nom-prov

git init

git remote add origin ssh://root@192.168.0.1:2222/srv/central

git add -A

git commit -a -m "Initial commit of ZTP resources"

git push origin master
```

where 192.168.0.1 is the IP address of Lighthouse.

After the repository has been initialized, subsequent users can operate on it using the `clone` command:

```
ssh-copy-id root@192.168.0.1

git clone ssh://root@192.168.0.1:2222/srv/central nom-prov

cd nom-prov
```

```
echo >> nom-prov.yml

git commit -a -m "Whitespace change for testing, please ignore"

git push origin master
```

where 192.168.0.1 is the IP address of Lighthouse.

ADDITIONAL RESOURCE FILES AND DEVICE TYPE FILES

You may also provide additional resource files that are not explicitly part of Device Resource Bundles, for example, final configuration files that may be conditionally fetched and applied by the device's primary ZTP script.

You may also extend Secure Provisioning to support additional device types by providing ISC DHCP configuration snippets, for example:

new-vendor.conf

```
class "new-vendor-class" {

 match if (option vendor-class-identifier = "new-vendor";

 option bootfile-name "new-vendor.cfg";

}
```

Additional files must be placed in the subdirectory named after the Node Inventory they will be deployed to. Within this subdirectory, files must be placed in the following:

- Resource files such as device configuration or image files are placed in the **downloads** directory.
- Advanced: DHCP snippets may be placed in the **dhcpd** directory.

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

```
.
└── nom-prov
├── nom-prov.yml
├── downloads
│   ├── demo_arista.cfg
│   ├── cumulus_interfaces
│   ├── cumulus_setup.sh
│   └── arista_eos.swi
└── my_inventory
    ├── downloads
    │   └── new-vendor.cfg
    └── dhcpd
        └── new-vendor.conf
```

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

CONFIGURE DEVICE RESOURCES VIA ZTP

There are two factors that determine which resources are delivered to which devices via ZTP:

- Device resource bundle matching.
- Resource distribution.

DEVICE RESOURCE BUNDLE MATCHING

As well as containing resource files themselves, each Resource Bundle itself has a few extra parameters: device vendor, device MAC address(es) and device serial number(s) (not supported by all vendors). Of these, only the device vendor is mandatory.

When a resource broadcasts a BOOTP/DHCP request to initiate ZTP, it advertises its vendor ID string, MAC address, and in some cases serial number. These values are compared to the values in each Resource Bundle contained on the local node.

If there's a match, the local node provisions the device with the resource files in the matching bundle.

RESOURCE DISTRIBUTION

Node Inventories are used to selectively control which Resource Bundles are pushed to which nodes.

A node will only respond to a BOOTP/DHCP request on its local network if a matched Resource Bundle has been pushed to it.

Note that resources are not distributed any nodes by default.

BASELINE VS FINAL DEVICE CONFIGURATION

Broadly speaking, there are two approaches to secure provisioning using ZTP.

You may use strict matching and distribution settings to provision specific devices with unique, final configurations.

Alternatively, you may use laxer matching and wider distribution settings to provision many devices with a baseline configuration, for example, "just enough configuration" to route to a central production configuration system for final configuration and service provisioning.

You may also combine the two approaches, for example, use a reverse MAC address match to opt a specific device or devices out of an otherwise general, baseline configuration.

RUN A SCRIPT ON A NEWLY PROVISIONED DEVICE

> **Note:** Post-provisioning scripting is an advanced feature only supported by Operations Manager nodes.

It is possible to upload a script and associated with a device Resource Bundle, to be run by remote Operations Manager node after a resource is considered provisioned. A device is considered to be in a provisioned state after it has downloaded all of the files in the Resource Bundle it is being provisioned with.

The script may be uploaded and associated via the UI during Resource Bundle creation using the **Post-Provisioning Script** option, or via **git/scp** and the CLI based workflow.

Scripts may be implemented in bash, Python 2 or Python 3, and must start with a shebang – i.e. the first line must be one of:

```
#!/bin/bash
```

```
#!/usr/bin/env python2
```

```
#!/usr/bin/env python3
```

- Scripts are run in a monitored background processes in the Secure Provisioning container (remote-dop) on the node.
- Scripts have a default 15 minute timeout, this can be manually configured in the YAML config (post_provision_script_timeout).
- Scripts may login to target device via the network using SSH key auth where `nom_remote_ssh_pub_key` has been injected into the device config, or using username/password with the `sshpass` command.

MONITOR THE ZTP PROGRESS OF A RESOURCE

The current provisioning state of resources can be monitored via syslog on the Operations Manager (local devices only) or Lighthouse (all devices).

Each Secure Provisioning `syslog` message contains a prefix identifying the MAC address of the device being provisioned, similar to:

```
 [NetOps-DOP device="01:23:45:67:89:AB"]
```

For example, here are sample messages showing an Opengear ACM7004 device being provisioned:

```
[NetOps-DOP device="00:13:C6:EF:00:08"] Received DHCP request from device with vendor ID Opengear/ACM7004-5-LMR
[NetOps-DOP device="00:13:C6:EF:00:08"] Assigned DHCP address of 10.0.0.2 to device
[NetOps-DOP device="00:13:C6:EF:00:08"] Provisioning device with resource bundle my\_acm7004
[NetOps-DOP device="00:13:C6:EF:00:08"] Device retrieved resource file /files/acm7004-5-4.3.1.flash via HTTP/HTTPS
```

Syslog can be viewed from the CLI by running:

```
tail -F /var/log/messages | grep NetOps-DOP
```

WAN GATEWAY SERVICES

In addition to LAN provisioning, the node can utilize its built-in cellular connection to act as a WAN gateway and provide a proxy to essential services for devices on day one.

The node's DHCP server hands out the node's address as:

Server	Description
NTP server	This is a local service, synced to the node's system clock.
DNS server	DNS lookups by devices are relayed via Lighthouse VPN, to the DNS server that Lighthouse is configured to use.
Syslog server	Note that incoming syslog messages are dropped unless LogZilla for NetOps Automation has been activated.
Default gateway	When the node is configured in cellular router mode (i.e. with forwarding and masquerading enabled), devices can route to external services, for example, to enroll with third-party management systems for additional configuration.

ADVANCED OPTIONS

USE VARIABLES IN CONFIGURATION FILE TEMPLATES

In addition to static files, you may create templated ZTP configuration or script files. This is useful if your file is required to reference site-specific values such as an assigned IP addresses.

Any file uploaded via the web UI, or into the downloads directly with a file suffix of `.j2` (Jinja2) will be automatically templated. The `.j2` suffix is stripped when serving templated files to devices.

Advanced Variable	Description	Example
{{ nom_remote_server }}	Address of provisioning interface on the node.	10.0.0.1
{{ nom_remote_interface }}	Name of provisioning interface on the node .	net2

Advanced Variable	Description	Example
{{ nom_remote_netmask }}	Netmask of provisioning interface on the node (and netmask assigned in DHCP offers).	255.255.255.0
{{ nomremotenetmaskcidr }}	CIDR format netmask (prefix length) of provisioning interface on the node	24
{{ nom_remote_ntp_server }}	Address of NTP server assigned in DHCP offers (same as nom_ remote_server)	10.0.0.1
{{ nom_remote_dns_server }}	Address of DNS server assigned in DHCP offers (same as nom_ remote_server)	10.0.0.1
{{ nom_device_ipv4_address }}	This feature is only supported by Operations Manager nodes. DHCP address assigned to target device.	10.0.0.13
{{ nomremotesshpubkey }}	Public part of an auto-generated SSH keypair on the remote node, which may be injected into device config to pre- authenticate the node for any post-provisioning activities.	
{ nom_device_hostname }}	This feature is only supported by Operations Manager nodes.Hostname advertised by target device.	router
{ nom_device_mac_address }}	MAC Address of target device.This feature is only supported by Operations Manager nodes.]	00:12:34:56:78:9A

For example, a basic Cumulus templated provisioning script may look like:

`cumulus_setup.sh.j2`

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

POST-PROVISIONING SCRIPTS

Note: This feature is only supported by Operations Manager nodes.

It is possible to upload a script and associated with a device Resource Bundle, to be run by remote Operations Manager node when a resource is considered provisioned. A device is considered to be in a provisioned state after it has downloaded all of the files in the Resource Bundle it is being provisioned with.

The script may be uploaded and associated via the UI during Resource Bundle creation using the **Post-Provisioning Script** option, or checked-in to the **downloads** directly via `git/scp` and specified in the YAML configuration for the device Resource Bundle:

