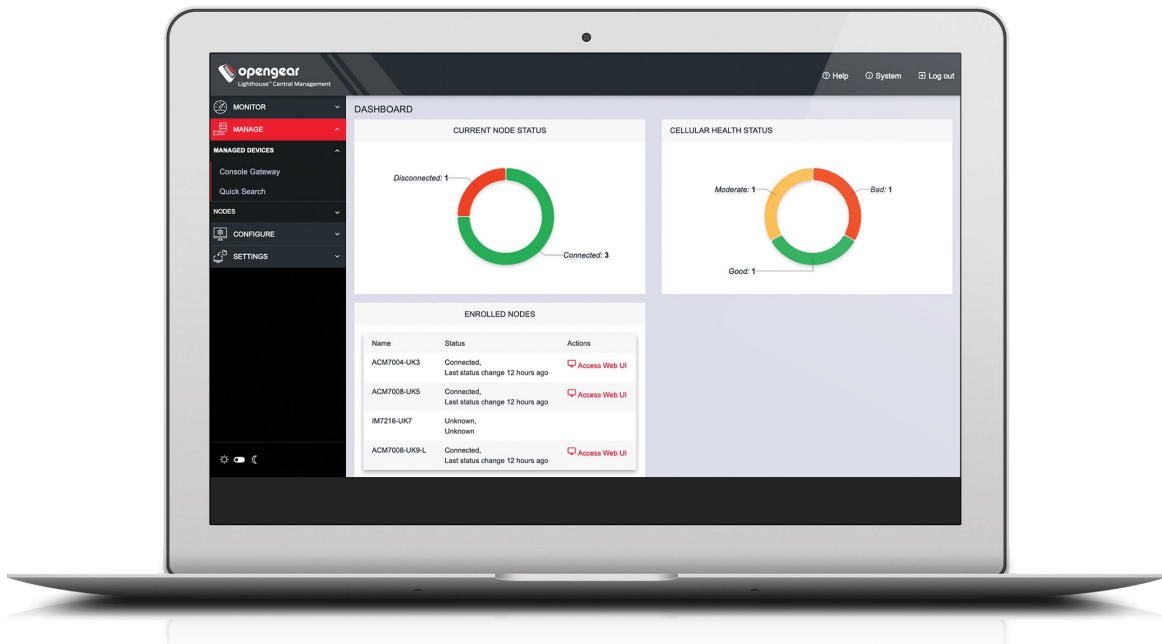


# Lighthouse

## User Guide

### 22.11 - Revision 1



## Copyright ©

Opengear Inc. 2022. All Rights Reserved.

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.

## Document Revision History

Document Version Number	Revision Date	Description
22.11 - REVISION 0		Additions: <ul style="list-style-type: none"><li>• Subscriptions</li><li>• Netops modules</li></ul> Removals <ul style="list-style-type: none"><li>• Zapier related content</li></ul>
22.11 - REVISION 1	Nov 2022	Update to Lighthouse configuration values on page 120 in Configuring the Subnets.

22.11 - REVISION 1	Document Revision History	3
--------------------	---------------------------	---

## Contents

<b>Copyright ©</b>	<b>2</b>
<b>Document Revision History</b>	<b>3</b>
<b>About This User Guide</b>	<b>16</b>
<b>Lighthouse Overview</b>	<b>17</b>
Lighthouse Architecture	18
The Lighthouse Virtual Machine Requirements	19
Lighthouse to Node interactions	21
Using Multiple Lighthouses	22
EULA and GPL	24
<b>Installing Lighthouse</b>	<b>25</b>
Lighthouse Virtual Machines Availability	26
<b>Installing Lighthouse VM on VMware</b>	<b>28</b>
VMware vSphere 6.0 client on Windows	29
Requirements	30
Launch the vSphere Client and connect to a vSphere instance.	31
Import the Lighthouse VM Open Volume Format (.ovf) image	33
Launch Lighthouse	35
Access the Console of a Running But Headless Lighthouse Instance	36
<b>VMware Workstation Player on Windows as Host</b>	<b>37</b>
Import the Opengear Lighthouse VM file into VMware Workstation Player	38
<b>VMware Workstation Pro on Windows as Host</b>	<b>39</b>
Import the Lighthouse VM File Into VMware Workstation Pro	40
<b>Installing Lighthouse VM on Hyper-V on Windows</b>	<b>42</b>
Local deployment on Hyper-V running on Windows 10/Windows Server 2016	43

Remote Hyper-V Deployment With Pre-authenticated User	44
VirtualBox Deployments	46
VirtualBox on Windows as host	47
VirtualBox on macOS as host	50
VirtualBox on Ubuntu as Host	53
VirtualBox on Fedora Workstation as host	55
Installing Lighthouse VM on Linux Hosts	57
Virtual Machine Manager (KVM) on Ubuntu as host	58
Boxes on Fedora Workstation as host	59
Boxes on RHEL and Compatible Distributions	61
Installing in the Cloud	62
Installing in Azure	63
Installing in Amazon Web Services(AWS)	65
Create a Lighthouse AMI Image	66
About the Lighthouse Bootstrap script	67
After creating the AWS EC2 Linux instance:	68
Set a password for the root user on Lighthouse:	69
Adding Disk Space to Lighthouse	70
Adding a New Disk to AWS	71
Adding a New Disk - qemu specific instructions	72
Adding a New Disk - Azure	73
Adding a New Disk - Hyper-V	74
Adding a New Disk - VirtualBox	75
Increase the lh_data logical volume	76
First Boot of the Lighthouse VM	77
<b>Setting up Lighthouse</b>	<b>79</b>
Loading Lighthouse	80
To load Lighthouse	81
Lighthouse IP addresses	82
22.11 - REVISION 1	Document Revision History
	5

Logging into Lighthouse	84
Setting the Lighthouse hostname	86
Adding external IP addresses manually	87
The Lighthouse SSL Certificate	89
Set the Lighthouse Internal Clock	90
<b>Setting up Networking Requirements</b>	<b>92</b>
Examine or Modify the Lighthouse Sessions	93
Examine or Change the MTU of the Lighthouse VPN Tunnel	94
Network Connections	95
Edit a network interface	96
Configure SNMP Manager Settings	97
Examine or modify SNMP Service	99
Cellular Health Dashboard	101
Changing Docker IP Ranges	102
Lighthouse MIBs	104
Examples of Lighthouse MIB queries using SNMP:	107
<b>Setting up Multiple Instances of Lighthouse</b>	<b>109</b>
Setting up a multiple instance	111
Enabling Alternate REST API Ports	114
Configuring Subnets for a Multiple Instance Lighthouse	116
Configuring the Subnets	117
Disconnecting a secondary instance	121
Promoting a secondary instance	122
Upgrading a multiple instance Lighthouse	123
Upgrading Dependent Multiple Instances of Lighthouse	124
<b>Upgrading Lighthouse</b>	<b>125</b>
Upgrading the system from within Lighthouse	126
Upgrading the Lighthouse system via the local terminal	128
Upgrade NetOps Modules	130
Configuration Backup	133



# LIGHTHOUSE

## Central Management

Configuration restore \_\_\_\_\_ 134

**Shut Down or Restart Lighthouse \_\_\_\_\_ 135**

Finding the current Lighthouse instance version \_\_\_\_\_ 136

Using Lighthouse \_\_\_\_\_ 137

Via the local Lighthouse shell \_\_\_\_\_ 138

Via other information sources \_\_\_\_\_ 139

Returning a Lighthouse instance to factory settings \_\_\_\_\_ 140

Shut Down a Running Lighthouse Instance \_\_\_\_\_ 141

Restarting a Running Lighthouse Instance \_\_\_\_\_ 142

**Managing Lighthouse Nodes \_\_\_\_\_ 143**

Monitor Nodes \_\_\_\_\_ 144

Monitor Ports \_\_\_\_\_ 146

Filtering pages displaying nodes \_\_\_\_\_ 149

Filtering using the Free Text Search field \_\_\_\_\_ 150

Filtering using the Smart Group Filtering drop-down menu \_\_\_\_\_ 151

Filtering using the Managed Device Filtering drop-down menu \_\_\_\_\_ 152

Monitor Jobs \_\_\_\_\_ 153

Enrol Nodes \_\_\_\_\_ 154

The Enrolled Nodes page \_\_\_\_\_ 156

Enrollment Bundles \_\_\_\_\_ 160

Assigning Subscriptions \_\_\_\_\_ 161

Creating an Enrollment bundle \_\_\_\_\_ 162

Structure of an Enrollment bundle \_\_\_\_\_ 166

Enrolling Nodes \_\_\_\_\_ 167

Enrollment via Node Web UI \_\_\_\_\_ 168

Enrollment via Lighthouse Web UI \_\_\_\_\_ 169

Enrolling nodes via OM, ACM, CM, and IM Web UI \_\_\_\_\_ 172

Enrollment via USB drive \_\_\_\_\_ 174

Backing up Nodes \_\_\_\_\_ 175





# LIGHTHOUSE

Central Management

Subscribing to Lighthouse	178
Adding a new subscription to Lighthouse	180
To add a subscription	181
Assigning subscriptions to nodes	183
View Subscriptions in the Lighthouse UI	184
To assign nodes to subscriptions	185
To assign a subscription	186
To change a node's subscription	187
Work with Nodes	188
Connect to the web-management interface of a node	189
Connecting to a node's serial ports	191
Find the Serial Ports	192
Access via HTML5 Web Terminal	196
Access a serial port via SSH	197
Example Console Gateway session	199
Selecting nodes using shell-based tools	200
Node organization and filtering	202
Filter Nodes	203
Creating Smart Groups	204
Editing a Smart Group	207
Creating Managed Device Filters	209
Editing an existing Managed Device Filter	212
Upgrade Nodes via the UI	214
Firmware Files	215
Upload a firmware file	216
Delete a firmware file	217
Create an upgrade task	218
Cancel an upgrade task	222
Copy a scheduled task	223

22.11 - REVISION 1	Document Revision History	8
22.11 - REVISION 1	Document Revision History	8

Delete an upgrade task _____	225
Retry an upgrade task _____	226
<b>Node Upgrade Runtime Behaviour _____</b>	<b>228</b>
Promoting a Secondary instance to Primary _____	229
Downgrading and Skipping Versions _____	230
Time Zones _____	231
Offline nodes _____	232
Node Connection interrupted _____	233
Unenrolling Nodes at Upgrade _____	234
Lighthouse Availability and Stability _____	235
 <b>Configuring Lighthouse _____</b>	 <b>236</b>
<b>Create Templates _____</b>	<b>237</b>
Creating new user and group templates _____	238
Modifying existing users and groups templates _____	241
Deleting users or groups from a template _____	242
Deleting users and groups templates _____	243
<b>Create Authentication Templates _____</b>	<b>244</b>
Creating new authentication templates _____	245
Modifying existing authentication templates _____	247
Deleting authentication templates _____	248
<b>Create Script Templates _____</b>	<b>249</b>
Creating new script templates _____	250
Modifying existing script templates _____	252
Deleting script templates _____	253
<b>Use Templates _____</b>	<b>254</b>
Apply Templates _____	255
Manually Activate NetOps modules via Template _____	257
<b>Configuration &amp; Technical support reports _____</b>	<b>259</b>

## **Managing Lighthouse Users \_\_\_\_\_ 262**

Role Description \_\_\_\_\_ 263

Work with Groups \_\_\_\_\_ 264

Creating New Groups and Roles \_\_\_\_\_ 265

    Create a new group \_\_\_\_\_ 266

    Available Roles: \_\_\_\_\_ 269

    Available Operations Permissions: \_\_\_\_\_ 270

Create a new Role \_\_\_\_\_ 273

Modifying existing groups \_\_\_\_\_ 276

Use an existing group as a template for a new group \_\_\_\_\_ 278

Work with Users \_\_\_\_\_ 279

    Creating new local users \_\_\_\_\_ 280

    Create New Local Users for Remote Authentication \_\_\_\_\_ 282

    Modify existing users \_\_\_\_\_ 284

    Expire user password \_\_\_\_\_ 286

    Setting Login Restrictions \_\_\_\_\_ 287

    Disabling a Lighthouse root user \_\_\_\_\_ 289

    Delete and Disable Users \_\_\_\_\_ 290

SAML Config for SSO \_\_\_\_\_ 291

Generic IdP Setup \_\_\_\_\_ 293

Generic IdP SAML Attribute \_\_\_\_\_ 296

Lighthouse Setup \_\_\_\_\_ 297

    Examples of Specific IdP Setups \_\_\_\_\_ 299

        Okta \_\_\_\_\_ 299

    Create an Application \_\_\_\_\_ 300

        Onelogin \_\_\_\_\_ 302

        Azure Active Directory \_\_\_\_\_ 305

Configure AUTH0 for IdP \_\_\_\_\_ 310

22.11 - REVISION 0	Create an Application (Enterprise applications) How-to	311	10
--------------------	--	-----	----

Configure AUTH0 Metadata for IdP _____	314
Configure Lighthouses for AUTH0 for IdP _____	315
Configure AUTH0 for IdP _____	316
Limitations of SAML Configuration _____	317
IdP metadata certificate expiry _____	318
Making Changes to User Permissions _____	319
SAML SSO Usergroups _____	320
SAML SSO Users _____	321
Configuring AAA _____	322
LDAP Configuration _____	323
Radius Configuration _____	326
TACACS+ configuration _____	328
Setting password policy _____	330
Password fields in Lighthouse _____	332
<b>Managing Automation _____</b>	<b>333</b>
Introduction _____	334
About NetOps _____	335
NetOps Platform Security _____	336
NetOps Module Management _____	337
Upgrade Modules from the UI _____	338
Upgrade Modules from the CLI _____	339
Update NetOps Modules Without Docker Access _____	340
Use a Pre-built Offline Installer (via GUI) _____	341
Use a Pre-built Offline Installer (via CLI) _____	342
Upgrade NetOps Modules _____	343
Activate a NetOps Module _____	346
Activate the NetOps Module on Lighthouse _____	347
'Always Activate' Mode on All Nodes (Automatic) _____	349
Activate NetOps on Selected Nodes (Automatic) _____	350

Activate the NetOps Module on Nodes (Manual) _____	351
Deactivate (remove) a NetOps Module _____	352
Automation Gateway _____	354
Differences Between IP Access and Automation Gateway _____	355
How to use Automation Gateway _____	356
PROCEDURE _____	357
Connect with REST/HTTP/HTTPS APIs _____	359
Automation Gateway Service Discovery _____	363
IP ACCESS _____	364
Connectivity _____	365
Enable IP Access in Lighthouse _____	368
Enable NetOps Automation _____	369
Activate the IP Access Module _____	370
Network Ports Used for IP Access _____	371
Generate a Certificate and Export Client Configuration _____	372
Connect the VPN Client _____	373
Password Authentication During VPN Connection _____	374
Advanced Options _____	375
Connecting to WAN Zone _____	376
Network Access Policies for Operations Manager _____	377
Understanding Access Policies _____	378
Setting up Network Access Policies _____	379
To set Access policies _____	380
Accessing Multiple VLANs or Ports _____	384
Group memberships _____	385
Firewall Zones _____	386
Multiple Layer 3 Network Connections _____	387
Troubleshooting IP Access _____	388
View the Docker logs _____	389
Using the Routing Table _____	391

<b>SECURE PROVISIONING</b>	<b>393</b>
Secure Provisioning Configuration Management	394
Stateless File Management	395
Stateful Device Management Gateway	396
How Secure Provisioning Works	397
Support for Secure Provisioning	399
Vendor Managed Devices Supported by Secure Provisioning	399
Local Network Services Provided by Nodes	401
Default Gateway	402
DNS Server	403
NTP Server	404
Syslog Server	405
Secure Provisioning Configuration	406
Device Resource Bundle	407
Node Inventory	408
Create Device Configuration	409
Activate the Secure Provisioning Module on Lighthouse	411
Install the Node	412
Configure a Per-node Module Activation Policy	413
Option A. Automatically Activate All Nodes Upon Enrollment	414
Option B. Automatically Activate Select Nodes Upon Enrollment	415
Option C. Manually Activate Nodes After Enrollment	416
Enroll the Node Into Lighthouse	417
Connect Target Device	418
PROCEDURE	419
UI-based Workflow	420
UI Workflow	421
Create Device Resource Bundle	422
Define Resource Distribution	425
Define a Static Node Inventory	426

Define a Dynamic Node Inventory _____	428
Push Resources _____	429
CLI-based Workflow _____	430
Create Configuration YAML _____	431
Define a Static Inventory: _____	434
Define a Dynamic Inventory: _____	435
How UI Fields Correspond to the YAML File (Example) _____	438
Upload Configuration and Resources _____	441
Option A. Secure Copy Method _____	442
Option B. git Method _____	443
Additional Resource Files and Device Type Files _____	444
Configure Device Resources via ZTP _____	446
Device Resource Bundle Matching _____	447
Resource Distribution _____	448
Baseline vs Final Device Configuration _____	449
Run a Script on a Newly Provisioned Device _____	450
Monitor the ZTP Progress of a Managed Device _____	452
WAN Gateway Services _____	453
Advanced Options _____	454
Using Variables in Configuration File Templates _____	455
Post-provisioning Scripts _____	458
Ordered Provisioning _____	460
Troubleshooting Secure Provisioning _____	462
Troubleshooting Commands _____	463
<b>Lighthouse CLI, Serial Port and REST API logging _____</b>	<b>464</b>
Using ogconfig-cli to enable logging _____	465
Add node and port to Lighthouse logs _____	466
Example logs _____	467
Enable logging _____	468

Disable logging	470
Syslog Export	471
<b>Command line tools</b>	<b>473</b>
Example node-command Output	474
node-info	475
Example node-info output	476
node-upgrade	477
An example node-upgrade run	479
Results and Error Messages in node-upgrade	480
cron	481
sysflash	483
<b>Glossary</b>	<b>484</b>



## About This User Guide

This user guide describes how to use Lighthouse and is current as of 2022.11

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

The current Lighthouse user guide, and other guides can always be found [here on the website](#)..

**Note:** OPERATIONS MANAGER support may be partial for earlier releases.

Partial support may currently involve:

**Mass node Enrollment using ZTP**

**Enrollment via USB drive**

All template types are supported.