```
demo_arista:
 device_type: arista
 config_file: 'demo_arista.cfg'
 image_file: 'arista_eos.swi'
 post_provision_script: arista_fixups_over_ssh.py
 post_provision_script_timeout: 900
```

Scripts may be implemented in bash, Python 2 or Python 3, and must start with a shebang – i.e. the first line must be one of:

```
#!/bin/bash

#!/usr/bin/env python2

#!/usr/bin/env python3
```

Notes:

- Scripts are run in a monitored background processes in the Secure Provisioning container `(remote-dop)` on the node.
- Scripts have a default 15 minute timeout, this can be manually configured in the YAML config (`postprovisionscripttimeout_`).
- Scripts may login to target device via the network using SSH key auth where `nomremotesshpubkey` has been injected into the device config (see Templated resource above), or using `username/password` with the `sshpass` command.

ORDERED PROVISIONING

> **Note:** This feature is only supported by Operations Manager nodes.

The **Provision After** option allows you to create basic dependency chains, to enforce the order in which devices are provisioned. In certain scenarios it may be advantageous to control the order in which devices are provisioned, for example:

- Ensure security infrastructure is provisioned ahead of systems that may other become inadvertently exposed on the network.
- Bring the production WAN up early to allow devices to provision services in-band, saving cellular data.
- Disallow local user access to the LAN until the network is fully up and running.

When a secondary Resource Bundle has the **Provision After** option set, the node will not respond to ZTP requests for these resources until all required dependencies have been met.

The Provision After property lists of one or more other, required Resource Bundles. Each required Resource Bundle creates a dependency that at least one device has been provisioned using the required bundle.

If multiples of a particular required device must be provisioned before a secondary device, simply specify the dependency multiple times in the list.

You may configure this via the UI during Resource Bundle creation using the Provision After option, or directly via git/scp in the YAML configuration for the device Resource Bundle:

```
access_switch:
 device_type: juniper
 config_file: 'jn-switch35.config'
 image_file: 'jinstall-ex-4200-13.2R1.1-domestic-signed.tgz'
 provision_after:
 - branch_router
```

TROUBLESHOOT SECURE PROVISIONING

SECURE PROVISIONING CONSISTS OF SEVERAL DOCKER CONTAINERS

Container	Description
`central-dop`	The container runs on Lighthouse, hosting `git` repository.
`dop-ui`	The container runs on Lighthouse, serving the Secure Provisioning web UI.
`remote-dop`	The container runs on Operations Manager nodes, running DHCP and TFTP/HTTP ZTP services.

Additionally, the deployment container runs on Lighthouse, orchestrates new module installation on Lighthouse and nodes.

TROUBLESHOOTING COMMANDS FOR NETOPS MODULES

Command	Description
`docker ps`	View running Docker containers.
`docker exec -ti container-name bash`	Spawn a bash shell inside a container.
`docker logs container-name`	View logs of a container.
`docker exec -ti deployment ansible-playbook -vvv /ansible/dop_2.0.0.yml`	Manually run module deployment in verbose mode, on Lighthouse.
`etc/scripts/post-receive`	Manually push ZTP resources from Lighthouse to Operations Manager nodes (inside central-dop container).
`/etc/scripts/netops_ui_handler`	If the Lighthouse UI fails to display after an upgrade, it's possible a NetOps UI component is failing to load and may be able to recover by running this command.

IP ACCESS

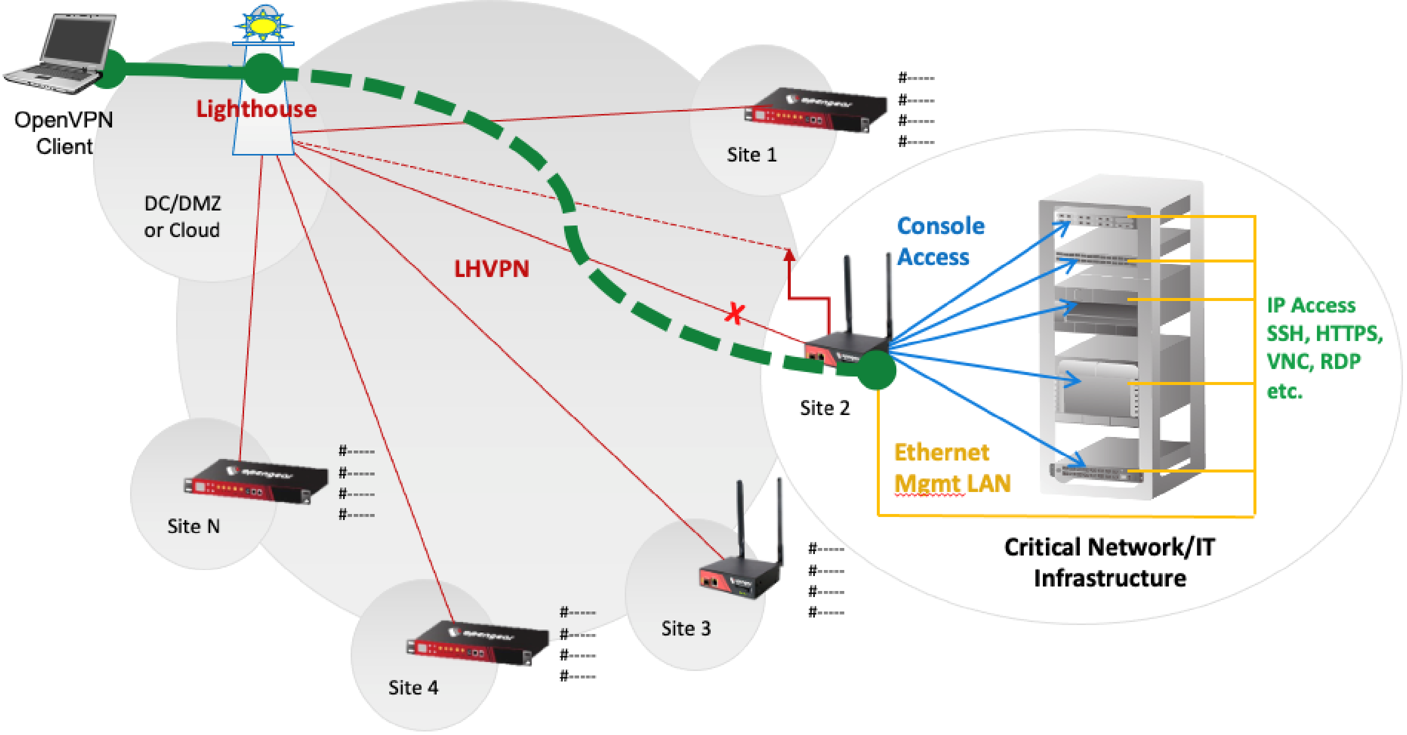
The Lighthouse IP Access feature allows an engineer to reach hosts on a remote site via an OpenVPN client through Lighthouse, over the Lighthouse VPN fabric, without physically traveling to the site. If IP Access is enabled for Lighthouse, it can be managed from using **NetOps > IP Access** on the Lighthouse web UI menu.

IP Access adds client VPN capability to Lighthouse. Network engineers, firewall and server administrators can launch a VPN client connection to Lighthouse, be authenticated, and then automatically connected to the remote site management network. The client PC has a secure VPN tunnel to the remote equipment the user is required to work on, providing the same TCP/IP access they would get if they travelled to the site and plugged into the management LAN.

The client can then access target devices on the remote network directly by their usual IP addresses and network ports. Requests from the client are masqueraded behind the node's IP address, so no additional routing configuration is required on the target devices.

CONNECTIVITY

By default, IP Access connects the client to the Management LAN of the Opengear appliance, or the interfaces in the LAN zone for the OM Series. A route for the directly attached subnet, plus any static routes configured on that interface (but never the default route) are also pushed automatically to the OpenVPN client.



In the diagram, the client PC has a virtual tunnel interface with a route to the yellow management network, and the user can access any target IP devices on the yellow network using their real IP addresses.

The basic configuration of this feature is:

- Activate the IP Access NetOps module – this starts the OpenVPN service in a Docker container on Lighthouse.
- Activate the IP Access NetOps module on each node you want to use for IP Access – this installs a remote connector service to allow the IP Access bridge to be created.
- Generate a certificate and export an associated OpenVPN client configuration file.
- Import the configuration into your preferred OpenVPN client.

The basic operation of this feature is:

- Connect the tunnel – this starts a connection to Lighthouse on UDP port 8194.
- Authenticate when prompted using your Lighthouse credentials, appending the node name to your Lighthouse username – client certificate authentication is automatic, this is a second factor of authentication.
- Wait a moment for the connection to complete – this builds the GRE bridge between the client and pushes routes to the node's remote network(s).

While connected, the client can access IP addresses on the node's remote network(s) LAN directly, for example, by using the ping command or by typing them into the browser address bar.

NODES SUPPORTED BY IP ACCESS

Opengear OM1200, OM2200, CM8100, ACM7000 and IM7200 nodes may be activated as IP Access nodes, to allow IP Access to their directly connected remote networks via Lighthouse.

> **Note:** Other vendors/models are not currently supported.

To view the available nodes supported by IP Access:

1. Select **NetOps > IP Access**.

   The **IP ACCESS** page displays.

2. Select the **NODE ACCESS** tab.

ADVANCED OPTIONS

The Advanced options of the IP Access page enable you to set a number of features, including setting expiry of certificates lifetime, changing default connection routes and enabling policies.

IP Access connects the client to the Management LAN or LAN zone by default, which is intended for deployments where the target devices are connected to those interfaces. If the Opengear appliances are deployed into a different or more complex network environment, then there are some advanced options that the Lighthouse administrator can use to control the IP Access connectivity, and these are described below.

CONNECT TO WAN ZONE

By default, IP access connects the client to the management LAN zone.

To enable IP Access for the WAN zone:

1. Select **NetOps > IP Access**.

   The **IP Access** page displays.

2. Select the **ADVANCED OPTIONS** tab.
3. Select the Push Wired WAN Routes checkbox to push routes for nodes' wired WAN interface to clients.
4. Click **Apply**.

Note: This is a global configuration and will affect all node that are enabled for IP Access.

In this case, the customer must have deployed Opengear appliances with the Network Interface (NET1) connected to the management network or facing target devices, and it results in IP Access connecting the client to the WAN (NET1) interface on OGCS.

NETWORK ACCESS POLICIES FOR OPERATIONS MANAGER

In a more complex deployment, Opengear appliances may be connected to multiple networks or virtual networks (VLANs), and in these cases it is often important to be able to control which of these networks each authenticated IP Access user is able to access.

This feature is only supported on Operations Manager or OM Series appliances, which support the zone-based firewall, designed to work with multiple VLANs and the optional built-in Ethernet switch with layer-3 capable ports. This flexibility and control is very useful, especially for customers who have a number of separate management networks (or VLANs) for different administrative teams.

The Network Access Policy mechanism on Lighthouse provides a way to dynamically map IP Access users, based on their group membership, to the firewall zone(s) that they can access on the Nodes. Each firewall zone is a collection of network interfaces which is configured on each Opengear appliance or Node. Firewall zones are used to provide policy abstraction by logical zone names – the physical or virtual interfaces on each Node may vary by site, but the zone names must stay consistent. It is recommended that zones and names are planned out in advance of implementation.

A firewall zone is a collection of network interfaces which is configured on each Opengear appliance or Node in Lighthouse terminology. The Network Access Policy mechanism on Lighthouse provides a way to map users, based on their group membership, to the firewall zone(s) that they can access on the Nodes.

UNDERSTAND ACCESS POLICIES

Putting it all together, when a user authenticates to Lighthouse, they are mapped into one or more group(s), which map into firewall zone(s), which allow authenticated users to reach the appropriate network interfaces(s), including switch ports or VLANs, via IP Access.

For example, users who belong to the security group may get mapped into the *secops* (security operations) zone. On each OM appliance, the appropriate switch port(s) and/or VLAN(s) for security operations should be configured to be in the *secops* zone.

Similarly, users in the server group may get mapped into the *serverops* zone, and again on the OM appliances the appropriate interfaces can be configured to be part of the *serverops* zone.

The result is that members of the security group get IP Access to the networks in the *secops* zone, and members of the server group get IP Access to the networks in the *serverops* zone, for each node that they connect to.

SET UP NETWORK ACCESS POLICIES

To enable this feature:

1. Select 品 **NetOps > IP Access** from the main menu.

   The **IP ACCESS** page displays.
2. Select the **ADVANCED OPTIONS** tab.
3. Check **Network Access Policies for Operations Manager Enabled**.
4. Click **Apply**.

SET ACCESS POLICIES

Note: A policy must be configured for each group whose members will use IP Access.

1. Select **NetOps > IP Access** from the main menu.

   The **IP ACCESS** page displays.

2. Select the **NETWORK ACCESS POLICIES** tab.

   > **Note:** The group to zone mapping column **ZONES** is empty by default.

3. Select the **GROUP** to view details.

   The **Group Details** page is displayed.

4. Click on **+ Add Zone** to add one or more firewall zones for this group.

5. Select the firewall zone and click **Add**.

   The **NETWORK ACCESS POLICIES** tab now displays the group with the Firewall zone.

TROUBLESHOOT IP ACCESS

The most effective way to troubleshoot IP Access is to view the logs.

VIEW THE DOCKER LOGS

Run the following command on Lighthouse, either as `root` or with `sudo` (for non-root admins):

```
sudo docker logs -t central-sdi
```

The logs for the `central-sdi` Docker container, which controls client IP Access display.