## Lighthouse Overview

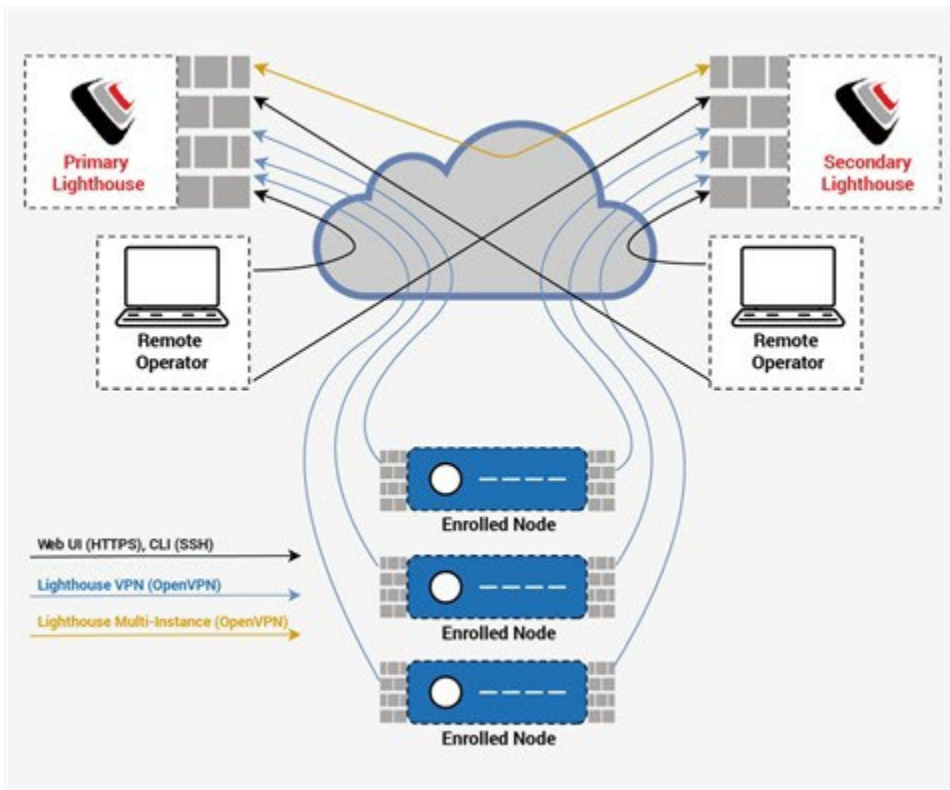
Lighthouse allows network engineers to centrally access, manage, and monitor a network of Opendgear console servers. Lighthouse is a virtual machine that can be hosted on various platforms like VMware or Hyper-V (on customer hardware) or on supported cloud providers (AWS or Azure).

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

## Lighthouse Architecture

Lighthouse functions as an API-driven platform with an HTML5 interface. It provides secure access to remote networks regardless of how they are connected or how users interact with the system. In combination with NetOps Console Servers, Lighthouse can push and manage Docker containers to each remote location to provide additional functionality and automation.

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 High Availability. If there is no secondary Lighthouse, the setup remains the same but without the secondary elements.

## The 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 Opendgear FTP site, and uploaded manually on Lighthouse.

To host Lighthouse, the VM needs to 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.

To dimension CPU and RAM resources, follow these guidelines:

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

For small deployments (Up to 500 nodes), allocate at a minimum:

22.11 - REVISION 1	Lighthouse Overview	19
--------------------	---------------------	----

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

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

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

For large deployments (more than 1000), allocate allocate at the minimum:

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

**Note:** In general it is recommended that you add more memory than you assume you need.

For large deployments, please 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.

## Using Multiple Lighthouses

Lighthouse offers a form of high availability with the Multi Instance feature. The Multi Instance feature allows you to set up secondary/dependent instances of Lighthouse that automatically receive updates from the primary Lighthouse instance, and maintains connections to all of its remote nodes.

It will also replicate various values from the primary Lighthouse to ensure in the event of a failover, all information is up to date.

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

The multiple instance feature has the following limitations:

- Up to ten secondary instances can be enrolled.
- Secondary Lighthouse instances are read-only. We recommended that you preconfigure instance specific settings such as hostname, external endpoints, and time zone on a secondary instance before adding it to the primary in a normal way through UI.
- Dependent Lighthouse instances must have zero nodes enrolled before being enrolled to the primary Lighthouse.
- Removing a dependent Lighthouse instance will initiate a factory reset of the removed Lighthouse.
- If external network addresses on the primary or secondary Lighthouses are updated after a secondary Lighthouse has been enrolled, it may break replication.

- Only Opendgear nodes with a version that supports multiple instance will connect to the secondary instance. Nodes that don't support multiple instance will behave normally on the primary.

The secondary instance user interfaces are read-only.



## EULA and GPL

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

## Installing Lighthouse

To install Lighthouse you require:

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

## Lighthouse Virtual Machines Availability

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 upgrade file for your upgrade.

- **lighthouse-22.11.0-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-22.11.0-vmx.zip** - A VMware configuration file inside a PKZip archive. This is for use with virtual machine managers from VMware.
- **lighthouse-22.11.0-raw.hdd.tar** - A raw (.hdd) file. This file has been compressed with tar and is for use with hosting services such as ElasticHosts.
- **lighthouse-22.11.0.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-22.11.0-hyperv.zip** - A Hyper-V configuration file with Powershell script inside a PKZip archive. This is for use in Microsoft Hyper-V deployment.
- **lighthouse-22.11.0.azure.zip** - A Microsoft Azure file, for deploying on Azure.
- **lighthouse-aws-bootstrap.sh**, and the **lighthouse-22.11.0.aws.raw.tar** image - A shell script for deploying on AWS.
- **lighthouse-22.11.0.lh\_upg** - An upgrade file.
- **lighthouse-22.11.0.aws.lh\_upg** - Upgrade on AWS.
- **lighthouse-22.11.0.azure.zip** - Upgrade on Azure.

There are also SHASUMS files to verify the downloaded files.

22.11 - REVISION 1	Installing Lighthouse	26
--------------------	-----------------------	----

**Note:** The latest format for the files is as follows:

**lighthouse-<year>.<month>.<release/patchnumber>**

## Installing Lighthouse VM on VMware

This section describes how to install Lighthouse VMs on VMware hosts.

## **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. The simplest way is to use the **Start Menu** shortcut added during installation.

**Start > All Programs > VMware > VMware vSphere Client**

The VMware vSphere Client opens a login window.



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



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

*Connecting*

*Loading inventory*

*Loading main form*

*Displaying main form*

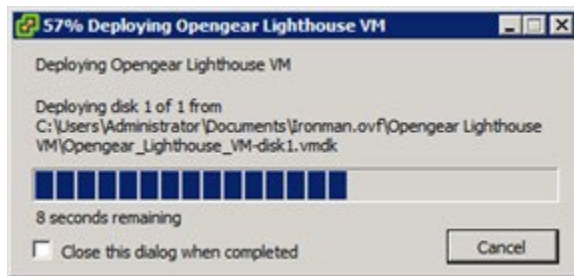
The vSphere main form window opens.

## Import the Lighthouse VM Open Volume Format (.ovf) 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**.
9. The **Deploying Opengear Lighthouse VM** progress dialog displays.



10. Once deployment has finished the **Deployment Completed Successfully** alert displays. 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 vSphere Client's inventory list, then do one of the following:

- Select **Inventory > Virtual Machine > Open Console** or
- Right-click and select **Open Console** from the 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-22.11.0-vmx.zip archive to **C:\Users\%USERNAME%\Virtual Machines\**.
2. Right-click the archive and select **Extract All** from the contextual menu.
3. A **Select a Destination and Extract Files** dialog opens. By default, the location is the same folder as the archive is in:  
**C:\Users\%USERNAME%\Virtual Machines\**. Leave this as the destination folder.
4. Uncheck the **Show Extracted Files When Complete** checkbox and then click **Extract**.
5. A folder called **lighthouse** is created inside **C:\Users\%USERNAME%\Virtual Machines\**.

## Import the Opendgear 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**.
5. The Lighthouse virtual machine is added to the VMware Workstation 12 Player virtual machines list.
6. With **Opendgear 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 needed.



## 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-22.11.0.zip** archive to **C:\Users\%USERNAME%\Virtual Machines\**.
2. Right-click the **lighthouse-22.11.0-vmx.zip** archive and select **Extract All** from the contextual menu.
3. A **Select a Destination and Extract Files** dialog opens. The location is the same folder as the PKZip archive is in: **C:\Users\%USERNAME%\Virtual Machines\**. Leave this as the destination folder.
4. Uncheck the **Show Extracted Files When Complete** checkbox and then click **Extract**.
5. A folder called lighthouse is created inside **C:\Users\%USERNAME%\Virtual Machines\**.

### Step 2. Import Lighthouse VM File:

1. Click **Open a Virtual Machine**.
2. Navigate to **C:\Users\%USERNAME%\Virtual Machines\lighthouse\**.
3. VMware Workstation Pro points to **Libraries > Documents** and this library includes **C:\Users\%USERNAME%\My Documents\**. Double-click **Virtual Machines** and then double-click **Lighthouse**.
4. If only one file, Lighthouse, 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**.

5. The Lighthouse virtual machine is added to the VMware Workstation Pro virtual machines list.
6. With the **Opengear Lighthouse VM** selected in the **My Computer** listing and the subsequent **Opengear Lighthouse VM** tab open, click **Power** on this virtual machine to boot Lighthouse.

## Installing 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 running on Windows 10/Windows Server 2016

This procedure assumes Hyper-V is already installed on a Windows 10/Windows Server 2016 host machine and the required Zip archive, **lighthouse-22.11.0-hyperv.zip** is in **C:\Users\%USERNAME%\Downloads**.

1. Unzip **lighthouse-22.11.0-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 Opendgear 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 archive **lighthouse-22.11.0-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-22.11.0-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-22.11.0-ovf.zip** is in **C:\Users\%USERNAME%\Downloads**.

1. Unzip **lighthouse-ovf**. It may appear as **lighthouse-22.11.0-ovf.zip** depending on the Windows Explorer preference settings).
2. Right-click the **lighthouse-ovf** archive and select **Extract** all from the context menu.
3. The **Select a Destination** and **Extract Files** dialog opens. The destination is **C:\Users\%USERNAME%\Downloads\Lighthouse-ovf**.
4. Uncheck the **Show extracted files when complete** checkbox and edit the destination by removing **Lighthouse-ovf** from the path.
5. Click **Extract**.
6. A folder called **lighthouse-ovf** is created inside **C:\Users\%USERNAME%\Downloads\**.
7. Launch VirtualBox.
8. The Oracle VM VirtualBox Manager window displays.
9. Choose **File > Import Appliance**.
10. The **Appliance to Import** dialog opens.
11. Click **Expert Mode**.



12. The **Appliance to import** dialog changes from **Guided Mode** to **Expert Mode**.
13. Click the icon of a folder with an upward pointing arrow superimposed. This icon is to the far right of the **Appliance to import** field.
14. The **Open File** dialog displays with **C:\Users\%USERNAME%\Documents** as the current folder.
15. Navigate to  
**C:\Users\%USERNAME%\Downloads\Lighthouse.ovf\Opengear Lighthouse VM\**.
16. Select the file **Opengear Lighthouse VM** and click **Open**.
17. Double-click the text **vm** in the **Name** row and **Configuration** column to make it editable.
18. Type **Opengear Lighthouse VM** and press **Enter**.
19. Click **Import**.
20. A new virtual machine, called **Opengear Lighthouse VM** is added to the list of virtual machines available to **Virtual Box**.
21. Select **Opengear Lighthouse VM** from the list.
22. Select **Machine > Settings** or click the **Settings** icon in the **VirtualBox Manager** toolbar or press **Control+S**.
23. The **Opengear Lighthouse VM - Settings** dialog displays.
24. Click the **System** option in the list of options running down the left-hand side of the dialog.
25. The dialog shows the **System** options available as three tabs: **Motherboard**, **Processor**, and **Acceleration**. Depending on the underlying hardware platform, Acceleration may be greyed-out and unavailable. The Motherboard

tab is preselected.

26. In the **Motherboard** tab, select the **Hardware Clock in UTC Time** checkbox.
27. Click **OK** or press **Return**
28. Select **Opengear Lighthouse VM** from the list and click **Start** in the Oracle VM VirtualBox Manager toolbar to boot Lighthouse. Double-clicking **Opengear Lighthouse VM** in the list also boots Lighthouse.

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

## VirtualBox on macOS as host

VirtualBox should already be installed on the host macOS machine and the required PKZip archive, **lighthouse-22.11.0-ovf.zip** is in **~/Downloads**.

1. Unzip **lighthouse-22.11.0-ovf.zip**.

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

```

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

```

2. Launch Virtual Box. The Oracle **VM VirtualBox Manager** window displays.
3. Select **File > Import Appliance** or press **Command+I**.
4. The **Appliance to Import** dialog sheet slides down from the Oracle VM VirtualBox Manager toolbar.
5. Click **Expert Mode**.  
The **Appliance to Import** dialog sheet changes from **Guided Mode** to **Expert Mode**.
6. Click the icon of a folder with an upward pointing arrow superimposed. This icon is to the far-right of the **Appliance to Import** field.
7. The **Open File** dialog sheet slides down from the **Oracle VM VirtualBox Manager** toolbar. This sheet opens with **~/Documents** as the current folder.
8. Navigate to **~/Downloads/Lighthouse.ovf/Opengear Lighthouse VM/**.
9. Select **Opengear Lighthouse VM** and click **Open**. (Depending on the **Finder Preferences** settings, the file may present as **Opengear Lighthouse VM.ovf**.)

10. Double-click the text **vm** in the **Name** row and **Configuration** column to make it editable.
11. Type **Opengear Lighthouse VM** and press **Return**.
12. Click **Import**.  
A new virtual machine, called Opengear Lighthouse VM is added to the list of virtual machines.
13. Select **Opengear Lighthouse VM** from the list.
14. Choose **Machine > Settings**. Or click the **Settings** icon in the **VirtualBox Manager** toolbar. The **Opengear Lighthouse VM Settings** dialog displays.
15. 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.
16. In the **Motherboard** tab, select the **Hardware Clock in UTC Time** checkbox.
17. Click **OK** or press **Return**.
18. **Select Opengear Lighthouse VM** from the list and click **Start** in the **Oracle VM VirtualBox Manager** toolbar to boot Lighthouse. Double-clicking **Opengear Lighthouse VM** in the list also boots Lighthouse.

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

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

## VirtualBox on Ubuntu as Host

Before beginning the procedure, make certain that VirtualBox and all required support files are installed on the host machine and the PKZip archive, **lighthouse-22.11.0-ovf.zip** is in **~/Downloads**.

1. Unzip **lighthouse-22.11.0-ovf.zip**. This creates a folder, **Lighthouse-ovf** in **~/Downloads** that contains the following files and folders:

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

2. Launch **Virtual Box**.
3. The **Oracle VM VirtualBox Manager** window displays.
4. Choose **File > Import Appliance**.
5. The **Appliance** to import dialog opens.
6. Click **Expert Mode**.
7. The **Appliance to Import** dialog changes from **Guided Mode** to **Expert Mode**.
8. Click the icon of a folder with an upward pointing arrow superimposed. This icon is positioned to the right of the **Appliance to Import** field.
9. A file-navigation dialog, **Choose a Virtual Appliance to Import**, opens with **~/Documents** as the current folder.
10. Navigate to **~/Downloads/Lighthouse.ovf/Opengear Lighthouse VM/**.
11. Select **Opengear Lighthouse VM.ovf** and click **Open**.

12. Double-click the text **vm** in the **Name** row and **Configuration** column to make it editable.
13. Type **Opengear Lighthouse VM** and press **Return**.
14. Click **Import**.  
A new virtual machine, called **Opengear Lighthouse VM** is added to the list of virtual machines available to Virtual Box.
15. 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 beginning, make certain that VirtualBox and all required support files are already installed on the host machine and the PKZip archive, **lighthouse-22.11.0-ovf.zip** is in **~/Downloads**.

1. Unzip **lighthouse-22.11.0-ovf.zip**. This creates a folder **Lighthouse.ovf** in **~/Downloads** that contains the following files and folders:

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

2. Launch Virtual Box.

The **Oracle VM VirtualBox Manager** window 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 **Return**.



10. Click **Import**.

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

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

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

## Installing Lighthouse VM on Linux Hosts

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

- Ubuntu
- Fedora Workstation
- RHEL

## Virtual Machine Manager (KVM) on Ubuntu as host

Virtual Machine Manager and all required support files should be installed on the host machine and the the .tar archive, **lighthouse-22.11.0-raw.hdd.tar** is in **~/Downloads**.

1. Expand **lighthouse-22.11.0-raw.hdd.tar**. This extracts **lighthouse-22.11.0-raw.hdd** in **~/Downloads**.
2. Launch **Virtual Machine Manager**.
3. Click **New** at the top left of the Virtual Machine Manager window (or choose **File > New Virtual Machine**). The **Source Selection** window opens.
4. Click **Select** a file. A **Select a Device or ISO File** dialog slides into view.
5. Navigate to **~/Downloads/**.
6. Select the file **lighthouse-22.11.0-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.
7. Click **Create** in the top right corner of the **Review** window.
8. A new virtual machine instance, **Opengear\_Lighthouse\_VM-disk1**, is created and presented in the **Boxes** window.
9. To rename the virtual machine instance, right-click on the machine instance and choose **Properties** from the contextual menu that displays. Click anywhere in the **Name** field to select and edit the name. Click the **Close** box to save the changes.

## Boxes on Fedora Workstation as host

Boxes and all required support files should be installed on the host machine and **lighthouse-22.11.0-ovf.zip** is in **~/Downloads**.

1. Unzip **lighthouse-22.11.0-ovf.zip**. This creates a folder **Lighthouse.ovf** in **~/Downloads** that contains the following files and folders:

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

2. Launch **Boxes**.
3. Click **New** in the Boxes window title bar. The **Source Selection** window opens.
4. Click **Select a File**. A **Select a Device or ISO File** dialog opens.
5. Navigate to **~/Downloads/Lighthouse.ovf/Opengear Lighthouse VM/**.
6. Select the file **Opengear\_Lighthouse\_VM-disk1.vmdk** and click **Open** in the top right-hand corner of the dialog. A **Review** window opens providing basic information about the virtual machine (or 'box', as Boxes calls them) to be created.
7. Click **Create** in the top right corner of the **Review** window.
8. A new virtual machine instance, **Opengear\_Lighthouse\_VM-disk1** is created and presented in the **Boxes** window.

9. To rename the virtual machine instance, right-click on the machine instance and choose **Properties** from the contextual menu that 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-22.11.0-ovf.zip** is in **~/Downloads**.

To install Lighthouse on CentOS:

1. Unzip **lighthouse-22.11.0-ovf.zip**.

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

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

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

## Installing in the Cloud

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

- . Azure
- . AWS.

## Installing 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 **Storage Explorer** and create a new blob container.
5. Upload the **lighthouse.vhd** image provided with the **lighthouse-22.11.0.azure.zip** file.
6. Go to the newly created image and click **Create VM**. Ensure the selected image is correct.
7. Choose the desired virtual machine instance size.
8. 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.

9. To login to the Lighthouse UI, login via SSH with key authentication and configure a password using the `ogpasswd` utility, for example,

```
sudo ogpasswd -u root -p NewPassword
```

10. Select the inbound ports enabled for the Lighthouse instance (SSH, HTTPS).
11. Navigate to the next page of configuration (Disks) and select the desired storage option for the boot disk.



12. Go to the **Review** page.
13. 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 will need to add the following firewall rules in the Microsoft Azure virtual machine control panel:
  - a. Go to the virtual machine configuration and select **Networking**.
  - b. Add a rule to allow UDP connections from any source to port 1194 on the instance's internal network address (10.0.0.x).
  - c. Add a rule to allow UDP connections from any source to port 1195 on the instance's internal network address (10.0.0.x).
  - d. HTTPS and SSH should already be allowed from the initial setup. If not, add them.
16. Confirm that the Azure instance public IP address has been added to external endpoints in **Settings > System > Administration**.

## Installing in Amazon Web Services(AWS)

To use Lighthouse with AWS, you will first need to

1. Create an account on AWS with an IAM user, and a key pair to log into your instance.
2. Create an AWS EC2 Linux instance, with 50GB or more of disk space.
3. Create an Amazon Machine Image.

Amazon offers a walkthrough of the necessary steps to set up on AWS. [Click here to read the instructions on Amazon's website.](#)

**Note:** Be sure you set up correctly on AWS before you attempt to create you Lighthouse VM on AWS.

## Create a Lighthouse AMI Image

Lighthouse does not have a public Amazon Machine Image (AMI). You will need to launch a Linux build-box instance in order to create a Lighthouse AMI.

**Note:** This is a one-time procedure, as once Lighthouse is running, you can upgrade it from the Lighthouse Web GUI.

Locate the **lighthouse-aws-bootstrap.sh script** and **lighthouse-22.11.0.aws.raw.tar** image at:

<http://ftp.opengear.com/download/lighthouse/current/aws/>

## About the Lighthouse Bootstrap script

**lighthouse-aws-bootstrap.sh** will create an AMI from a Lighthouse image and has the following options:

- f `FILENAME` Use the specified local file to create the image
- r `URI` Download the image file from the specified URI
- d `DEVICE` Attach temporary disk images to the specified device (eg, xvde)
- n `NAME` The name to use for generated images (default: Lighthouse)
- h Display help message

## After creating the AWS EC2 Linux instance:

1. Connect via SSH to your instance on AWS using the username `ec2-user` and the private key you created previously.
2. Use `sudo -i` to become root.
3. Run the following command:  

```
aws --output json configure
```

4. Provide an access key with administrative privileges.

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

5. Use the `wget` command to download the aws-bootstrap script, **lighthouse-aws-bootstrap.sh** onto the AWS EC2 Linux instance.
6. Using `wget` download the Lighthouse image **lighthouse-22.11.0.aws.raw.tar** to the AWS EC2 Linux instance.
7. Untar the file, **lighthouse-22.11.0.aws.raw.tar**
8. From AWS, run the **lighthouse-aws-bootstrap.sh** script with the appropriate parameters to tell it where to find the untarred Lighthouse image on the instance. For example,

```
./lighthouse-aws-bootstrap.sh -f FILENAME.lh_hdd -n Lighthouse
```

9. A Lighthouse AMI is created.

You can use the AMI to create a Lighthouse instance with any hardware configuration you require. After the AMI has been created, this EC2 instance should be terminated, as it will incur costs to keep it running.

## Set a password for the root user on Lighthouse:

Open the **Configure instance details** page of the AMI launch process.

1. Under the **Advanced Details** section, add a root password using the userdata field in the *format password=Whatever123*. If you do not, you will have to log in via SSH to set it.
2. If you are logged into Lighthouse via SSH keys, you will need to set root password to login via GUI. Use the "ogpasswd" utility to do this. We suggest using alpha numeric character passwords with this utility and then set a stronger/complex password once you are logged in via the GUI.

```
ogpasswd -u root -p MySecretPassword
```

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

**Note:** Currently AWS support is limited to: All standard Lighthouse operations

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

At this time we do not support:

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

**Note:** The root password must be specified in the **Advanced Details**.

## Adding Disk Space to Lighthouse

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

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 the primary instance.

## Adding 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 5GB volume in the same availability zone as your LH instance.
2. Once the volume is created, select it and click the **Actions** button and select **Attach Volume**.
3. Enter the LH instance ID for the instance field and `/dev/xvdb` (or `/dev/xvdd`, `/dev/xvde` and so on) as the device and click **Attach**.

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



## Adding 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 you should make sure that the new disk is specified last, otherwise your disk will appear out of order when you boot the Lighthouse.

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

## Adding 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>

## Adding a New Disk - Hyper-V

1. Launch the LVM Lighthouse instance as per our guidelines or your own deployment. To add a volume to the instance:
2. Shutdown your Hyper-V Lighthouse instance.
3. Open your Hyper-V manager.
4. Navigate to the VM list and locate your Lighthouse VM.
5. Right click on the instance and click Settings.
6. Click on the SCSI controller.
7. Select Hard drive on the right and click Add.
8. Select Virtual hard disk and click New.
9. Follow the prompts and select the options that best suit your needs and environment.
10. Once you've created the disk, click Apply in the VM settings window.
11. Restart the Lighthouse.

## Adding 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.

## 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 (assuming you are now using `/dev/sdb1`, note that `/dev/xvdb1` will now be mapped to `/dev/sdb1` so make sure you use `sdb1`).  

```
pvccreate /dev/sdb1
```
9. Extend the volume group with the new physical volume.  

```
vgextend lhvg /dev/sdb1
```
10. Assuming the new disk gives you at least 2GB of extra space, expand the `lh_data` logical volume.  

```
lvextend -L +2G /dev/mapper/lhvg-lh_data
```
11. Update the file system of the `lh_data` disk to use the extra space.  

```
resize2fs /dev/mapper/lhvg-lh_data
```
12. When you log into the shell, the disk should no longer be listed as "unused".

## 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:  
2021.Q4.0
```

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

**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.

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:~#
```

## Setting up Lighthouse

This section describes the initial setup and loading of the Lighthouse application to effectively monitor and manage your network as follows:

- Loading Lighthouse
- Setting up Networking Requirements
- Setting up Multiple Instances
- Upgrading Lighthouse
- Shutting down or Restarting Lighthouse



## Loading Lighthouse

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

## To load Lighthouse

1. Open a new browser window or tab, then enter one of the following:

```
https://192.168.0.1/
```

```
https://[DHCP-supplied address]/
```

or an IPv6 address, for example:

```
https://[fe80::a00:27ff:fe39:daa3/64].
```

2. Press **Return**. The Lighthouse Login page loads. 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 **SETTINGS > SERVICES >HTTPS Certificate** to remove the warning message.