```
2022-10-03T10:03:19.051579136Z INFO:root:[NetOps-SDI node="nodes-36"
username="maverick"] VPN client authenticated
2022-10-03T10:03:19.094313762Z 2.29.37.12:65437 TLS: Username/Password
authentication succeeded for username 'maverick123:OM1208-UK2'
2022-10-03T10:03:19.422971278Z 2.29.37.12:65437 Control Channel: TLSv1.3, cipher
TLSv1.3 TLS_AES_256_GCM_SHA384, peer certificate: 2048 bit RSA, signature: RSA-
```

```
SHA256
2022-10-03T10:03:19.422999989Z 2.29.37.12:65437 [LH5-UK3-22] Peer Connection
Initiated with [AF_INET]2.29.37.12:65437
2022-10-03T10:03:22.000300953Z INFO:root:[NetOps-SDI node="nodes-36"
username="maverick123"] VPN client connected with IP 172.31.0.8 netmask
255.255.0.0
2022-10-03T10:03:22.084275387Z LH5-UK3-22/2.29.37.12:65437 OPTIONS IMPORT: reading
client specific options from: /tmp/openvpn_cc_3aa5d19c3ea09b3bbac56f6bacf7b6e.tmp
2022-10-03T10:03:22.084324213Z LH5-UK3-22/2.29.37.12:65437 Data Channel: using
negotiated cipher 'AES-256-GCM'
2022-10-03T10:03:22.084332293Z LH5-UK3-22/2.29.37.12:65437 Outgoing Data Channel:
Cipher 'AES-256-GCM' initialized with 256 bit key
2022-10-03T10:03:22.084337215Z LH5-UK3-22/2.29.37.12:65437 Incoming Data Channel:
Cipher 'AES-256-GCM' initialized with 256 bit key
2022-10-03T10:03:22.084342034Z LH5-UK3-22/2.29.37.12:65437 SENT CONTROL [LH5-UK3-
22]: 'PUSH_REPLY,ping 10,ping-restart 120,route 192.168.2.0 255.255.255.0
172.31.0.1 1,ifconfig 172.31.0.8 255.255.0.0,peer-id 0,cipher AES-256-GCM'
(status=1)
2022-10-03T10:03:22.084950386Z LH5-UK3-22/2.29.37.12:65437 PUSH: Received control
message: 'PUSH_REQUEST'
2022-10-03T10:03:22.085204467Z LH5-UK3-22/2.29.37.12:65437 PUSH: Received control
message: 'PUSH_REQUEST'
2022-10-03T10:03:22.085220316Z LH5-UK3-22/2.29.37.12:65437 PUSH: Received control
message: 'PUSH_REQUEST'
2022-10-03T10:03:22.432350278Z LH5-UK3-22/2.29.37.12:65437 PUSH: Received control
message: 'PUSH_REQUEST'
2022-10-03T10:03:23.120616769Z INFO:root:[NetOps-SDI node="nodes-36"
username="maverick"] VPN client identified by MAC c6:87:ca:4a:3b:2c
```

Note the PUSH_REPLY line above shows that the IP Access client has been pushed the route to the network 192.168.2.0/24 which in this case is the interface on this Node in the firewall zone that the user was mapped into. (Unfortunately, the firewall zones are not listed in this log output). If there are errors with authentication, then they will show up here.

USE THE ROUTING TABLE

When troubleshooting IP Access it is useful to look at the routing table on the target Node to make sure that routes to the target networks are installed. If the interface is down, for example, then the route is not present and will not be pushed to the client.

The following commands can be used to display the routing table on the target OM Series Node:

- `route`
- `ip route`

Displaying the Routing table with the `route` command:

```
root@OM1208-UK2:~# route

Kernel IP routing table

Destination Gateway Genmask Flags Metric Ref Use Iface

default _gateway 0.0.0.0 UG 110000001 0 0 net1

5.5.5.0 0.0.0.0 255.255.255.0 U 0 0 0 sw0p8

172.17.0.0 0.0.0.0 255.255.0.0 U 0 0 0 docker0

172.31.0.0 0.0.0.0 255.255.0.0 U 0 0 0 ipa-br0

192.168.0.0 0.0.0.0 255.255.255.0 U 0 0 0 net2

192.168.1.0 0.0.0.0 255.255.255.0 U 0 0 0 net1

192.168.2.0 0.0.0.0 255.255.255.0 U 0 0 0 sw0p2

192.168.128.0 0.0.0.0 255.255.224.0 U 0 0 0 tun0

root@OM1208-UK2:~#

root@OM1208-UK2:~# ip route

default via 192.168.1.1 dev net1 proto static metric 110000001
```

```
5.5.5.0/24 dev sw0p8 proto kernel scope link src 5.5.5.5
172.17.0.0/16 dev docker0 proto kernel scope link src 172.17.0.1 linkdown
172.31.0.0/16 dev ipa-br0 proto kernel scope link src 172.31.0.1
192.168.0.0/24 dev net2 proto kernel scope link src 192.168.0.48
192.168.1.0/24 dev net1 proto kernel scope link src 192.168.1.48
192.168.2.0/24 dev sw0p2 proto kernel scope link src 192.168.2.8
192.168.128.0/19 dev tun0 proto kernel scope link src 192.168.128.6
root@OM1208-UK2:~#
```

The `ip` command output is useful as it shows the interface IP address of the OM, which should be reachable via IP Access as long as that interface is up.

Other standard network troubleshooting techniques can be used from the Node. For example testing the ability to ping a target device, or using `curl` if it has an http or https interface.

Example:

```
ping 192.168.2.4
curl -k https://192.168.2.4/
```

AUTOMATION GATEWAY

Opengear allows you to set up your network using automation tools.

Automation Gateway allows Lighthouse users and automation code to discover and manage IP-based management interfaces via the Opengear management system, with the same level of simplicity and efficiency as if they were serial consoles.

Resources like firewalls and servers may present an IP-based management interface in addition to (or sometimes instead of) the traditional serial or USB console port. This interface may serve a web-based GUI, VNC or RDP based KVM, SSH-based CLI and/or a programmable network API like RESTful HTTPS. The device itself may be physical or virtual.

By their nature, the IP-based management interfaces are more dynamic (for example, they may change IP address), varied (for example, protocols vary from device to device) and harder to reach (for example, on an un-routable private network).

The Automation Gateway module addresses the challenges of discovering, auditing and connecting to IP-based management services in a distributed, heterogeneous network. It is available on Operations Manager product lines, OM120x and OM22xx.

Notes: Opengear is transitioning from Automation Gateway (AG) to Connected Resource Gateway (CRG) for access, control, and management of connected resources:

- CRG will be the primary solution moving forward, offering improved scalability, security, and continued enhancements.
- New users should begin with CRG as the recommended option, while existing AG users are encouraged to migrate to CRG.
- AG discovered resources are not automatically accessible on CRG.
- To migrate to CRG, you must set up CRG as if you are setting up a new implementation, see Set up Lighthouse as a Connected Resource Gateway (CRG).

DIFFERENCES BETWEEN IP ACCESS AND AUTOMATION GATEWAY

The **IP Access** module provides the **IP Access** feature and the **Automation Gateway** module provides the **Automation Gateway** feature.

These two features are similar in that they both allow Lighthouse users to access network services on remote resources, however they accomplish this in different ways. One way to think about it is that **IP Access** transports the user to the remote network, whereas **Automation Gateway** transports a specific remote network service to user.

IP ACCESS

Using **IP Access**, the user must establish a VPN tunnel from their computer to Lighthouse, which then provides them with a routed connection to the entire network(s) connected to the remote node. When the tunnel is up, the user can access any network service on any network device by their standard management IP addresses.

AUTOMATION GATEWAY

Using the Automation Gateway feature, the user clicks through their existing Lighthouse browser session to access HTTP/HTTPS web GUI services of specific devices that have been discovered by the remote node. Access is limited to these services only, and the connections are proxied via Lighthouse's central address – so no client or network reconfiguration is required.

HOW TO USE AUTOMATION GATEWAY

Using Automation Gateway, Lighthouse users can connect to the web UI of a remote physical & virtual resource such as a firewall, lights-out server or SD-WAN appliance.

Access is proxied via Lighthouse VPN via a remote node, allowing simply, point & click access to what may be an otherwise unreachable remote device.

TO ACTIVATE THE AUTOMATION GATEWAY ON AN ENROLLED NODE

> **Note:** To perform these steps, a Lighthouse web UI as a user with at least **NetOps Modules: Read Only** permission, and at least **Nodes & Devices (Base): Read Only** permission for the activated node.

1. Select **NetOps > Automation Gateway** from the main menu.

   The **AUTOMATION GATEWAY** page displays.

2. In the **FILTER BY** menu, select *HTTPS* or *HTTP* from the **Services** drop-down.
3. Locate the remote device by hostname or IP address.

   > **Tip:** If many remote devices have been discovered, use the FILTER BY menu to search by full or partial device hostname or IP address.

4. To initiate an Automation Gateway session, click the device's web UI icon.

   You are now connected the web UI of the remote device.

> **Note:** While an Automation Gateway session is active, all new browser tabs and windows that connect to Lighthouse are proxied through to the remote device web UI.

5. To close the Automation Gateway session, click the link at the bottom of the device web UI:

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

CONNECT WITH REST/HTTP/HTTPS APIS

Automation Gateway allows central automation code to reach the HTTP/S API services running on devices on remote networks.

For more details refer to the latest documentation:

- https://ftp.opengear.com/download/documentation/api/lighthouse/

This is accomplished by modifying the request to add the X-Ip-Access-Auth HTTP header to your API request, and substituting the device's remote IP with Lighthouse's central IP. Requests containing the header are reverse proxied via Lighthouse then via an Operations Manager node that is local to the remote device.

The example Python (3.5+) code below illustrates this. Note that this code is primarily for illustrative purposes with no error handling or modularity, for clarity & brevity.

The device being accessed in this example is an HP ILO's Redfish REST API service.

```
#!/usr/bin/env python3
# Example code showing how to reach a remote device REST API via Opengear
# Lighthouse Automation Gateway -> Operations Manager node -> device
#
# This code is primarily for illustrative purposes with no error handling
# or modularity, for clarity & brevity

import requests
import json
import base64

# Authenticate to Lighthouse

lighthouse_address = '192.168.67.20'
lighthouse_account = 'root'
lighthouse_password = 'default'

data = { 'username': lighthouse_account, 'password': lighthouse_password }
r = requests.post('https://%s/api/v3.7/sessions/' %
  lighthouse_address, data=json.dumps(data), verify=False)
lh_token = json.loads(r.text)['session']

print('Authenticated to Lighthouse, token %s' % lh_token)

lh_headers = { 'Authorization': 'Token ' + lh_token }

# Find the node that is local to the remote device's API service

device_address = '10.0.0.71' # Equivalent to the UI fields under CONFIGURE >
device_service = 'https' # AUTOMATION GATEWAY > Devices > Filter By
```

```
r = requests.get('https://%s/api/v3.7/nom/ag/devices?ip=%s&service=%s' %
  (lighthouse_address, device_address, device_service),
  headers=lh_headers, verify=False)
j = json.loads(r.text)

print(json.dumps(j, indent=4))

for _device in j['devices']:
 for _host in _device['hosts']:
  for _service in _host['services']:
   if _service['nmap']['name'] != device_service:
    continue
   for _avail in _service['availability']:
    node_id = _avail['id']
    print('Service available via %s' % node_id)
    break

# Generate Automation Gateway token

data = { 'session': lh_token, 'request_type': 'new',
  'url': 'https://%s' % device_address, 'node': node_id }
r = requests.post('https://%s/api/v3.7/nom/ag/auth_tokens' %
  lighthouse_address, data=json.dumps(data), headers=lh_headers, verify=False)
j = json.loads(r.text)
ag_token = j['token']

print('Automation Gateway token is %s' % ag_token)

ag_headers = { 'X-Ip-Access-Auth': ag_token }
```

```
# Now we can query the device API using the Lighthouse address, by including
# the access token in the request headers

# The remaining code is specific to the device being accessed. this example
# hits a Redfish-compliant REST API

device_username = 'baz'
device_password = 'foobarfoobar'
device_auth = base64.b64encode(('%s:%s' %
  (device_username, device_password)).encode('utf-8')).decode('utf-8')
device_headers = { 'Authorization': 'Basic %s' %
  device_auth, 'Content-Type': 'application/json' }

# Add the access token to whatever headers you'd usually use to talk to the
# device's native API -- this header will be stripped by Automation Gateway

_headers = { **ag_headers, **device_headers }

r = requests.get('https://%s/redfish/v1/systems/1/' %
  lighthouse_address, headers=_headers, verify=False)
j = json.loads(r.text)
print(json.dumps(j, indent=4))
```

AUTOMATION GATEWAY SERVICE DISCOVERY

Automation Gateway discovery process can take a varied amount of time to complete, entirely based on the size of your scannable network.

Services discovered by Automation Gateway are listed in the Lighthouse web UI, under **NetOps > Automation Gateway**. The discovery process can be restarted using the **Rescan Devices** button on the **NetOps > Automation Gateway** page.

When an HTTP or HTTPS service has been discovered, it may also be accessed via this page.

When a node has been activated for Automation Gateway, it begins to discover remote services. The discovery process is initiated by Lighthouse, and runs every 10 minutes.

Each time the discovery process is initiated, the node runs an nmap script scan against all IPv4 connections belonging to the node's LAN firewall zone.

> **Note:** Large logical networks with address space larger than 254 hosts (i.e. with a minimum netmask of /24 or 255.255.255.0) are excluded from the scan.

The nmap scan runs the *default* (non-intrusive) suite of nmap NSE scripts. These can be listed by running the following command on a node that has been activated for Automation Gateway:

```
sudo docker exec ag-remote cat /usr/share/nmap/scripts/script.db | awk -F\"
'/"default"/ { print $2 }'
```

ACCESS MULTIPLE VLANS OR PORTS

There are several ways a user can access multiple target networks, virtual networks (VLANs) or physical ports.

GROUP MEMBERSHIPS

A user may belong to multiple groups, in which case they will have access to the sum of the zones mapped to those groups, in the same way port access works for Console Gateway and Node

Filters matching. Note that this works regardless of Local or Remote user authentication on Lighthouse; user authorization (which determines access to Nodes and resources, and now firewall zones) is always derived from group membership.

FIREWALL ZONES

In the **NetOps > IP Access** dashboard under the **NETWORK ACCESS POLICIES** tab, each group can be configured to have access to one or more firewall zones. Some groups can be configured to have access to no zones, some to just one zone, and other groups to have access to multiple zones.

MULTIPLE LAYER 3 NETWORK CONNECTIONS

On each Node a layer 3 network connection or “conn” is required on the OM to communicate with other hosts on a network or VLAN. Multiple conns on each OM can be mapped into the same firewall zone. This may be used to provide access to multiple switch ports, though it is perhaps more likely that those switch ports would be configured in a bridge group if they are all part of the same LAN, and the bridge group only requires a single layer 3 conn. If multiple LANs or virtual LANs (VLANs) are managed by the same team, then it may make sense to combine them into the same firewall zone.

Warning: The supported appliance's firewall will allow traffic to pass between interfaces in the same firewall zone, so to maintain security, multiple “separate” management VLANs should not be configured in the same zone, but should each have its own zone. If required, one of the mechanisms above can be used to allow user access to multiple zones and therefore to multiple “separate” VLANs.

COMMAND LINE TOOLS

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

1. Select ⌨ **Terminal**.

   A login prompt displays

2. Enter an administrator’s username and press **Enter**.

   A password: prompt displays.

3. Enter the administrator’s password and press **Enter**.

   A bash shell prompt displays.

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

CERT_MANAGE

Caution:

- Running `cert_manage` run may cause nodes to temporarily disconnect from Lighthouse, and/or secondary Lighthouse instances to temporarily disconnect from the primary.
- The certificate manager may take a while to complete. If running manually, do not interrupt the process.
- Renewing a CA certificate will result in each node being updated with the new CA details. A status summary of these update jobs can be shown using `cert_manage status`.
  It is recommended that these be allowed to complete before using `cert_manage run` to make further changes. If renewing the CA and Lighthouse VPN certificates, the certificate processing will occur on separate scheduled runs to avoid a limitation where the Lighthouse VPN certificate is renewed before all nodes are informed of the CA renewal.
  If the Lighthouse VPN certificate renewal is forced too early, any nodes that did not receive the CA renewal notification will be disconnected and will require re-enrollment into Lighthouse.

Command	Description
`cert_manage`	• The cert_manage CLI tool can only be run on a primary lighthouse, by a root user. A lighthouse user may sudo to assume root permissions. • This tool cannot be run on a secondary Lighthouse. • The cert_manage tool is not tied to a specific license or subscription. • Certificate renewal jobs are scheduled using cron to run at 1am (Lighthouse time), every day. An administrator may choose to update the frequency of the cron job under `/etc/cron.d/rotate_certificates.cron`.

Syntax
`cert_manage [-h] [--config CONFIG_URI] {run,renew,show,status,validity,offset,logs}`

Option	Description
`-h | --help`	Display usage information and exit.
`--config CONFIG_URI`	Config file for lipy.

Sub-Command	Description
`run`	Process certificates due for renewal (current time > renewal time) and rotate them.
`renew`	Schedule certificates for early renewal. Updates the renewal time of specified certificates to current time. The next certificate manager run will renew them. This will happen overnight or can be triggered using the run command
`show`	Show certificate or default value information. Display information for all specified certificates, or it can display the default values that are used for newly created certificates.
`status`	Show status summary including job status.
`validity`	Set default validity period for new certificates.

Sub-Command	Description
`offset`	Set renewal offset period in days to update the delta between expiry and renewal time for the specified certificates. The renewal period is calculated from the certificate expiry date using the offset.
`logs`	Prints the last 20 logs from the log file `/var/log/cert_manager.log`.

SUB-COMMAND :: RUN

Syntax
`cert_manage run [-h] [--dry-run]`

Option	Description
`-h | --help`	Display usage information and exit.
`--dry-run`	Lists affected certificates.

SUB-COMMAND :: RENEW

Syntax

```
cert_manage renew [-h] (--cn COMMON_NAMES [COMMON_NAMES ...] | --ca | --nodes | --
all-clients) [--dry-run]
```

Option	Description
`-h | --help`	Display usage information and exit.
`--cn COMMON_NAMES [COMMON_NAMES ...]`	Specify certificates by Common Name.
`--ca`	Certificate Authority certificate.
`--nodes`	Certificates for all enrolled nodes.
`--all-clients`	All existing client certificates.
`--dry-run`	Lists affected certificates without executing the action.

SUB-COMMAND :: SHOW

Syntax
`cert_manage show [-h] (--cn COMMON_NAMES [COMMON_NAMES ...] | --ca | --nodes | --all-clients | --defaults)`

Option	Description
`-h | --help`	Display usage information and exit.
`--cn COMMON_NAMES [COMMON_NAMES ...]`	Specify certificates by Common Name.
`--ca`	Certificate Authority certificate.
`--nodes`	Certificates for all enrolled nodes.
`--all-clients`	All existing client certificates.
`--defaults`	Show certificate defaults.

SUB-COMMAND :: VALIDITY

Syntax
`cert_manage validity [-h] --days NUM_DAYS (--default-client | --default-ca) [--dry-run]`

Option	Description
`-h | --help`	Display usage information and exit.
`--days NUM_DAYS`	Number of days from current time.
`--default-client`	Specify the default value for new client certificates.
`--default-ca`	Specify the default value for new CA certificates.
`dry-run`	Lists affected certificates without executing the action.

SUB-COMMAND :: OFFSET

Syntax
`cert_manage offset [-h] --days NUM_DAYS (--cn COMMON_NAMES [COMMON_NAMES ...] | --ca | --nodes | --all-clients | --default-client | --default-ca) [--dry-run]`

Option	Description
`-h | --help`	Display usage information and exit.
`--days NUM_DAYS`	Number of days from expiry.
`--cn COMMON_NAMES [COMMON_NAMES ...]`	Specify certificates by Common Name.
`--ca`	Certificate Authority certificate.
`--nodes`	Certificates for all enrolled nodes.
`--all-clients`	All existing client certificates.
`--default-client`	Specify the default value for new client certificates.
`--default-ca`	Specify the default value for new CA certificates.
`dry-run`	Lists affected certificates without executing the action.

SUB-COMMAND :: LOGS

Syntax
`cert_manage logs [-h]`

Option	Description
`-h | --help`	Display usage information and exit.

CRON

Command	Description
`crontab`	The `cron` service can be used to schedule file execution at specific times. The server daemon executes commands at specified dates and times based on the entries in the cron jobs table. The service can be managed via the `/etc/init.d/crond` interface, and the cron tables managed via `crontab`.

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

Option	Description
`-u <user>`	Define user.
`-e`	Edit user's crontab. • Each line can contain one command to run. • The following format is used: `minute hour day-of-month month day-of-`

Option	Description
	`week command`. • When finished, save and close the `crontab` file.
`-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.

Related Command	Description
`/etc/init.d/crond start`	To start the `crond` service.
`/etc/init.d/crond stop`	To stop the `crond` service.
`/etc/init.d/crond restart`	To restart the `crond` service.
`/etc/init.d/crond status`	To verify the current crond status.

MASS_ENROLL_NODES

Command	Description
`mass_enroll_nodes`	A tool to enroll nodes.

Syntax
`mass_enroll_nodes [-h] [--lh-address LH_ADDRESS] [--dry-run] [--insecure] [--auto-approve] [--preflight] --lh_username LH_USERNAME --lh_password LH_PASSWORD --node_addresses NODE_ADDRESSES --node_username NODE_USERNAME --node_password NODE_PASSWORD`

Required	Option	Alternate Syntax	Description
	`-h`	`--help`	Display usage information and exit.
	`-a <LH_ADDRESS>`	`--lh-address <LH_ADDRESS>`	The address of the Lighthouse to enroll nodes to (default=localhost).
	`-d`	`--dry-run`	Don't actually enroll anything - just list the nodes that would be enrolled.
	`-i`	`--insecure`	Don't verify https cer-

Required	Option	Alternate Syntax	Description
			tificates. This is only recommended for localhost.
	`-x`	`--auto-approve`	Automatically approve nodes when they reply.
	`y`	`--preflight`	Run the preflight enrollment check.
Y	`-u <LH_USERNAME>`	`--lh_username <LH_USERNAME>`	The username of a Lighthouse root/admin user.
Y	`-p <LH_PASSWORD>`	`--lh_password <LH_PASSWORD>`	The password of a Lighthouse root/admin user.
Y	`-n <NODE_ADDRESSES>`	`--node_addresses <NODE_ADDRESSES>`	A comma-separated list of opengear node addresses. They must all have the same username/password.
Y	`-U <NODE_USERNAME>`	`--node_username <NODE_USERNAME>`	The username for all of the nodes to be enrolled.
Y	`-P <NODE_PASSWORD>`	`--node_password <NODE_PASSWORD>`	The password for all of the nodes to be enrolled.

MASS_UNENROLL_NODES

Command	Description
`mass_unenroll_nodes`	A tool to unenroll nodes.

Syntax
`mass_unenroll_nodes [-h] [--lh-address LH_ADDRESS] [--dry-run] [--verbose] [--insecure] --lh-username LH_USERNAME --lh-password LH_PASSWORD (--all | --never-seen | --approved | --connected | --disconnected | --pending | --pending-connected | --pending-disconnected)`

Required	Option	Alternate Syntax	Description
	`-h`	`--help`	Display usage information and exit.
	`-a <LH_ADDRESS>`	`--lh-address <LH_ADDRESS>`	The address of the Lighthouse to unenroll nodes to (default=localhost).
	`-d`	`--dry-run`	Don't actually delete anything - just list the nodes that would be deleted.

Required	Option	Alternate Syntax	Description
	`-v`	`--verbose`	List the number of nodes in each state.
	`-i`	`--insecure`	Don't verify https certificates. This is only recommended for localhost.
Y	`-u <LH_USERNAME>`	`--lh_username <LH_USERNAME>`	The username of a Lighthouse root/admin user.
Y	`-p <LH_PASSWORD>`	`--lh_password <LH_PASSWORD>`	The password of a Lighthouse root/admin user.
	`--all`		Unenroll all nodes.
	`--never-seen`		Unenroll nodes that have never been seen.
	`--approved`		Unenroll all approved nodes.
	`--connected`		Unenroll all connected nodes.
	`--disconnected`		Unenroll all disconnected nodes.
	`--pending`		Unenroll all nodes that are pending.
	`--pending-connected`		Unenroll nodes that are

Required	Option	Alternate Syntax	Description
			pending, and connected.
	`--pending-disconnected`		Unenroll nodes that are pending, but disconnected.

Note: The following options are related to the unenrollment strategy and only one option to be selected when entering the command:

```
(--all | --never-seen | --approved | --connected | --disconnected | --pending | --pending-connected | --pending-disconnected)
```

NODE-COMMAND

Command	Description
`node-command`	A shell-based tool for pulling more detailed information from console servers.

Syntax
`node-command [options] command`

Option	Alternate Syntax	Description
`-h`	`--help`	Display usage information and exit.
`-l`	`--list-nodes`	List all nodes matching query, or all nodes if none selected.
`-i <ID>`	`--node-id=ID`	Select node by config ID.
`-n <name>`	`--node-name=name`	Select node by name.
`-a <address>`	`--node-address-s=address`	Select node by VPN address.

Option	Alternate Syntax	Description
`-g <name>`	`--smartgroup=name`	Select nodes by the smart group they resolve to.
`-A`	`--all`	Select all available nodes.
`-C`	`--connected>`	Select all connected nodes.
`-P`	`--pending`	Select all pending nodes.
`-D`	`--disconnected`	Select all disconnected nodes.
`-d`	`--disable-fslog`	Disable logging to file system.
`-p`	`--par=num`	Number of parallel processes to run at a time.
`-q`	`--quiet`	Suppress command output.
`-b`	`--batch`	Suppress node-command output.
`-c`	`--copy-file`	Copy a file to the node(s).
`-s`	`--source-file`	The file to copy to the node(s). Required if `--copy-file` is specified.
`-e`	`--exact`	Run the command exactly as entered. Avoids quoting problems caused by shell interpretation.
`--`		Use this to stop argument parsing. This allows arguments that node-command understands to be passed to commands on the node. eg. `node-command [options] -- command [args]`

NODE-INFO

Command	Description
`node-info`	A shell-based tool for pulling more detailed information from console servers.

Syntax
`node-info [options]`

Option	Alternate Syntax	Description
`-h`	`--help`	Display usage information and exit.
`-l`	`--list-nodes`	List all nodes matching query, or all nodes if none selected.
`-i <ID>`	`--node-id=ID`	Select node by config ID.
`-n <name>`	`--node-name=name`	Select node by name.
`-a <address>`	`--node-address=address`	Select node by VPN address.
`-g <name>`	`--smartgroup=name`	Select nodes by the smart group they resolve to.

Option	Alternate Syntax	Description
`-A`	`--all`	Select all available nodes.
`-C`	`--connected>`	Select all connected nodes.
`-P`	`--pending`	Select all pending nodes.
`-D`	`--disconnected`	Select all disconnected nodes.

EXAMPLE

```
$ 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
```

NODE-UPGRADE

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

Syntax

```
node-upgrade [-h] [-q | -b | -V | -D] [-l] [-I] [-n NODE_NAME] [-i NODE_ID] [-a
NODE_ADDRESS] [-A] [-p {ACM500X,ACM550X,ACM700X,ACM7004-
5,CM71XX,CM7196,IM42XX,IM72XX,OMXXXX,CM81XX}] [-g {Enrolled Nodes 2,Connected
Nodes,Disconnected Nodes 2,Cell Health Sim Issues,Cell Health Connectivity Check
Failed,retester,Good Health,SamTestFilter,JW Test}] (-f FIRMWARE_DIR | -F FIRMWARE_
FILE) [-v VERSION]
```

Option	Alternate Syntax	Description
`-h`	`--help`	Display usage information and exit.
`-q`	`--quiet`	Suppress log messages.
`-b`	`--batch`	DEPRECATED: use `--quiet` instead.
`-V`	`--verbose`	Display logs generated while upgrading.

Option	Alternate Syntax	Description
`-D`	`--debug`	Display detailed log messages, implies --verbose.
`-l`	`--list-nodes`	Display nodes and their upgradeable paths without executing upgrade.
`-I`	`--ignore-version`	Ignore firmware version warnings for upgrade.
`-f <directory>`	`--firmware-dir <directory>`	The directory of the firmware files(s). This is the directory node-upgrade looks to for the firmware image used as the source for all the firmware upgrade attempts.
`-F <path>`	`--firmware-file <path>`	The path to the firmware image to use for upgrade.
`-v <version>`	`--version <version>`	The firmware version to upgrade to.
`-n <name>`	`--node-namee=<name>`	Select a specific node by its name, this option may be used multiple times.
`-i <id>`	`--node-id <id>`	Select a specific node by its ID, this option may be used multiple times.
`-a <address>`	`--node-address <address>`	Select a specific node by its Lighthouse VPN address, this option may be used multiple times.
`-A`	`--all`	Select all nodes. All other node selection options are ignored if used at the same time.
`-p <family>`	`--product <family>`	Select node by product family where values available are: {ACM500X,ACM550X,ACM700X,ACM7004-

Option	Alternate Syntax	Description
		5,CM71XX,CM7196,IM42XX,IM72XX,OMXXXX,CM81XX}
`-g <name>`	`--smartgroup <name>`	Select nodes by smartgroup. When using along side `--product`, 'product' and 'smartgroup' are used with an 'AND' operator.

EXIT STATUS

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

OUTPUT MESSAGES

Result	Causes
SUCCESS	Node upgrade succeeded.
FileNotFoundError	No upgrade file found matching provided device family or version.

Result	Causes
UpgradeError	Device already has same or higher firmware version or network connection lost.
IncompatibleFirmwareError	Firmware file provided does not match the product family.

EXAMPLE