## Lighthouse IP addresses

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

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

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
```

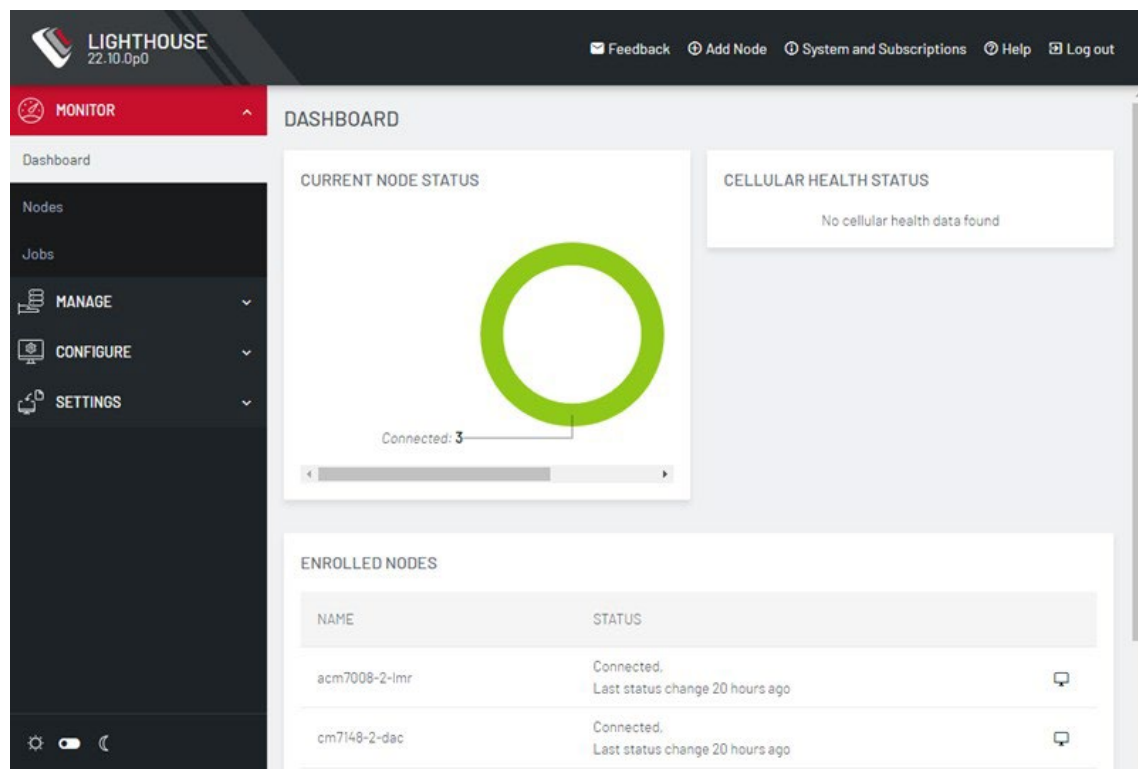
**Tip:** Type `ogset<tab>` and tab completion will give you the full command.

## Logging into Lighthouse

To login to Lighthouse:

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

When you log in, the standard Lighthouse menus, options and panes display:



- The primary menu options - MONITOR, MANAGE, CONFIGURE, and SETTINGS

- A light/dark mode toggle on the bottom left of the interface. This control allows you to modify the appearance of the dashboard for low light situations.
- System menu options **Add Node**, **Help**, **System & Subscriptions**, and **Log Out** on the header.
- Various panes - some column headings have Arrow Icons next to them. Click to toggle between ascending and descending order.

**Note:** The elements that appear on the **Dashboard** page depend on the privileges granted to the currently logged in user. In this guide, screenshots represent what the root user sees.

For root users, the Dashboard displays the following panes; **Enrolled Nodes**, **Cellular Health Status**, and **Current Node Status**.

Click **Cellular Health** to go to **MANAGE > NODES > Node Web UI** page where you can view the **Cellular Health** column with information on each node.

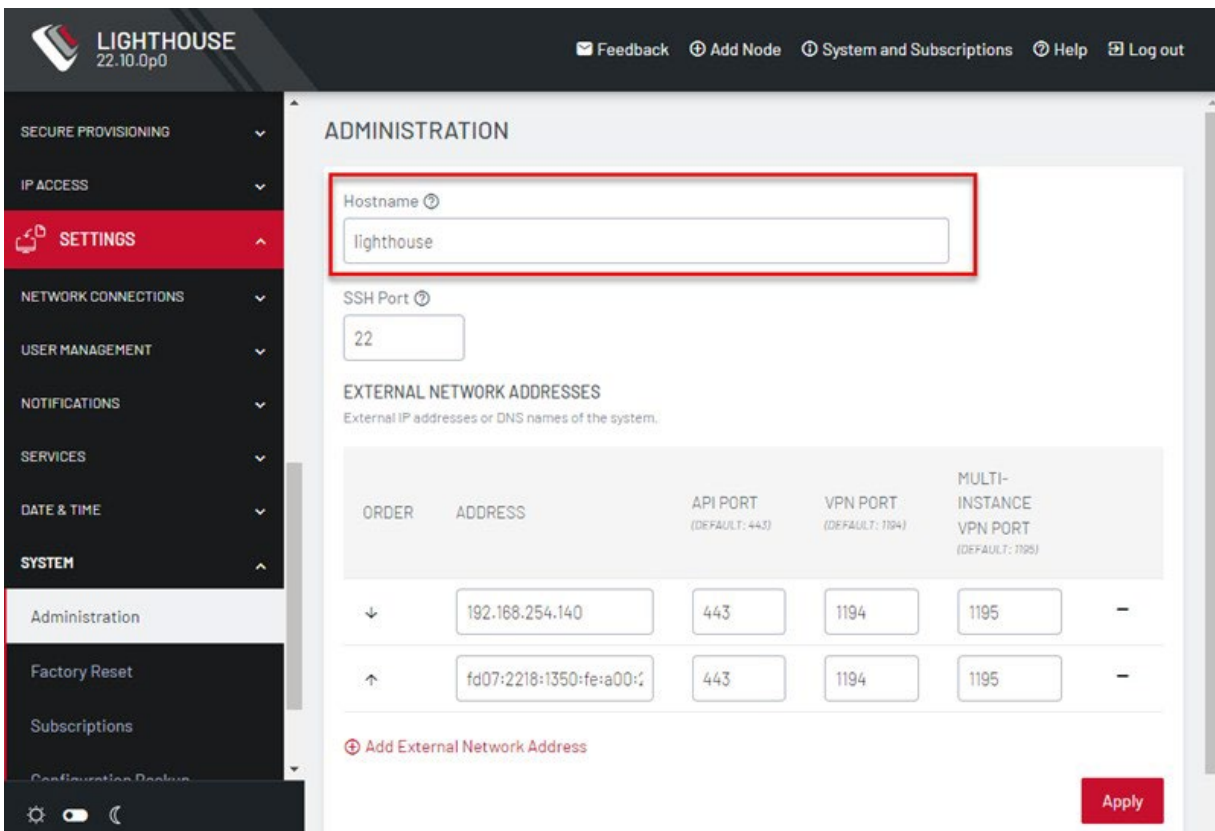
**Note:** The Cellular Health only displays if Cellular Health reporting is enabled.

Click **Current Node Status** to go to the appropriate page, that is, **Connected**, **Disconnected**, or **Pending** Node pages.

## Setting the Lighthouse hostname

To set the hostname for a running Lighthouse instance:

1. Select **SETTINGS > SYSTEM > Administration**.
2. Edit the **Hostname** field as desired. Hostnames must follow the naming conventions
  - ASCII alphanumeric characters plus - and .
  - Maximum 64 characters



**LIGHTHOUSE** 22.10.0p0

Feedback Add Node System and Subscriptions Help Log out

**ADMINISTRATION**

Hostname ⓘ  
lighthouse

SSH Port ⓘ  
22

**EXTERNAL NETWORK ADDRESSES**  
External IP addresses or DNS names of the system.

ORDER	ADDRESS	API PORT (DEFAULT: 443)	VPN PORT (DEFAULT: 1194)	MULTI- INSTANCE VPN PORT (DEFAULT: 1195)	
↓	192.168.254.140	443	1194	1195	-
↑	fd07:2218:1350:fe:a00::	443	1194	1195	-

+ Add External Network Address

Apply

3. Click **Apply**.

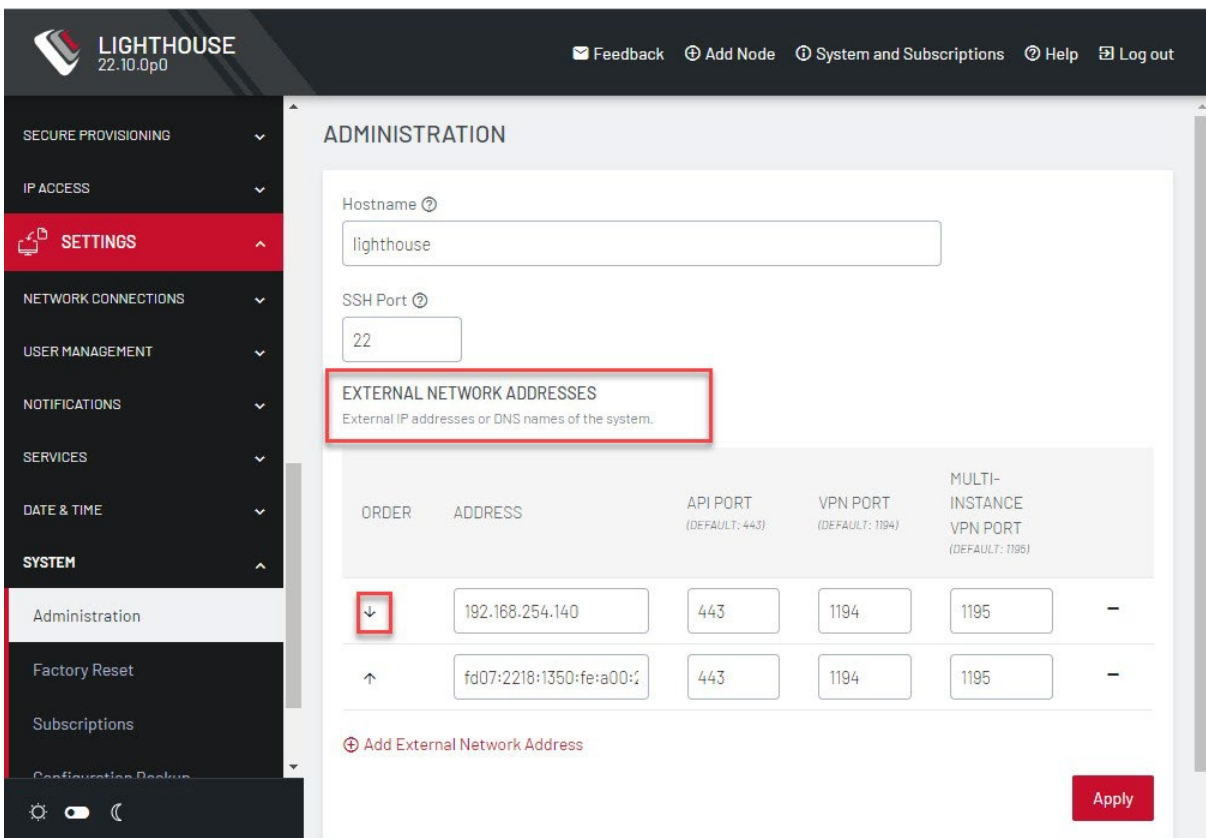
## Adding external IP addresses manually

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. For more information on the network ports used by Lighthouse, see [this link on the network architecture](#).

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

To add a single external address:

1. Select **SETTINGS > SYSTEM > Administration**.



The screenshot shows the Lighthouse Administration interface. The left sidebar contains a menu with items like SECURE PROVISIONING, IP ACCESS, SETTINGS (highlighted), NETWORK CONNECTIONS, USER MANAGEMENT, NOTIFICATIONS, SERVICES, DATE & TIME, and SYSTEM. The main content area is titled 'ADMINISTRATION' and contains fields for Hostname (lighthouse) and SSH Port (22). Below these is a section titled 'EXTERNAL NETWORK ADDRESSES' with the subtitle 'External IP addresses or DNS names of the system.' This section contains a table with columns: ORDER, ADDRESS, API PORT (DEFAULT: 443), VPN PORT (DEFAULT: 1194), and MULTI-INSTANCE VPN PORT (DEFAULT: 1195). The table has two rows: the first row has a down arrow in the ORDER column, the address 192.168.254.140, API PORT 443, VPN PORT 1194, and MULTI-INSTANCE VPN PORT 1195; the second row has an up arrow in the ORDER column, the address fd07:2218:1350:fe:a00::1, API PORT 443, VPN PORT 1194, and MULTI-INSTANCE VPN PORT 1195. Below the table is a link to 'Add External Network Address' and an 'Apply' button.

ORDER	ADDRESS	API PORT (DEFAULT: 443)	VPN PORT (DEFAULT: 1194)	MULTI- INSTANCE VPN PORT (DEFAULT: 1195)
↓	192.168.254.140	443	1194	1195
↑	fd07:2218:1350:fe:a00::1	443	1194	1195

[+ Add External Network Address](#) Apply



2. In the **Address** field of the **External Network Addresses** section, enter an IP address or DNS name.
3. Change the **API Port**, **VPN Port**, or **Multi-Instance VPN Port** if the ports used on the entered IP address are different from the default settings.
4. Click **Apply**.

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

1. Click the **+** button. A second row displays in the **External Network Addresses** section.
2. In the **Address** field, enter an IP address.
3. Change the **API Port**, **VPN Port**, or **Multi-Instance VPN Port** if the ports used on the entered IP address are different from the default settings.
4. Click **Apply**.

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

1. Click the **up** and **down** arrows in the **Order** column to change the order in which the IP addresses are listed.
2. Click **Apply**.

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

The external IP addresses are sent to a remote node during Enrollment in the order configured on the **SETTINGS > System > Administration** page.

## The Lighthouse SSL Certificate

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

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

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

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

## Set the Lighthouse Internal Clock

Lighthouse and Node system times need to 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 date and time.

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

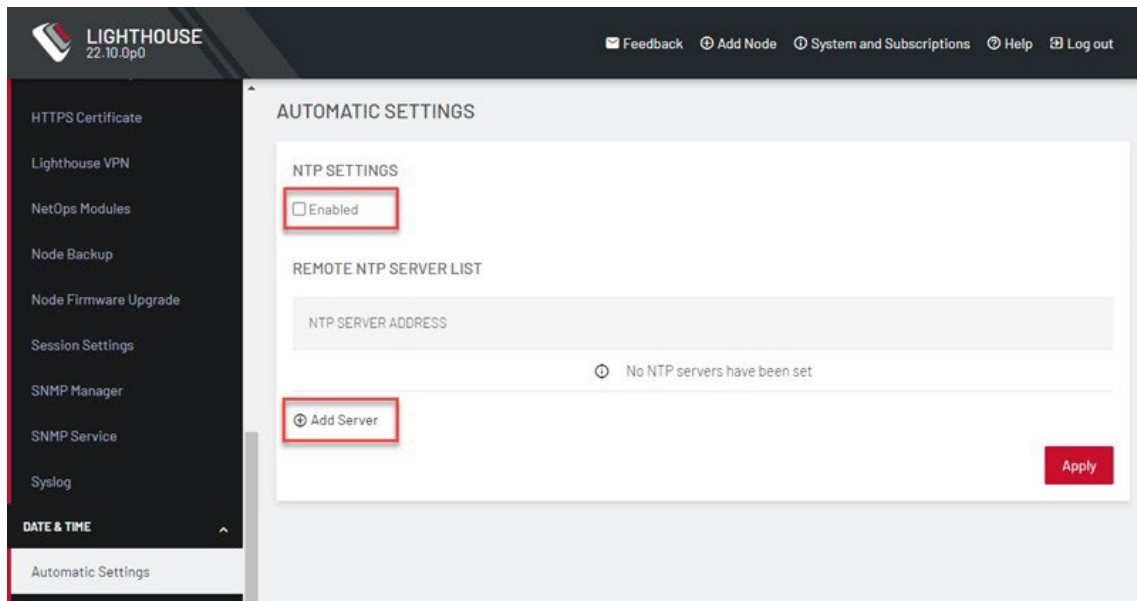
To set the time zone:

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

### Set time automatically

It is highly recommended that you use an NTP Server to automatically manage date and time:

1. Select **SETTINGS > DATE & TIME > Automatic Settings**.



2. Click the **Enabled** checkbox.
3. Click on **+ Add Server**.
4. Enter a working NTP Server address in the **NTP Server Address** field.
5. Click **Apply**.

## Setting up Networking Requirements

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

## Examine or Modify the Lighthouse Sessions

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

- **Web Session Timeout:** This value can be set from 1 to 1440 minutes.
- **CLI Session Timeout:** This value can be set from 1 to 1440 minutes or set it to 0 to disable the timeout. Changes take effect the next time a user logs in via the CLI.
- **Enable additional Enrollment-only REST API port:** This port defaults to 8443. 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.

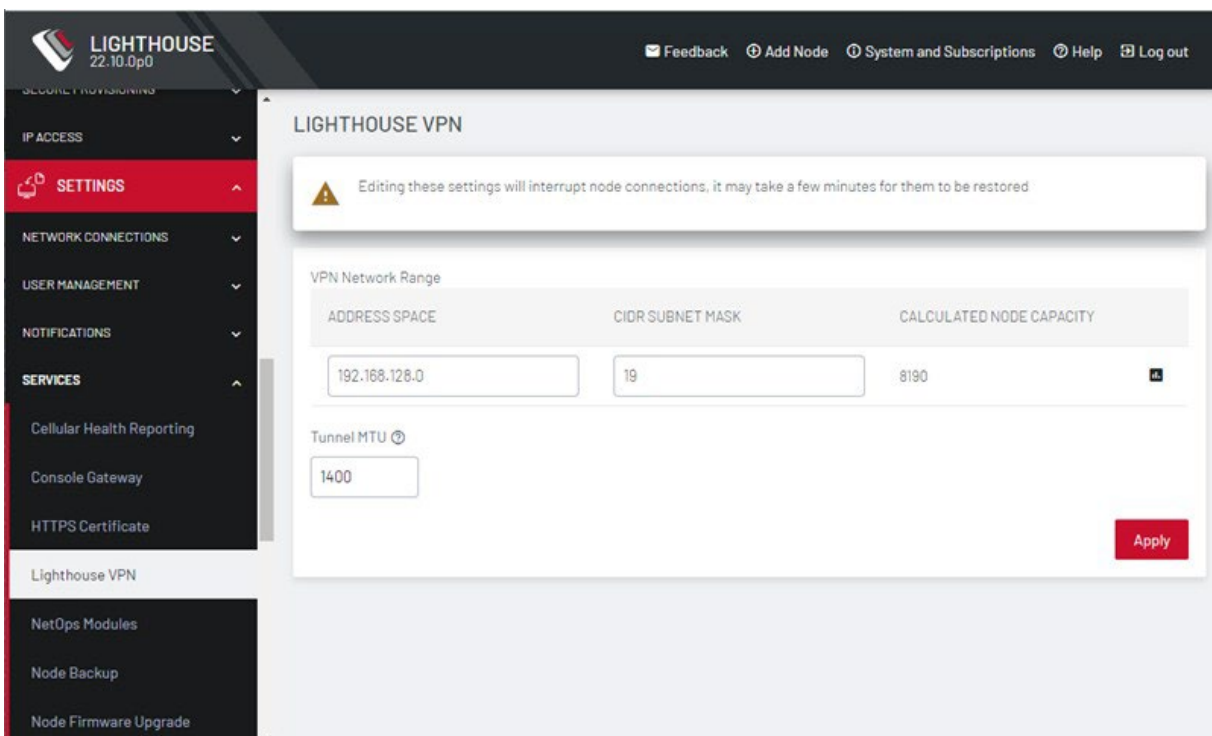
## Examine or Change the MTU of the Lighthouse VPN Tunnel

The **maximum transmission unit** (MTU setting can be configured for traffic that is travelling through the Lighthouse VPN in an attempt to solve MTU path discovery problems.

The maximum transmission unit (MTU) is the size, in bytes, of the largest packet supported by a network layer protocol, including both headers and data.

To modify the MTU of the Lighthouse VPN tunnel select **SETTINGS > SERVICES > Lighthouse VPN**.

Allowed values are between 1280 and 1500.



The screenshot shows the Lighthouse Central Management interface. The left sidebar contains a menu with the following items: SECURITY PROVISIONING, IP ACCESS, **SETTINGS** (highlighted in red), NETWORK CONNECTIONS, USER MANAGEMENT, NOTIFICATIONS, **SERVICES** (expanded), Cellular Health Reporting, Console Gateway, HTTPS Certificate, Lighthouse VPN (selected), NetOps Modules, Node Backup, and Node Firmware Upgrade. The main content area is titled "LIGHTHOUSE VPN". At the top of this area is a warning message: "Editing these settings will interrupt node connections, it may take a few minutes for them to be restored". Below this is a section for "VPN Network Range" with three columns: ADDRESS SPACE, CIDR SUBNET MASK, and CALCULATED NODE CAPACITY. The ADDRESS SPACE field contains "192.168.128.0", the CIDR SUBNET MASK field contains "19", and the CALCULATED NODE CAPACITY field contains "8190". Below this is a "Tunnel MTU" field with a value of "1400". An "Apply" button is located at the bottom right of the settings area.

## Network Connections

To see the network connections available to Lighthouse:

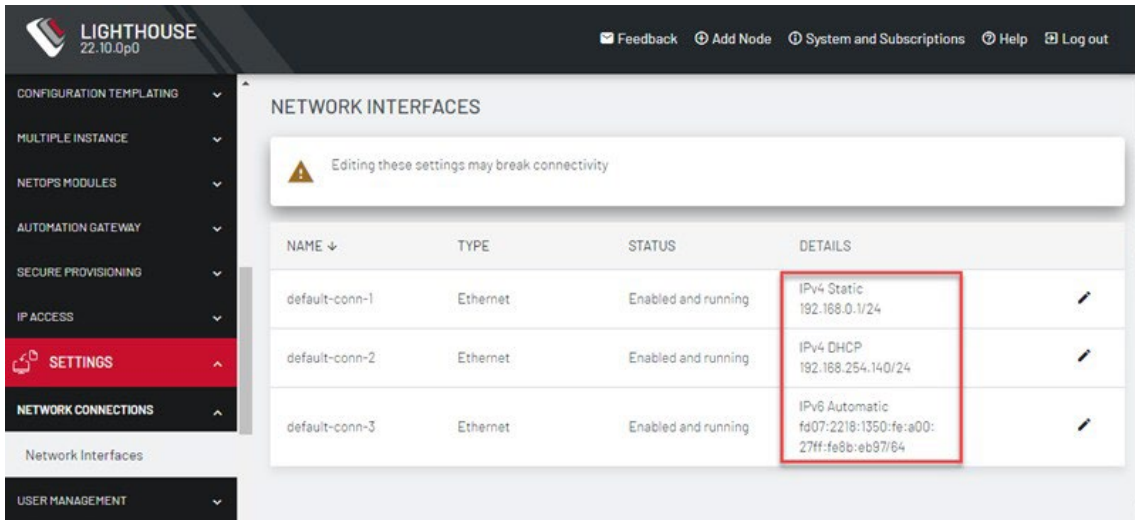
1. Select **SETTINGS > NETWORK CONNECTIONS > Network Interfaces**.
2. The three connections display:
  - a. Static interfaces
  - b. DHCP IPv4 interfaces and by default,
  - c. An automatic IPv6 connection.

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



## Edit a network interface

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



**LIGHTHOUSE** 22.10.0p0

Feedback Add Node System and Subscriptions Help Log out

CONFIGURATION TEMPLATING  
 MULTIPLE INSTANCE  
 NETOPS MODULES  
 AUTOMATION GATEWAY  
 SECURE PROVISIONING  
 IP ACCESS  
**SETTINGS**  
 NETWORK CONNECTIONS  
 Network Interfaces  
 USER MANAGEMENT

### NETWORK INTERFACES

⚠ Editing these settings may break connectivity

NAME ↓	TYPE	STATUS	DETAILS
default-conn-1	Ethernet	Enabled and running	IPv4 Static 192.168.0.1/24
default-conn-2	Ethernet	Enabled and running	IPv4 DHCP 192.168.254.140/24
default-conn-3	Ethernet	Enabled and running	IPv6 Automatic fd07:2218:1350:fe:a00: 27ff:fe8b:eb97/64

**Note:** Do not change the **Connection Type** of default network interfaces. If a default interface is not required, edit the interface and uncheck the **Enabled** checkbox.

Instead, disable the interface which will not be used by unchecking the **Enabled** checkbox. If **default-static** and **default-DHCP** are changed to the same configuration method (that is, if both are set to **Static assignment** or both are set to **DHCP**) neither interface works.

## Configure SNMP Manager Settings

Administrative users can configure the **Simple Network Management Protocol** (SNMP) Manager settings. The SNMP Manager allows SNMP TRAP/INFORM messages to be sent from Lighthouse to a configured server any time a node connection status is changed.

To enable the SNMP Manager:

1. Select **SETTINGS > SERVICES > SNMP Manager**.
2. Under the **Settings** section, select the **Enable** checkbox.
3. In the **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**.
6. From the SNMP protocol **Version** select v1 or v2c or v3.

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

For **v1**, 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**.
2. Specify an optional **Engine ID** for sending an SNMP TRAP message. If left blank, the auto-generated Engine ID from the SNMP Service will be used. An Engine ID is not needed for an SNMP INFORM message.

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

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

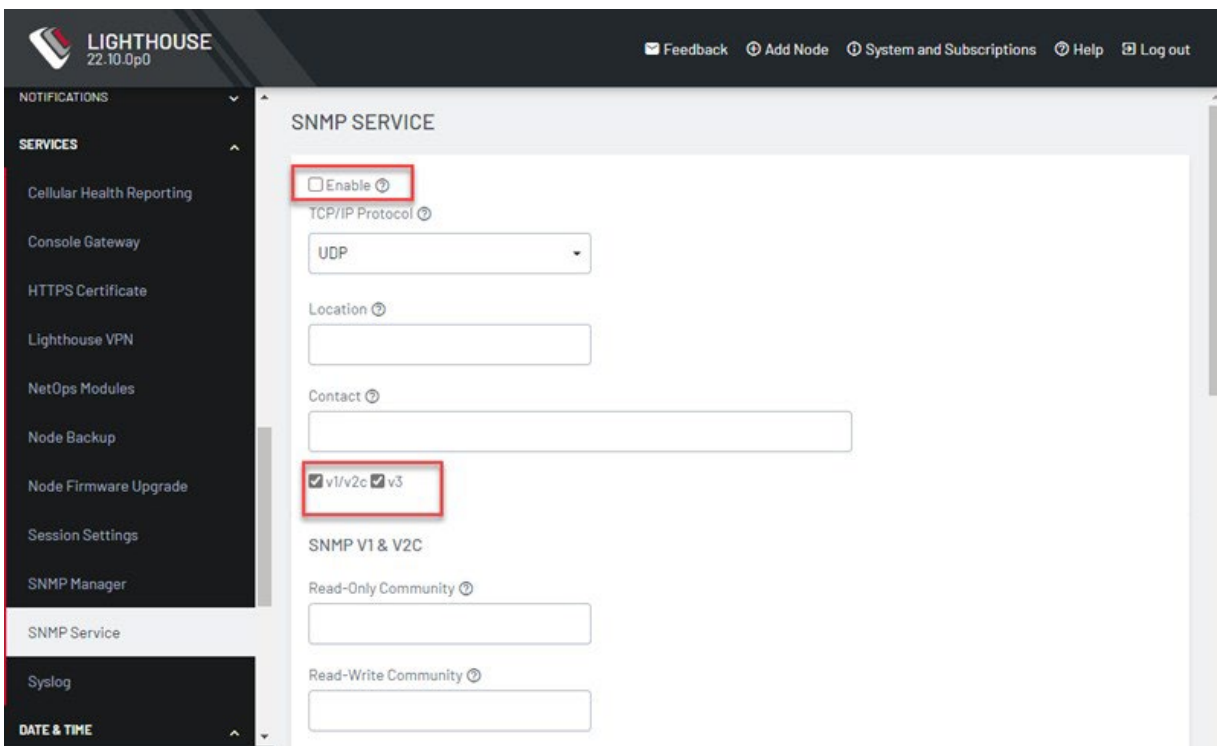
When a node connection status changes, a nodeStatusNotif notification is sent, populated with data about the node's connection status, address and name.

For information on Structure of notifications for Opendaylight nodes, refer to [OG-LIGHTHOUSE-MIB.mib](#).

## Examine or modify SNMP Service

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

Lighthouse supports both v1/v2 and v3 SNMP versions, which can be running at the same time. The SNMP service is not enabled by default. The SNMP service starts after it has been configured correctly. If the user does not provide an **Engine ID**, an auto-generated ID displays. Lighthouse Health statistics (load/uptime/memory usage, etc.) can be retrieved.



The screenshot shows the Lighthouse web interface for configuring the SNMP Service. The left sidebar lists various services, with 'SNMP Service' selected. The main content area is titled 'SNMP SERVICE'. At the top, there is an 'Enable' checkbox, which is checked and highlighted with a red box. Below it, there is a 'TCP/IP Protocol' dropdown menu set to 'UDP'. Further down, there is a 'Location' text input field and a 'Contact' text input field. Below these, there are two checkboxes: 'v1/v2c' and 'v3', both of which are checked and highlighted with a red box. At the bottom, there is a section titled 'SNMP V1 & V2C' with two text input fields: 'Read-Only Community' and 'Read-Write Community'.

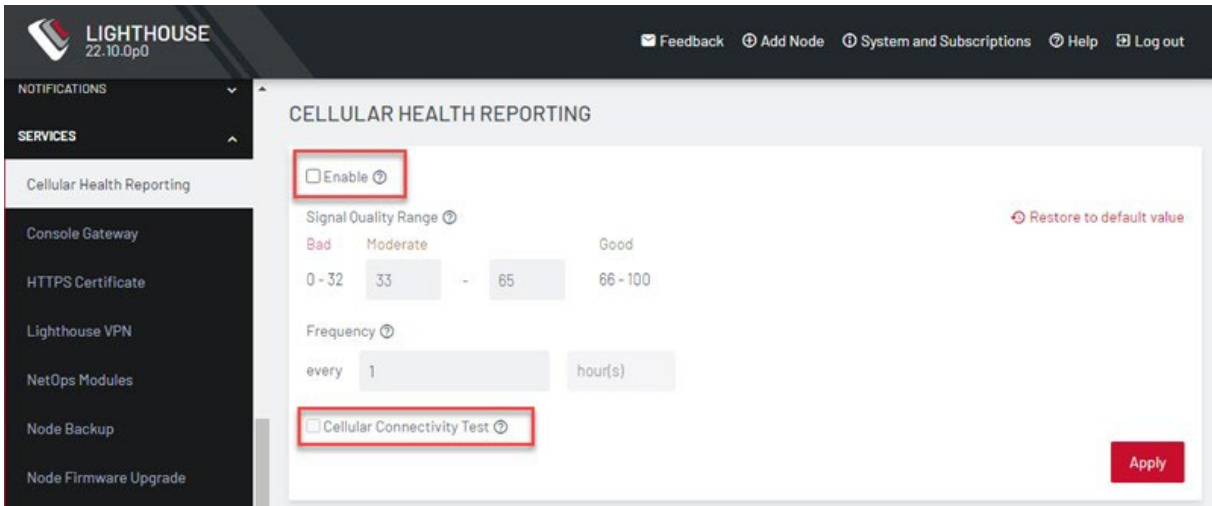
To enable SNMP Service:

1. Select the **Enable** checkbox.
2. Choose from the **v1/v2c** and **v3** checkboxes.

3. Fill in the appropriate information for the SNMP versions.
4. Click **Apply**.

## Cellular Health Settings

Administrative users can control the cellular health reporting settings under **SETTINGS > SERVICES > Cellular 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

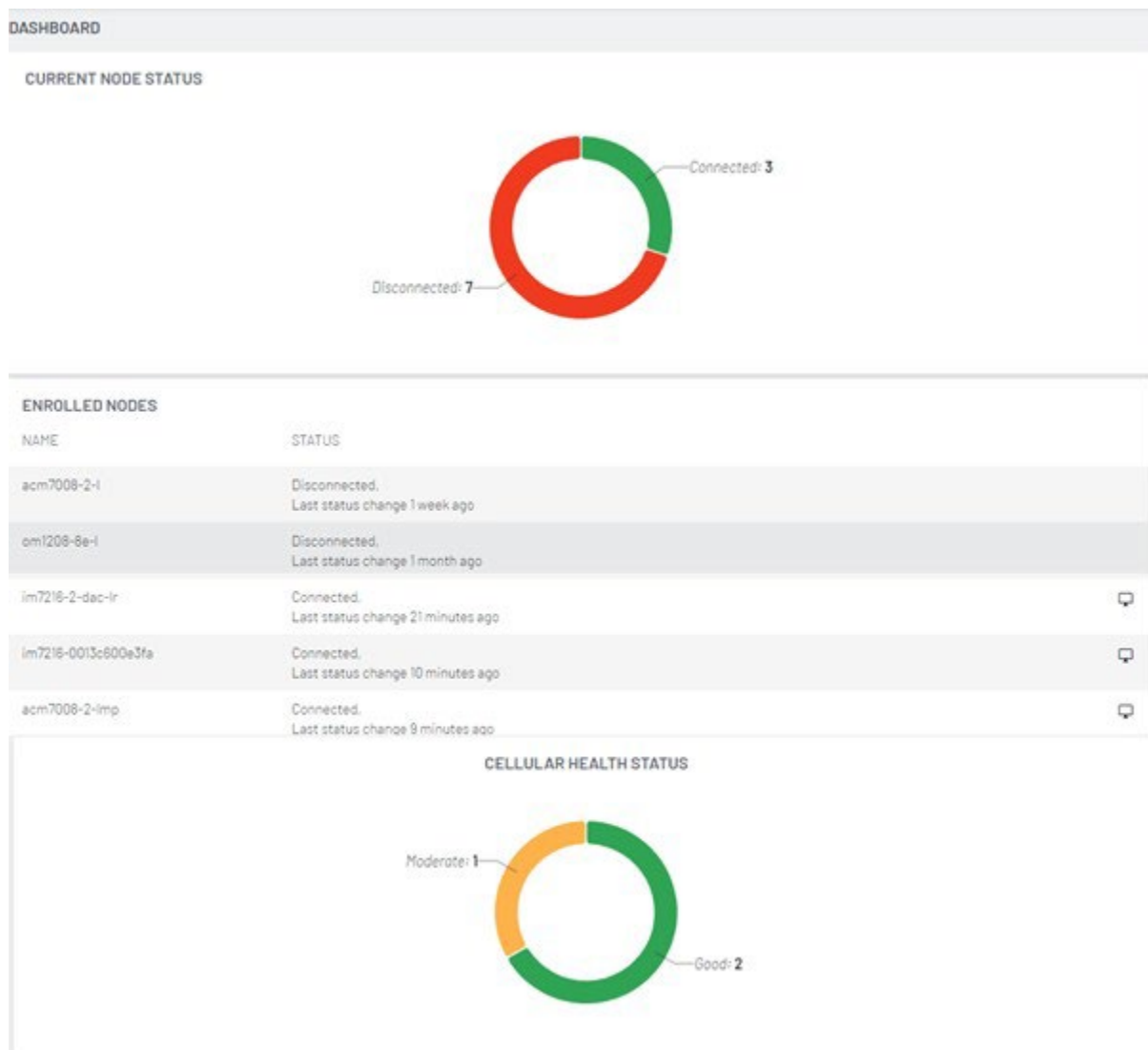
If a managed node has the modem disabled/off, an appropriate status is shown in Lighthouse for the node.

- Check the **Enable** box to enable Cellular Health monitoring.
- If required, adjust the signal quality ranges corresponding to **Good**, **Bad**, and **Moderate**. This will change the Cellular Health information displayed in various node lists and on the **Dashboard**.
- Adjust how frequently Lighthouse will check the signal quality.

- Finally, you can run a periodic **Cellular Connectivity Test** which will make sure the cellular can actually connect. This will use cellular data.

## Cellular Health Dashboard

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



## Changing Docker IP Ranges

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

To avoid this, you can change these settings.

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

Login to the Lighthouse shell CLI as a Lighthouse Administrator or the root user

Ascertain the number of running containers to ensure you select an appropriate subnet size

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

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

```
ogconfig-cli
```

Set the IP Range of the Docker subnet in CIDR format

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

Set the IP Range of the NetOps subnet in CIDR format

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

Push the config to become the running config

```
push
```

Exit the configuration management

```
exit
```

Restart Docker

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

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.



## 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.

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

```
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

## To query licensing information:

22.11 - REVISION 1	Setting up Lighthouse	105
--------------------	-----------------------	-----

ogLhLicenseStatus:  
ogLhLicInstalled  
ogLhLicSupported  
ogLhLicExpiry  
ogLhLicStatus  
ogLhLicFeatureName

**To query enrolled node cellular health information:**

`ogLhNodeCellularHealth`

SNMP commands such as `snmpwalk` or `snmpgetretrieve` Lighthouse specific information.

Setup: SNMP is configured with version 1 and public is community string

Lighthouse public IP address is 192.168.1.1

All MIBs, including Lighthouse MIB are available in **`/usr/share/snmp/mibs`**

## Examples of Lighthouse MIB queries using SNMP:

Walk through the entire **ogLighthouseMib** using name:

```
snmpwalk -m ALL -v1 -c public 192.168.1.1 ogLighthouseMib  
snmpwalk -m ALL -M /usr/share/snmp/mibs -v1 -c public 192.168.1.1  
ogLighthouseMib
```

Walk through the entire **ogLighthouseMib** using the OID directly:

```
snmpwalk -m ALL -M /usr/share/snmp/mibs -v1 -c public 192.168.1.1  
1.3.6.1.4.1.25049.18.1
```

Get the total nodes enrolled in Lighthouse:

```
snmpget -m ALL -v1 -c public 192.168.1.1 ogLhNodesTotal.0  
snmpwalk -m ALL -v1 -c public 192.168.1.1 ogLhNodesTotal
```

Get serial number with enrolled node having VPN address 192.168.128.2:

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

Get cellular health for all enrolled nodes:

```
snmpwalk -m ALL -c public -v 1 192.168.124.143 ogLhNodeCellularHealth  
OG-LIGHTHOUSE-MIB::ogLhNodeCellularHealth.192.168.128.2 = INTEGER:  
good(4)  
OG-LIGHTHOUSE-MIB::ogLhNodeCellularHealth.192.168.128.3 = INTEGER:
```

good (4)

OG-LIGHTHOUSE-MIB::ogLhNodeCellularHealth.192.168.128.4 = INTEGER:

bad (2)

OG-LIGHTHOUSE-MIB::ogLhNodeCellularHealth.192.168.128.5 = INTEGER:

unknown (0)

OG-LIGHTHOUSE-MIB::ogLhNodeCellularHealth.192.168.128.6 = INTEGER:

bad (2)

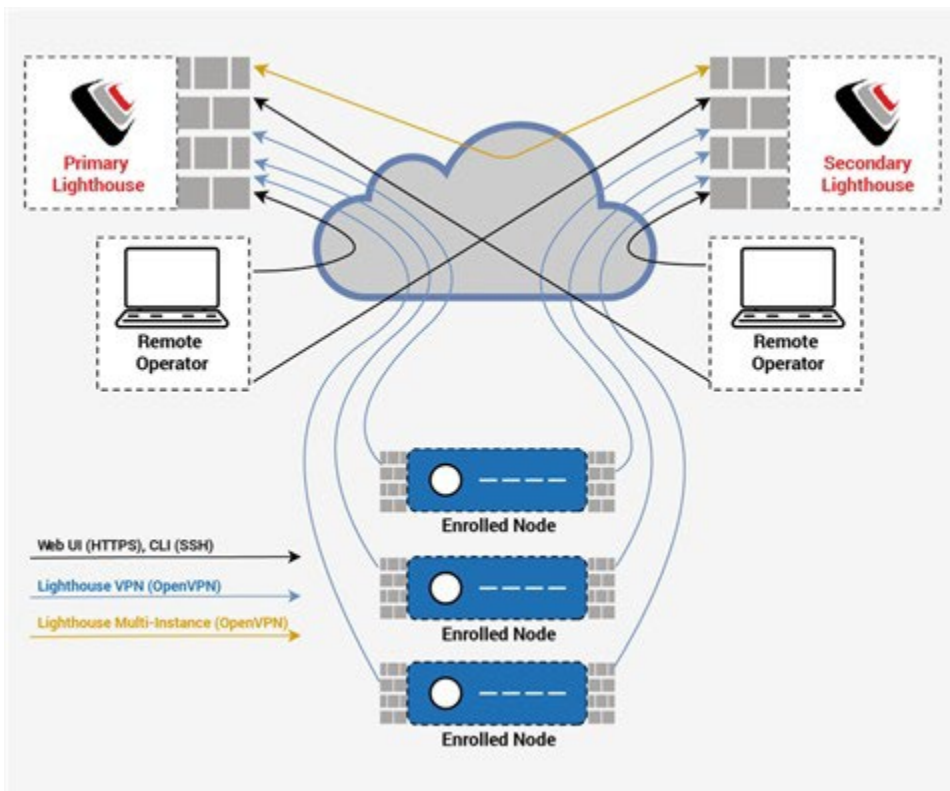
## Setting up Multiple Instances of Lighthouse

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 will then update the information displayed on the secondary instance. For more details, see Lighthouse Architecture in "[Lighthouse Overview](#)" on page 17



## Setting up a multiple instance

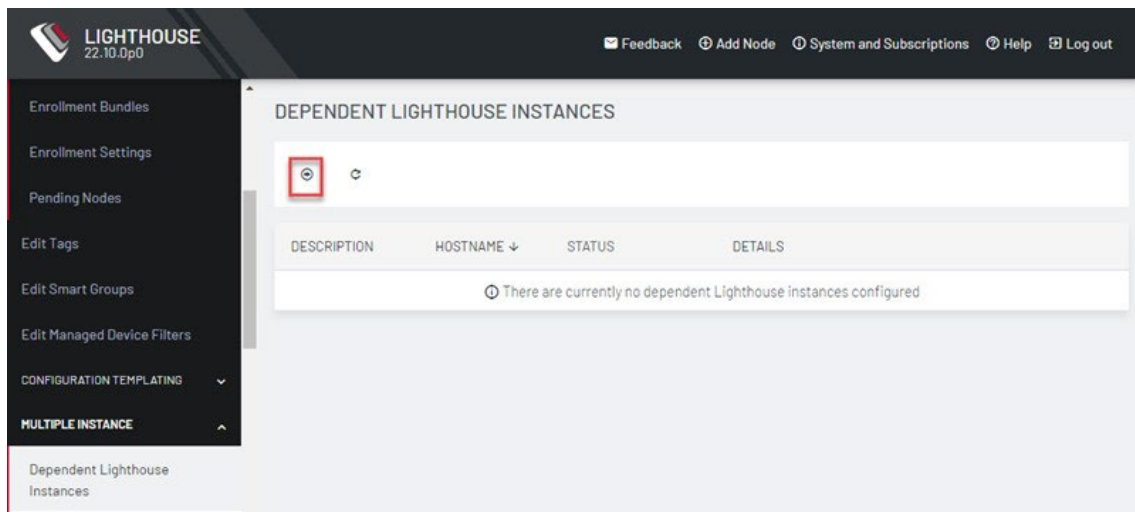
Lighthouse supports up to 10 secondary instances for each primary Lighthouse instance. A secondary instance is also known as a dependent instance.

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.

To set up a multiple instance feature on the primary Lighthouse:

1. On the primary Lighthouse, click **Configure > MULTIPLE INSTANCE > Dependent Lighthouse Instances**.





2. Click **Add**. Enter the following details of a Lighthouse instance to enroll it as a secondary instance:

**Description**

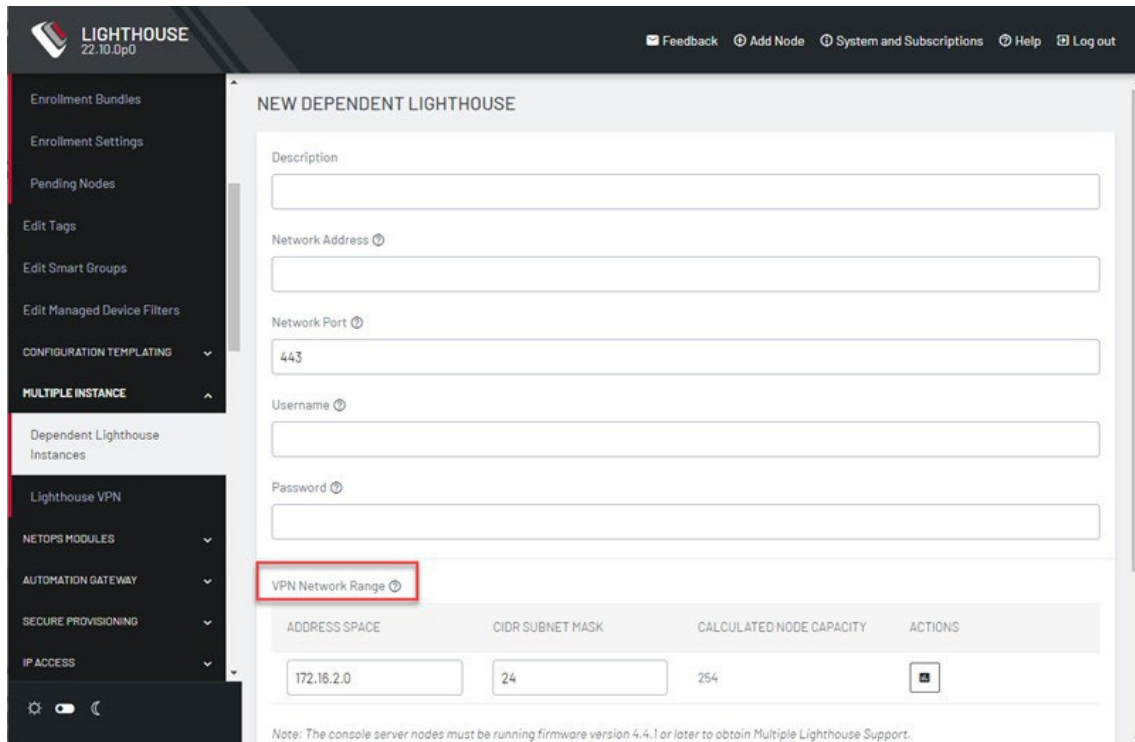
**Network address**

**Network port**

**Username**

**Password**

In the **VPN Network Range** enter a valid, unused network subnet to use as the dependent lhvpn address range. See ["Configuring Subnets for a Multiple Instance Lighthouse"](#) on page 116.



**NEW DEPENDENT LIGHTHOUSE**

Description

Network Address


Network Port

443

Username

Password

VPN Network Range

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

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

**Note:** The secondary Lighthouse instance must be able to reach the primary instance on UDP Port 1195.

3. Click **Apply**.

4. Dependent Lighthouse Enrollment displays status as it moves from **Pending** > **Registered** > **Enrolled**.
5. When the VPN connection is established between primary and secondary Lighthouse, this page will display **Connected** with the time since the last status change and **Disconnected** when the connection is lost. Any errors in the Enrollment process will display in the status column.