```
# 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
```

Note: The information in the output shows the device version prior to upgrade.

SUPPORT-REPORT

Command	Description
`support-report`	Tool to generate a Lighthouse support report. This program should be run as super-user. All output is sent to stdout, and should be redirected to a file if required.

Syntax
`support-report [-h] [--zip] [--content-config CONTENT] [--config CONFIG_URI]`

Option	Long Option	Description
`-h`	`--help`	Display usage information and exit.
`-z`	`--zip`	Create .zip archive, containing even more logs, config, and information.
`-c CONTENT`	`--content-config CONTENT`	Override default content (YAML) configuration file.
	`--config CONFIG_URI`	Config file for lipy.

EXAMPLE

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

SYSFLASH

Command	Description
`sysflash`	Is a 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.

Syntax
`sysflash [options] (filename|URL)`

Option	Long Option	Description
`-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).

Option	Long Option	Description
`-r`	`--no-reboot`	Do not reboot after upgrading.
	`filename|URL`	Location of the upgrade image: `filename`- path to system image `.lg_upg`. Filename cannot include spaces. `URL`- percent-encoded or quoted URL to firmware image `.lg_upg`
`-h`	`--help`	Display usage information and exit.

TRAFFIC_MIRRORING

Command	Description
`traffic_mirroring`	Is a tool that allows network administrators to set up an integration with their enterprise Intrusion Detection System (IDS). Network Traffic Mirroring can only be configured by a network administrator with `sudo` access and is available only through a command line interface.

Syntax

```
traffic_mirroring [-h] [--config CONFIG_URI] [--enable | --disable | --status | --
test] [--instance-id INSTANCE_ID] [--destination-ip DESTINATION_IP] [--vlan-id VLAN_
ID] [--mirror-smf] [--ignore-multi-instance]
```

Option	Description
`-h | --help`	Display usage information and exit.
`--config <CONFIG_URI>`	Config file for lipy.
`--enable`	Enable traffic mirroring.
`--disable`	Disable network traffic mirroring.

Option	Description
`--status`	Get the current status of the traffic mirroring config. Can be executed on a secondary Lighthouse.
`--test`	Test the setup by sending a single ping on each VPN and attempt to confirm that the current setup is valid, and the correct rules and interfaces exist, and the destination IP is reachable. Execute on a secondary Lighthouse to check if the network traffic mirroring is in use.
`--instance-id <INSTANCE_ID>`	The instance ID of the Lighthouse for which to manage traffic mirroring. If omitted, all Lighthouses will be affected.
`--destination-ip <DESTINATION_IP>`	The destination IP where mirrored traffic will be sent to (e.g: Intrusion Detection System). Because mirrored traffic preserves the original source/destination IP address, this IP should be on the same subnet as Lighthouse, or it may not be possible to get the MAC address to route the traffic to the destination.
`--vlan-id <VLAN_ID>`	The VLAN ID to use for traffic mirroring. This will result in a new vlan-tagged interface called 'mirror' being created, where ALL traffic to the destination IP will be routed through, even traffic that is not created by traffic mirroring (e.g. ping, ssh).
`--mirror-smf`	Mirror traffic for the Smart Management Fabric (SMF). SMF traffic is encapsulated in a Wireguard VPN tunnel within the OpenVPN tunnel that connects nodes together. Thus, to mirror the correct data for this tunnel, it must be handled twice. Reminder: The amount of traffic sent from Lighthouse is already doubled by enabling traffic mirroring. If this is enabled, then SMF traffic will be tripled (instead of doubled), and all other non-SMF traffic will still be doubled. This may result in huge amounts of traffic being sent by Lighthouse, as SMF can route traffic from many hosts, and the traffic might be bouncing multiple times (e.g. From a node, to LH, to a secondary LH, to another node). Be careful, and assess how you are planning to use SMF, and if you are capable of handling the network load.
`--ignore-multi-instance`	Do not mirror traffic between multi-instance Lighthouses. Only mirror traffic between Lighthouse and nodes.

EXAMPLE :: ENABLE

```
root@lighthouse:~# traffic_mirroring --enable --destination-ip 10.97.100.1 --vlan-
id 100
Configuring for Primary Lighthouse instance (Instance ID: 1).
Confirming that Traffic Mirroring is set up...
Traffic Mirroring successfully enabled.
```

EXAMPLE :: STATUS

```
root@lighthouse:~# traffic_mirroring --status
Primary Lighthouse 1 has traffic mirroring enabled.
Mirroring Node VPN (tun0) -> 10.97.100.1 (VLAN 100)
Mirroring Multi Instance VPN (tun1) -> 10.97.100.1 (VLAN 100)
```

EXAMPLE :: TEST

```
root@lighthouse:~# traffic_mirroring --test
Preparing to test Node VPN (tun0) interface.
Pinging address '192.168.128.2' on interface 'tun0'.
Preparing to test Multi Instance VPN (tun1) interface.
Pinging address '172.16.1.2' on interface 'tun1'.
Test complete. The pings sent across the VPN's should have been mirrored.
```

EXAMPLE :: DISABLE

```
root@lighthouse:~# traffic_mirroring --disable
Traffic Mirroring disabled.
```

GLOSSARY

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

Term	Definition
AUTHDOWNLOCAL (RADIUS/LDAP/TACAS)	When AUTHDOWNLOCAL 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/TACAS)	When AUTHLOCAL 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.
CONNECTED RESOURCE GATEWAY	A catalog of resources that are within the Smart Management Fabric discovered networks with support for clientless network access to resources via either SSH, HTTP, or HTTPS proxy services.
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	>A 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.
INTRUSION DETECTION SYSTEM	An Intrusion Detection System (IDS) is a network security technology built for detecting vulnerability exploits against a target application.
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 LOCALAUTH 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.
OSPF	OSPF (Open Shortest Path First) is an interior gateway protocol used to distribute routing information within a single autonomous system. It is based on link-state technology. OSPF routers exchange link-state information with their neighbors to build a complete map of the network topology. This information is used to calculate the shortest path to each destination using Dijkstra's algorithm. OSPF supports multiple paths of equal cost and can load balance traffic across these paths.
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. With Lighthouse 24.06 onwards, Smart Groups are now renamed to Node Filters within the Lighthouse UI. However, Smart Groups retain their naming within the CLI and API.
SMART MANAGEMENT FABRIC	Smart Management Fabric (SMF) is a turnkey management network overlay that uses dynamic routing to allow IP connectivity to IT resources regardless of whether these are connected via USB, serial, SSH, HTTPS (GUI), SPs/BMCs (iLO, iDRAC, etc.), RDP, Ansible, Python, vCenter or other commonly used technologies.
TAG	User-defined attribute and value that is assigned to one or more nodes or ports. Tags are used when creating Smart Groups for filtering views or access to nodes and ports.
THIRD-PARTY NODE	A third-party node is any device that is not an Opengear node; and is enrolled via the Lighthouse Web UI.