## Enabling Alternate REST API Ports

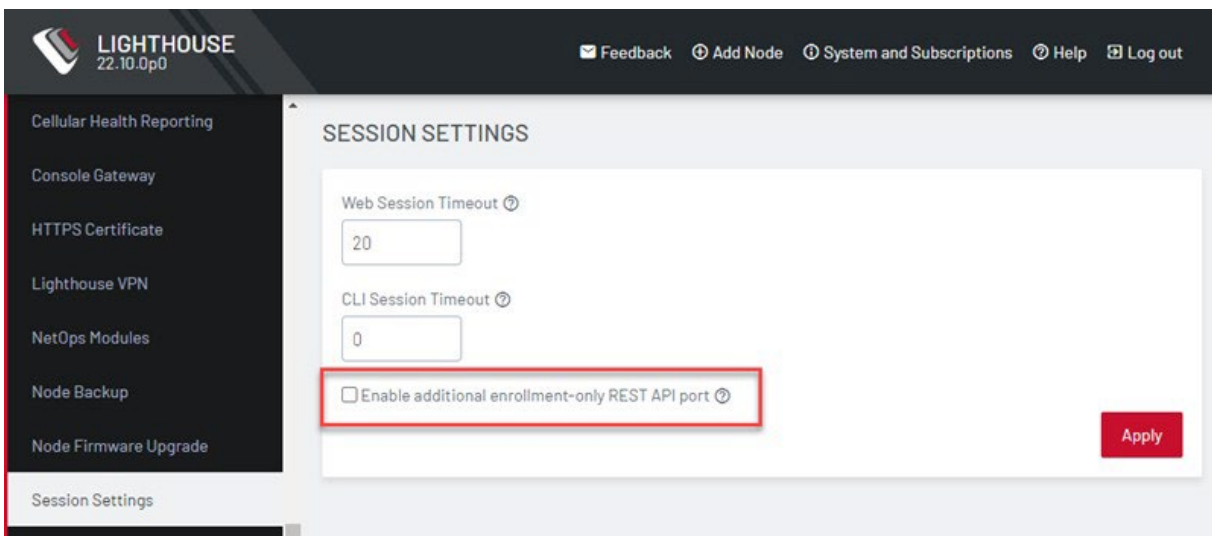
If you are planning to use the alternate REST API ports, you will need to make sure this option is enabled on both the primary and dependent Lighthouse servers, prior to Enrollment of the dependent(s). Lighthouse will prevent the Enrollment of a dependent Lighthouse instance if there is a mismatch in these settings. If this occurs, the message "Lighthouse is using Alternate API port" will be displayed on the Dependent Lighthouse Instances 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.

The Alternate REST API Port is configured as follows:

1. Select **Settings > Services > Session Settings**. The page displays:



2. Select **Enable additional enrollment-only REST API port**.

**Note:** The alternate REST API Port for enrollment is 8443

## Configuring Subnets for a Multiple Instance Lighthouse

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

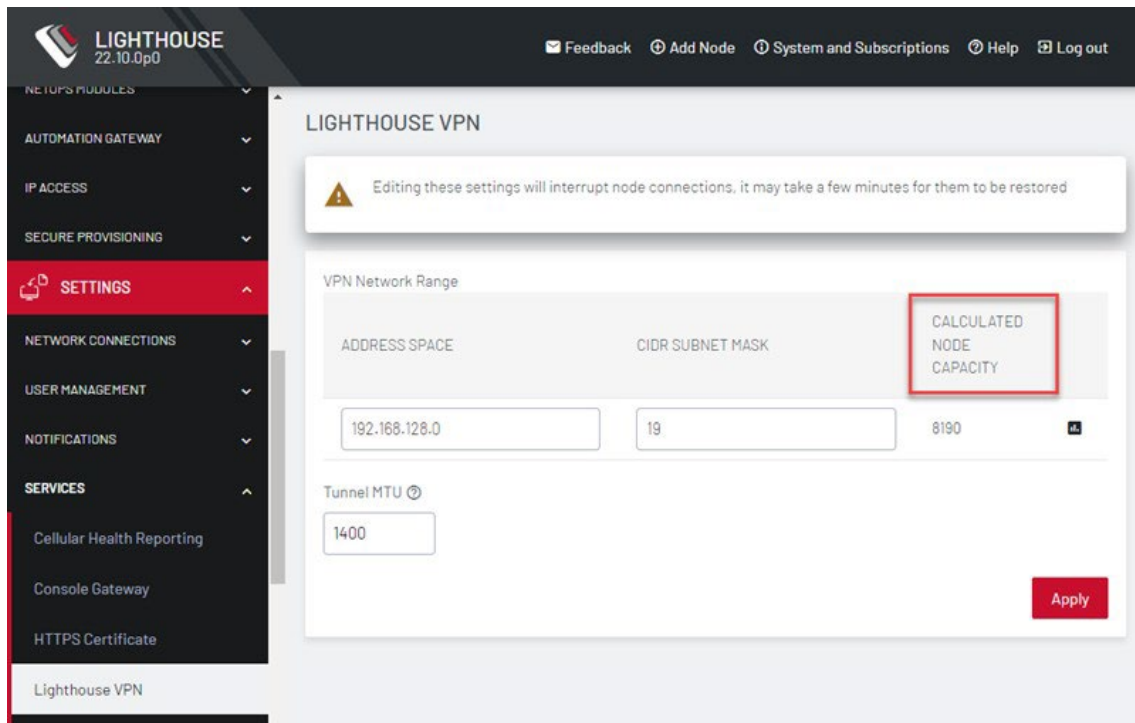
- Between each instance and its nodes
- Between the primary and dependent Lighthouses.

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

## Configuring the Subnets

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

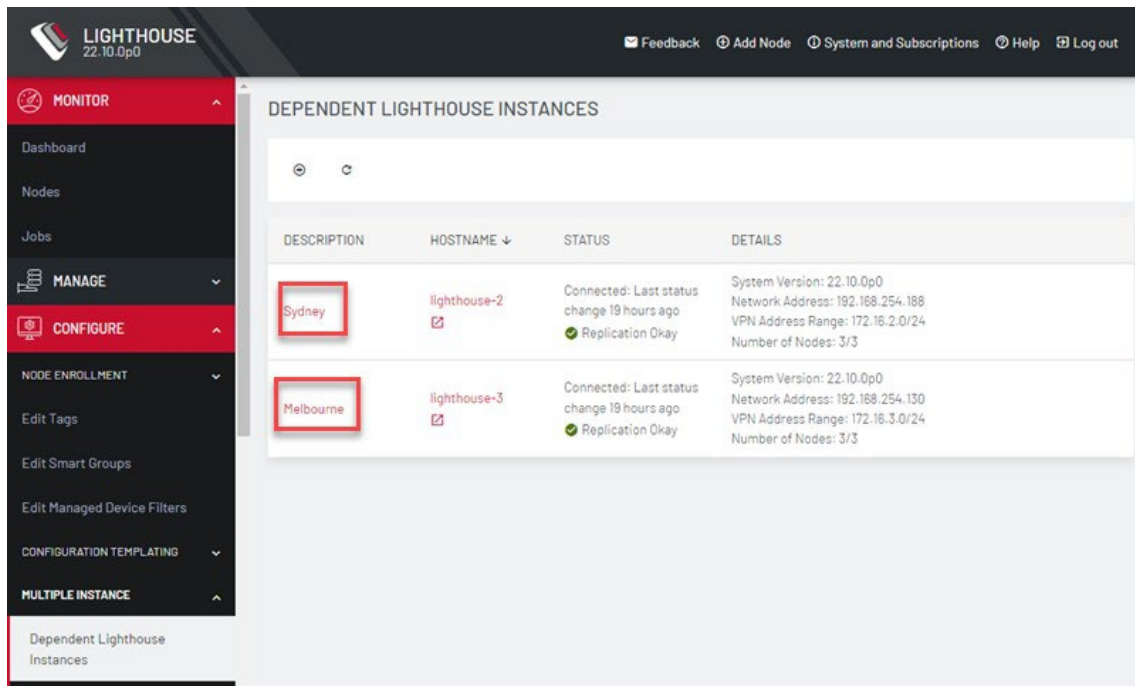
1. Select **SETTINGS > SERVICES > Lighthouse VPN** on the primary Lighthouse.



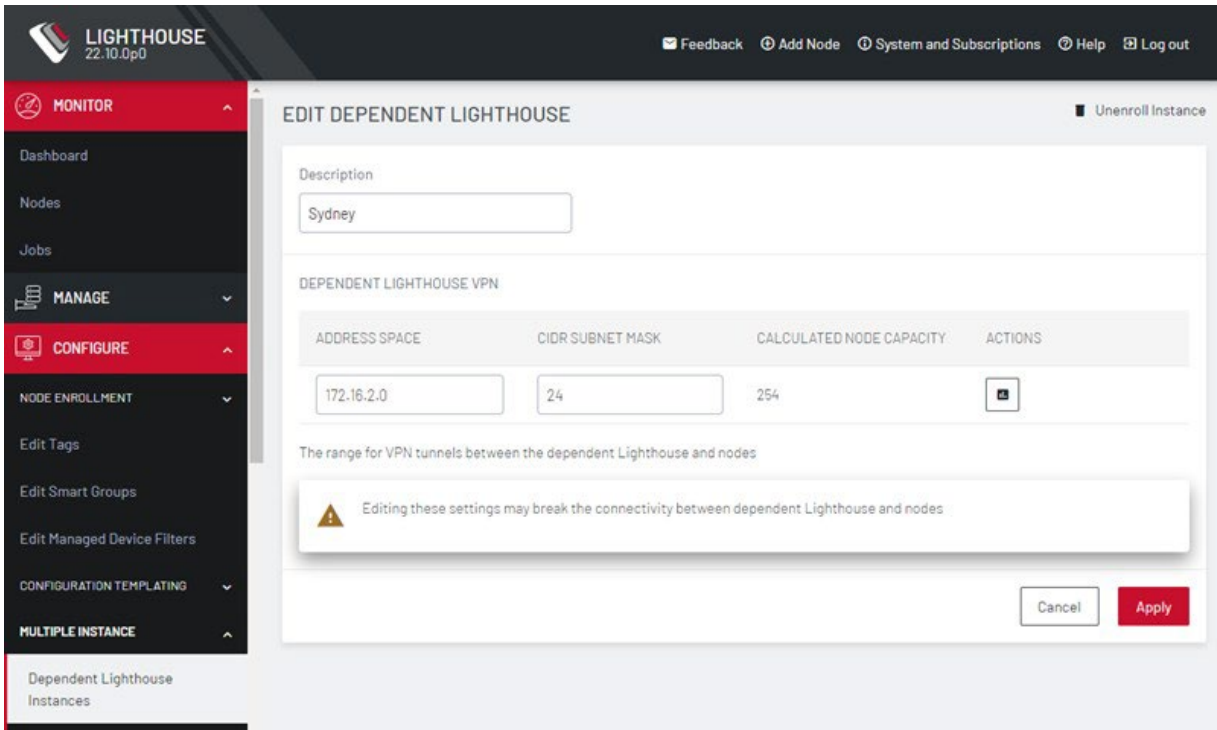
**Calculated Node capacity** displays the addressable nodes based on the network **Address Space** and **CIDR Mask**.

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

- To configure the subnet between each secondary Lighthouse and its nodes select **CONFIGURE > MULTIPLE INSTANCE > Dependent Lighthouse Instances**.
- The Dependent Instances page displays. Click on the name of the dependent Lighthouse to be modified.



- The Edit Dependent Lighthouse page displays.



**LIGHTHOUSE** 22.10.0p0

Feedback Add Node System and Subscriptions Help Log out

**MONITOR**

Dashboard

Nodes

Jobs

**MANAGE**

**CONFIGURE**

NODE ENROLLMENT

Edit Tags

Edit Smart Groups

Edit Managed Device Filters

CONFIGURATION TEMPLATING

MULTIPLE INSTANCE

Dependent Lighthouse Instances


## EDIT DEPENDENT LIGHTHOUSE

Unenroll Instance


Description

Sydney

DEPENDENT LIGHTHOUSE VPN

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

The range for VPN tunnels between the dependent Lighthouse and nodes:

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

Cancel Apply

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

Instance specific information includes:

**Hostname**

**Time zone**

**Networking**

**External interfaces**

The instance specific information is 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 will have a value of `Primary` for its role, and dependent 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` will output `n` for a primary Lighthouse or `y` for a secondary.

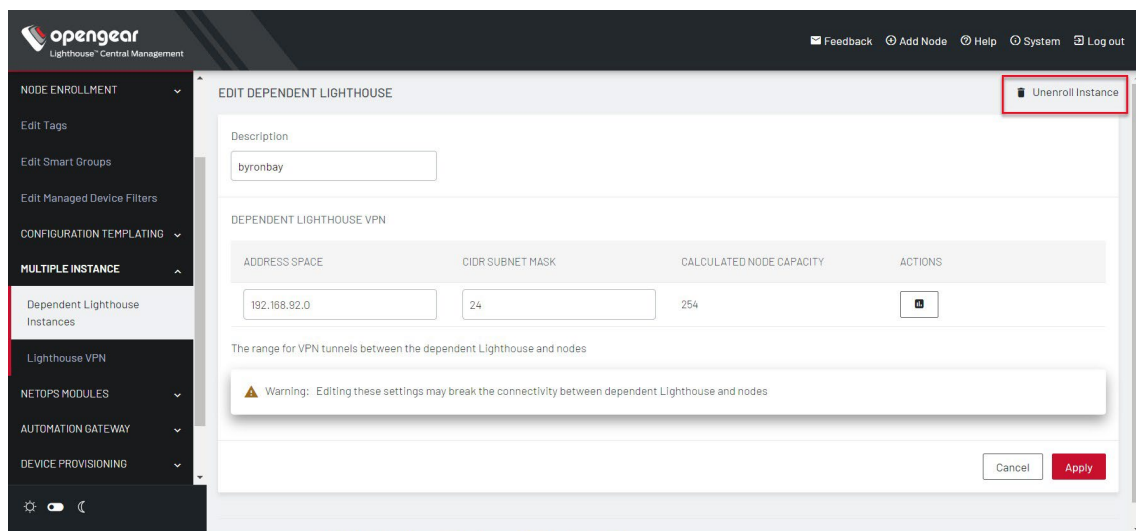
6. You can modify the configuration of secondary lighthouses from the primary Lighthouse. For example, to update the hostname of the secondary Lighthouse, run the following commands on the primary Lighthouse:

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

## Disconnecting a secondary instance

To disconnect Dependent Lighthouse instances from the primary Lighthouse:

1. Click **CONFIGURE > MULTIPLE INSTANCE > Dependent Lighthouse Instances**, and click the instance name.x
2. The secondary Lighthouse displays. Click the Unenrol Instance icon in the top right of the page.



3. Click **Yes** in the message that asks you to confirm.
4. You must manually remove the connection to the secondary Lighthouse from each connected node. Click the **Delete** link in the Console Server.

## Promoting a secondary instance

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

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

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

```
promote-secondary-lighthouse
```

Remove all dead connections from node side using the node's web UI. The Promotion tool deletes connection between primary and secondary instance but does not touch node connections.

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

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

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

## Upgrading a multiple instance Lighthouse

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

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

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

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



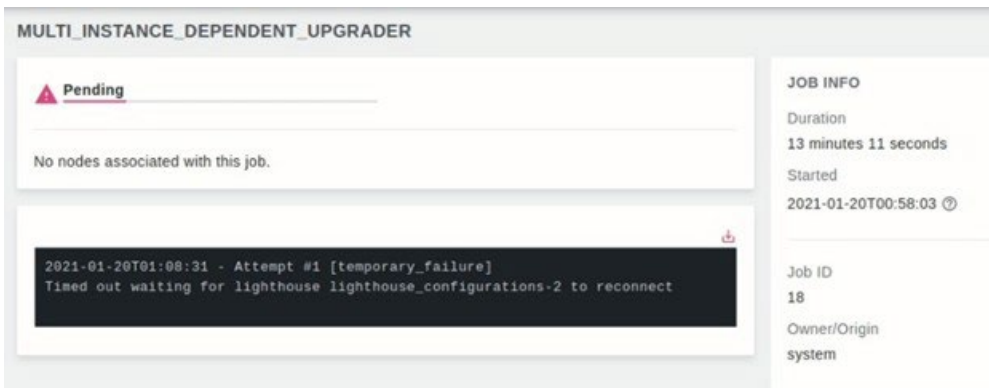
When the primary Lighthouse is updated, any secondary Lighthouses will be updated in a rolling fashion after the primary has successfully booted. If any Lighthouse fails to successfully update along the way, the update will stop.

## Upgrading Dependent Multiple Instances of Lighthouse

To upgrade a Multiple Instance Lighthouse:

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

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



The screenshot shows a web interface titled "MULTI\_INSTANCE\_DEPENDENT\_UPGRADER". On the left, a "Pending" status is indicated with a red triangle icon and a progress bar. Below this, it says "No nodes associated with this job." In the center, a black box displays a log message: "2021-01-20T01:08:31 - Attempt #1 [temporary\_failure] Timed out waiting for lighthouse lighthouse\_configurations-2 to reconnect". On the right, a "JOB INFO" sidebar lists: Duration (13 minutes 11 seconds), Started (2021-01-20T00:58:03), Job ID (18), and Owner/Origin (system).

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

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

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

## Upgrading Lighthouse

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.
- Due to disk restructuring, there were no `lh_upg` files for 20.Q3.x.
- Incremental upgrades to Lighthouse using `lh_upg` files are only supported from 20.Q3.x and not earlier releases.

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

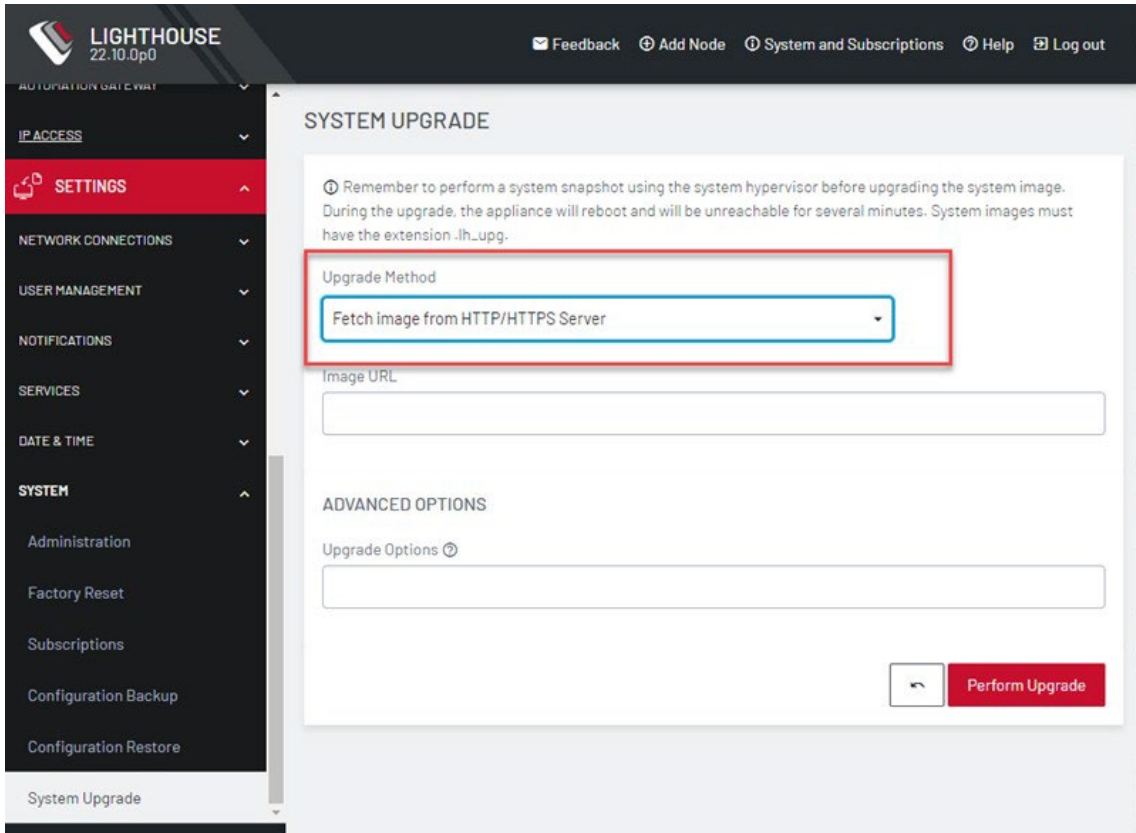
Once the upgrade is complete, the Lighthouse instance reboots. It is unavailable during the reboot process.

## Upgrading the system from within Lighthouse

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

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

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



**LIGHTHOUSE**  
22.10.0p0

Feedback Add Node System and Subscriptions Help Log out

**SYSTEM UPGRADE**

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

Upgrade Method  
Fetch image from HTTP/HTTPS Server

Image URL

ADVANCED OPTIONS

Upgrade Options ⓘ

Perform Upgrade

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

- a. Enter the URL for the system image in the Image URL text-entry field.
  - b. Click Perform Upgrade.
4. Or if upgrading via **Upload Image**:
- a. Click the **Choose file** button.
  - b. Navigate to the directory containing the appropriate upgrade image file.
  - c. Select the upgrade image file and press **Return**.
  - d. Click **Perform Upgrade**.

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

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

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



## Upgrading the Lighthouse system via the local terminal

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

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

To upgrade Lighthouse instance's system using the Lighthouse local terminal:

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

At the local terminal bash shell prompt, enter a URL. It must be URL-encoded:

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

5. Press Return.

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

1. At the local terminal bash shell prompt enter:  
`sysflash /path/to/system-upgrade-image.lh_upg.`

2. Press Return.

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

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

```
sysflash -h or
```

```
sysflash --help
```

For more information on `sysflash` See the Command line tools chapter.

## Upgrade NetOps Modules

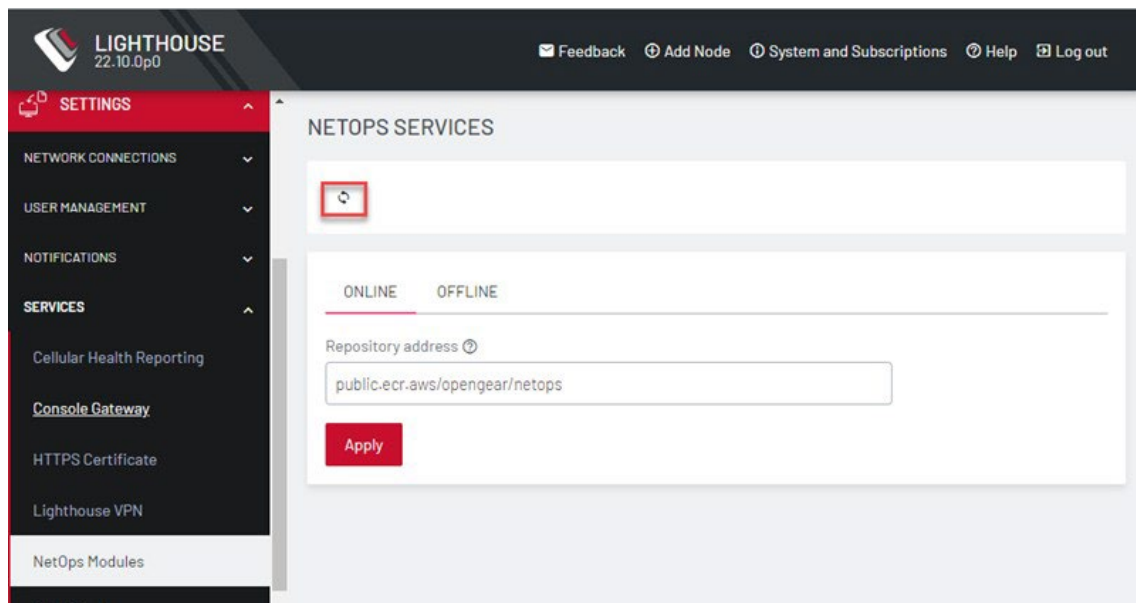
NetOps Modules are released independently of Lighthouse software or Operations Manager firmware.

NetOps releases are uploaded to Opendgear's file server, where they can be fetched by Lighthouse then deployed to all activated nodes by Lighthouse.

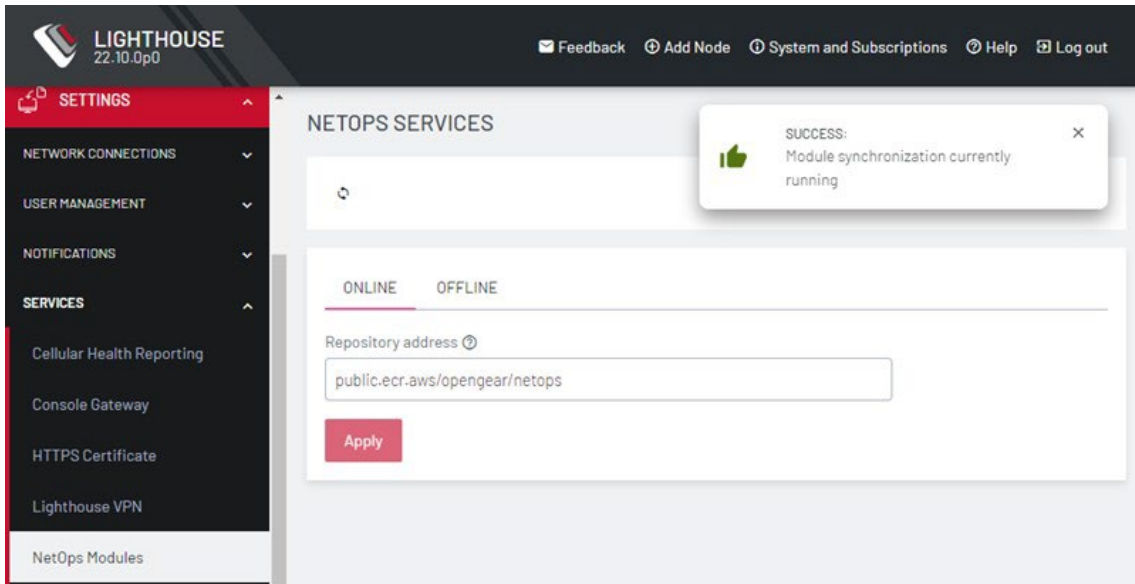
Node upgrades may be carried out through either the Lighthouse UI or the CLI.

### PROCEDURE for Lighthouse UI

1. Log in to the Lighthouse web UI as a Lighthouse Administrator or the root user.
2. From the menu, select **SETTINGS > Services > NetOps Modules**. Select either Online or Offline.



- Click the  **Synchronize** icon. A message displays with the status.



- From the menu, select **CONFIGURE NODES > NetOps Modules > Manage Modules**.
- Click the **Redeploy** icon.

## PROCEDURE for Lighthouse CLI

Replace **root** and **default** with a Lighthouse Administrator or root credentials, then run the following:

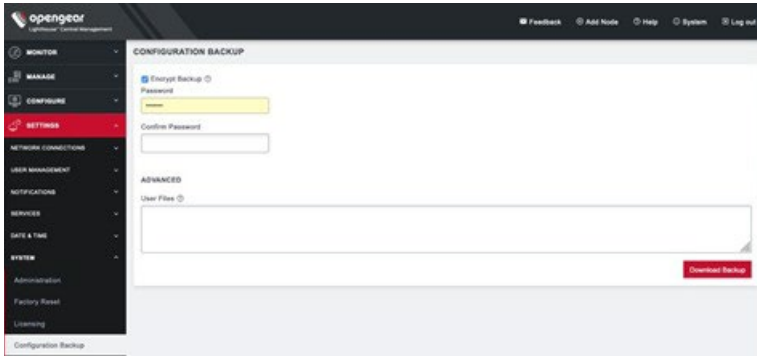
```
USERNAME=root
PASSWORD=default
/etc/scripts/netops_sync_handler
token=$(curl -k -L -d '{
  "username":"'$USERNAME'", "password":"'$PASSWORD'"
}'
  "https://127.0.0.1/api/v3.0/sessions/" | python -c 'import sys, json;
  print json.load(sys.stdin)["session"]')
```

```
curl -k -L -H "Content-Type: application/json" -H "Authorization:  
Token $token"  
"https://127.0.0.1/api/v3.0/netops/modules/dop/redeploy"
```

## Configuration Backup

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

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



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

## Configuration restore

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

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



2. Locate the file you downloaded when you performed the Configuration Backup.
3. If you chose **Encrypt backup** when creating the backup, enter the **Backup Password**.
4. Click **Restore Backup**.
5. A **Configuration Restore Confirmation** dialog opens, click Yes.



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

## Shut Down or Restart Lighthouse

The following sections describe the process to shutdown or restart Lighthouse.

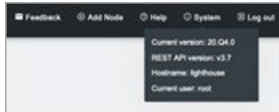


## Finding the current Lighthouse instance version

There are two ways to find the current Lighthouse version.

## Using Lighthouse

1. Click **System & Subscriptions** on the Lighthouse banner.
2. The Details menu displays, listing the Lighthouse instance's:
  - Current version
  - REST API version
  - Hostname, and
  - Current user.



## Via the local Lighthouse shell

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

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

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

## Via other information sources

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

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

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

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

For example:

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

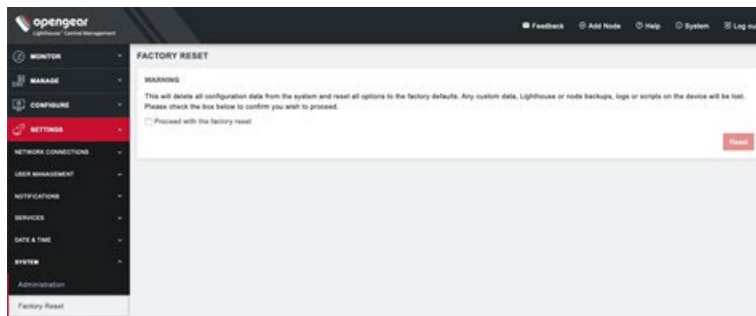
## Returning a Lighthouse instance to factory settings

At some stage, you may need to 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 will be overwritten and user files will be deleted. If you wish, you can create a backup of the configuration and any desired user files.

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

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



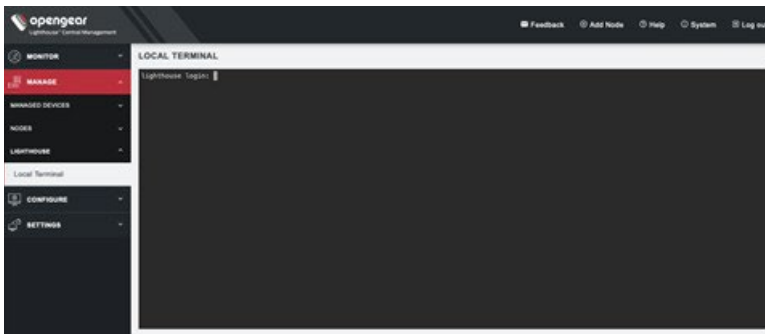
3. Select the **Proceed with the factory** reset checkbox.
4. Click **Reset**.

## Shut Down a Running Lighthouse Instance

Shut down a running Lighthouse instance

To shut down a running Lighthouse instance:

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



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

## Restarting a Running Lighthouse Instance

To restart a running Lighthouse instance, follow the first three steps of the "[Shut Down or Restart Lighthouse](#)" on [page 135](#) instance procedure above. At the shell prompt, enter one of these commands and press Return:

```
reboot
```

or

```
shutdown -r now
```

The Lighthouse virtual machine shuts down and reboots.

## Managing Lighthouse Nodes

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

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

This section covers:

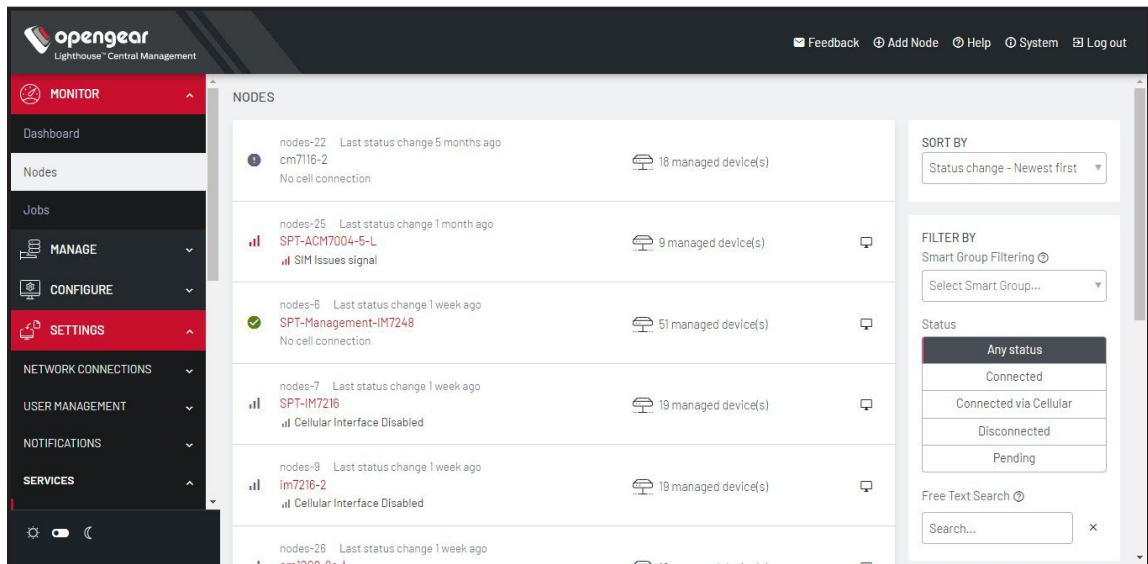
- Enrolling nodes
- The Enrolled Nodes page
- Filtering pages displaying nodes
- Creating and editing Smart Groups
- Creating and editing Managed Device Filters
- Connecting to a node's web management interface
- Connecting to a node's serial ports via Console Gateway



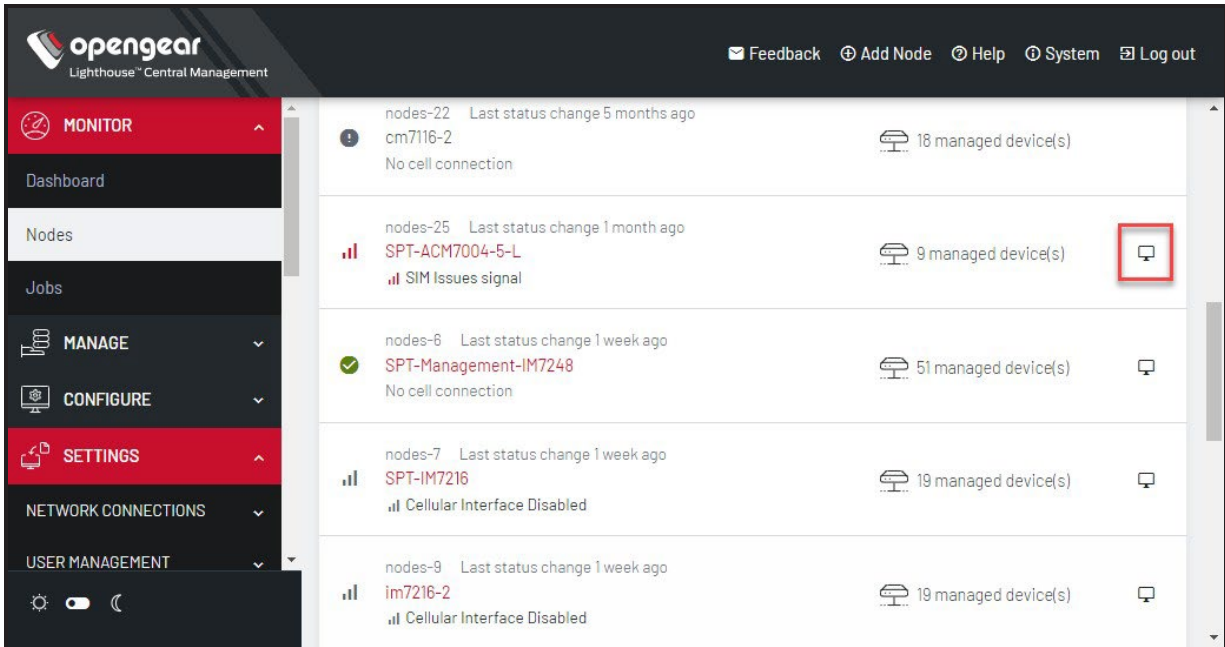
## Monitor Nodes

Lighthouse allows you to view all nodes including connected, not connected, enrolled, and pending, as well as nodes using cellular. **MONITOR > Nodes** displays the cellular health and number of ports. You can also click the icon on the right of connected nodes to access the Web UI of the node





1. Select **MONITOR > Nodes**.



2. Use the **SORT BY** menu to sort the display of nodes, by *Status Change*, *Name* or *Connected Devices*, in ascending or descending order, or newest or oldest.
3. Use the **Status** menu to display the nodes by *Any status*, *Connected*, *Connected via Cellular*, *Disconnected* or *Pending*.
4. Use the **Free Text Search** to *Search* using a text string.
5. The **Monitor Nodes** page displays the result of the search in a Details table.



The screenshot shows the Lighthouse Central Management interface. The top navigation bar includes 'Feedback', 'Add Node', 'Help', 'System', and 'Log out'. The left sidebar has a 'MONITOR' tab selected, with sub-items 'Dashboard', 'Nodes', and 'Jobs'. Below this are 'MANAGE', 'CONFIGURE', 'SETTINGS' (highlighted), 'NETWORK CONNECTIONS', and 'USER MANAGEMENT'. The main content area displays a list of nodes:

Node ID	Last status change	Managed device(s)	Actions
nodes-22 cm7116-2	5 months ago No cell connection	18 managed device(s)	
nodes-25 SPT-ACM7004-5-L	1 month ago SIM Issues signal	9 managed device(s)	
nodes-6 SPT-Management-IM7248	1 week ago No cell connection	51 managed device(s)	
nodes-7 SPT-IM7216	1 week ago Cellular Interface Disabled	19 managed device(s)	
nodes-9 im7216-2	1 week ago Cellular Interface Disabled	19 managed device(s)	

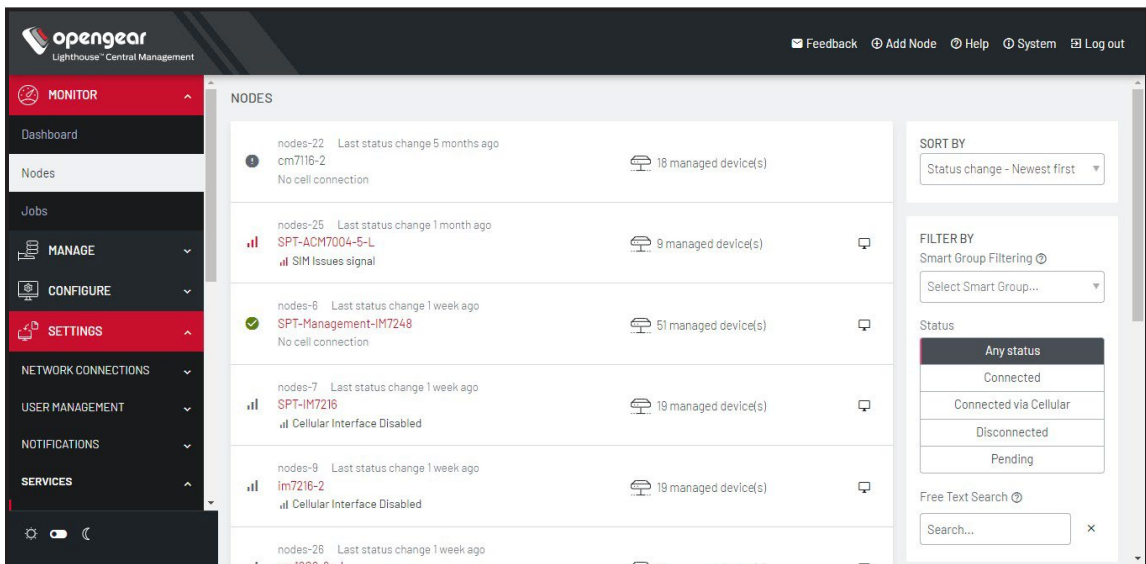
You can click on any connected node to view the node details, list of connected devices, and a list of unconfigured ports. From this detail page, you can access the web terminal and SSH of each connected device.

## Monitor Ports

Lighthouse allows you to view all ports on a node to get a quick snapshot of the health of the nodes. For example, to trace a fault on a node, a network engineer will want to investigate the latest logs for a port.

Select **MONITOR > Nodes** displays the cellular health and number of ports. You can also click the icon on the right of connected nodes to access the Web UI of the node

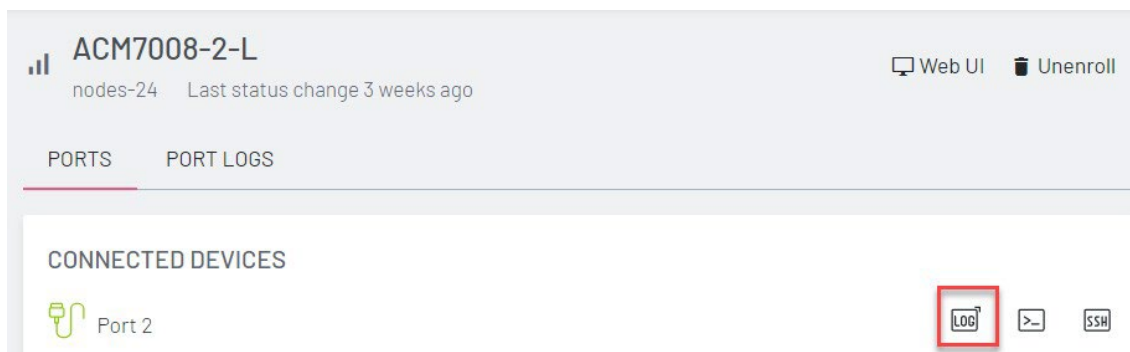
1. Go to **MONITOR > Nodes**.



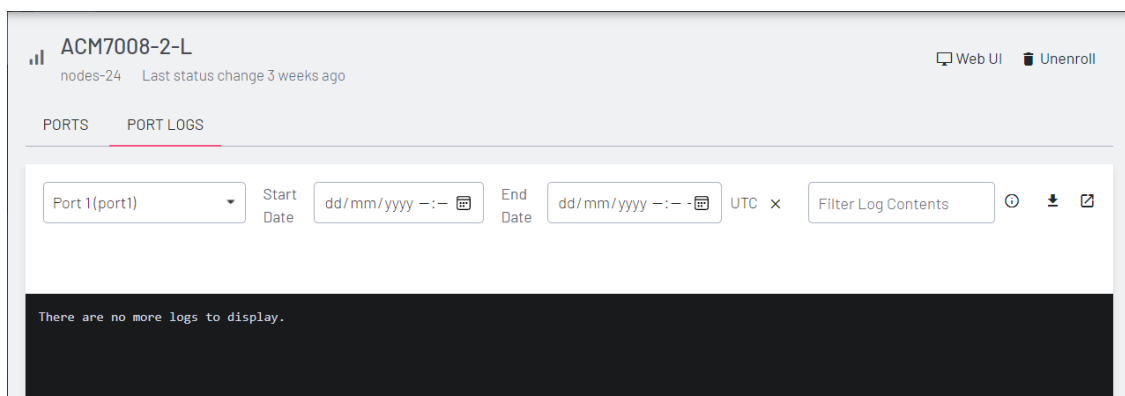
2. Click on the required node. The Node page displays all the ports, as well as information on the port. including lists of **Connected Devices** and **Unconfigured ports, Network Info, Cellular details, Node tags** and **Node details**.



3. Select **Ports** to display the ports of any Connected Device. You can also check the logs of a port by clicking the Log icon in the Port row.



4. Or you can select **Port Logs**.
5. The **Port Logs** page displays.



Select the **Port** from the list, and specify the date range (**Start Date** and **End**

**Date)** to view the logs, high display in the log window below. You can also access the web terminal and SSH of each connected device.

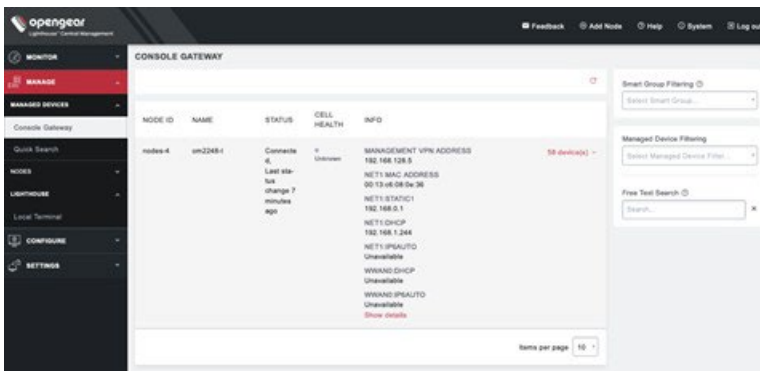
6. Alternatively, you can search for a log with a particular text, using the **Filter Log Contents** Logs matching the search parameter display in the log window.
7. You can also download the displayed logs, or open a new window to view the logs in more detail.

## Filtering pages displaying nodes

If want to quickly find a particular node or set of nodes, you can drill down to specific nodes by using filters. There are three ways to filter search results:

- **Free Text Search**
- **Smart Group Filtering**, and
- **Managed Device Filtering**.

The filters can be used independently from each other or in combination. **MANAGE > MANAGED DEVICES > Console Gateway** uses all of them because it is the only page which lists all nodes with managed devices.



## Filtering using the Free Text Search field

The **Free Text Search** input field allows near real-time filtering of nodes and managed devices using the following criteria:

- . Node name
- . Firmware version
- . Management VPN address
- . MAC address
- . Serial number and
- . Port label.

Enter your search term in the **Free Text Search** field and press **Enter**..

The **Free Text Search** field treats multiple search terms separated by the space character as being combined with the logical AND operator, returning results only if all terms are present in the item.

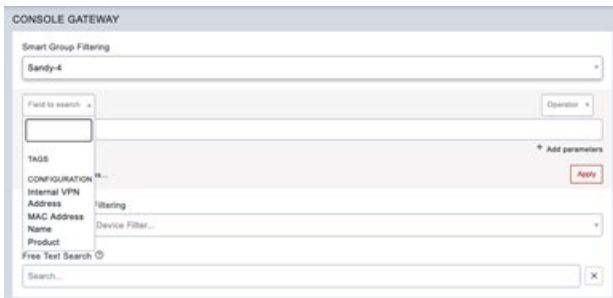
For example, the search phrase `production switch` returns only nodes that contain both `production` AND `switch` anywhere in searchable fields.

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

## Filtering using the Smart Group Filtering drop-down menu

You can search for groups of nodes by using the **Select Smart Group** field. Enter the search criteria in **Select Smart Group** and a menu displays the nodes that belong to the selected group. See Creating Smart Groups for how to create such groups.

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



In the example above, the **CONFIGURE > NODE Enrollment > Enrolled Nodes** page is being filtered on the **Sandy-4** smart group.

To add more filtering options:

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

The list of nodes that meet the specified criteria display.



## Filtering using the Managed Device Filtering drop-down menu

You can search for groups of nodes by using the ***Managed Device Filter*** field.

Enter the search criteria in **Managed Device Filter** and the page displays the nodes that belong to the selected group. See [Creating Smart Groups](#) for how to create such groups.

After a particular **Managed Device Filter** has been selected, further filtering options become available under ***Fields to search:***

To add more filtering options:

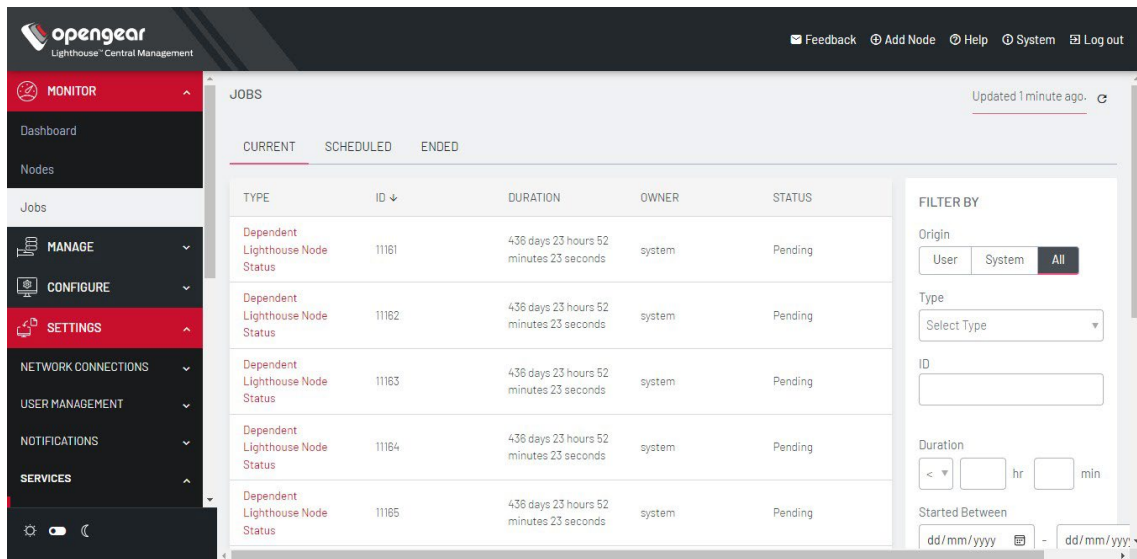
1. Click **Field to search**.
2. Select a field and enter a value in the text box.
3. Select an **Operator** from the drop-down box on the right.
4. Click the **+ Add Parameters** button. Select a parameter.
5. Click **Apply**.

## Monitor Jobs

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

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

1. Go to **MONITOR > Jobs**.



2. At the top of the page, select the type of jobs you wish to see - **CURRENT**, **SCHEDULED**, or **ENDED**
3. You can also use the **FILTER BY** fields to further refine your results by **Origin**, **Type**, **ID**, **Job duration**, **Dates the job began**, and **Owner**.

## Enrol Nodes

Enrolling nodes is the process of connecting nodes to Lighthouse to make them available for access, monitoring, and management.

Enrollment can be performed via:

- the Lighthouse Web UI
- the Node Web UI
- ZTP
- USB key.

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

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

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

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

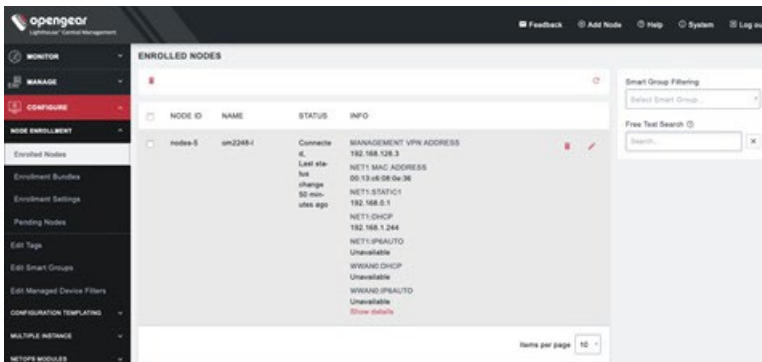
Enrollment is a two-step process:

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

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

## The Enrolled Nodes page

The Enrolled Nodes page lists all the enrolled nodes on a Lighthouse instance.



To display the enrolled nodes:

1. Select **CONFIGURE > NODE Enrollment > Enrolled Nodes**
2. The enrolled nodes display in the chronological order in which they were enrolled to Lighthouse. For each node, the following details display:
  - model
  - firmware version
  - serial number
  - status.
3. Click **Items per page** to select the number of nodes per page. Choose a default value of 10, 20, 50, 80, or 100 nodes per page, or enter a custom value between 1 and 100.  
 This setting applies to the current user session only and will be lost when the current user logs out.

**Note:** **Items per page** also displays on the **Pending Nodes**, **Console Gateway**, and **Node Web UI** pages.

4. **Status** is the current connection status of the node and displays either of two things:
  - **Connected: Last status change x [time unit] ago:** The time since Lighthouse connected to the console server.
  - **Disconnected: last status change x [time unit] ago:** The time since Lighthouse disconnected from the console server.
  - **Configuration Retrieval Status** displays if any configuration retrieval sections failed when performing a configuration sync with this node, such as Groups, Users, Node Description, Authorization, or Serial Ports.
  - **Configuration Template Run Status** displays the result of the most recent configuration template push on this node, listing which templates finished applying, or failed to apply to the node. This information is displayed until the next template push has completed on this node.

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

- Results of the **Configuration Retrieval Status** and **Configuration Template Run Status** indicate:
  - **Success:** all templates were successfully executed on the node.
  - **Partial Failure:** some templates failed to execute on the node, or some config sections failed to synchronize.
  - **Failure:** all templates failed to execute on the node, or all config sections failed to synchronize.

You can select a summary of each status to see more detailed information as follows:

- Retrieval failed for: section\_name, section\_name, section\_name.
  - Template(s) failed to apply: template\_name, template\_name, template\_name.
  - Template(s) successfully applied: template\_name, template\_name, template\_name.
5. If **SETTINGS > SERVICES > Cellular Health Reporting** is Enabled, the **Cellular Health** column displays the node's current cellular status. If this state is **Good|Moderate|Bad**, click the color indicator or the text to view more detailed health information as follows:

Cellular Health Details for lm7216-0013c6045f08

Cellular IP Address	Status	Conditions	Signal Quality	RSSI	Connection Type	Sim Issues	Connectivity Test
10.92.141.156 2001:8004:1140:a75:1c8f:9d83:9a44:dab8	Up	Follower Disabled	81	-43	lte	No	Connectivity Test Disabled

OK

- Cellular IP Address (IPv4 and IPv6)
- Cellular interface status (Up|Down)
- Conditions
- Signal Quality
- RSSI
- Connection Type
- SIM Issues
- Connectivity Test (Passed|Failed|Connectivity Test Disabled)

6. The INFO column displays more details about each node. Click **Show Details** to see more information about the node, or **Hide Details** to display fewer details.



## Enrollment Bundles

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

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

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

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

## Assigning Subscriptions

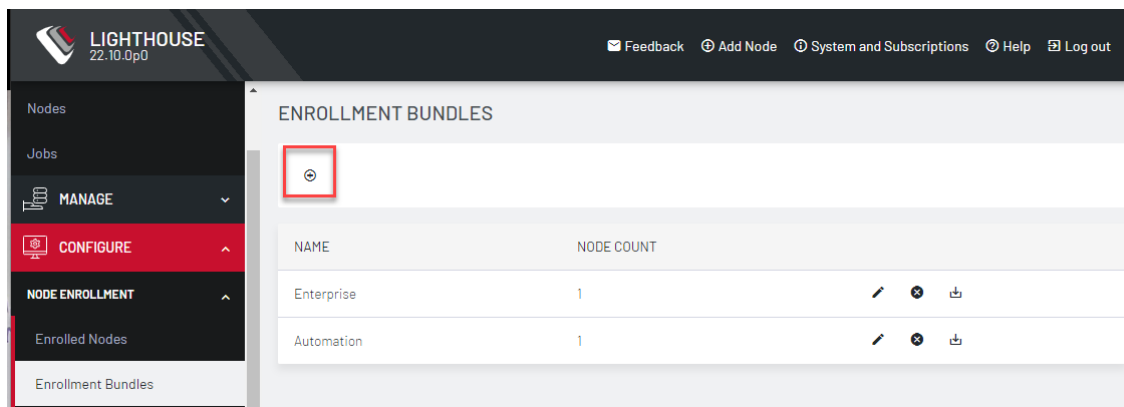
When a user has subscribed to Enterprise Edition, Enterprise Automation Edition, or both then assigning a subscription type to the bundle will be necessary. Based on the feature set for the node to utilize, this will mean they might need to create two bundles:

- An Enterprise tier subscription bundle to enrol nodes, and perform other basic functions (set up AAA, run scripts, create groups/users, set up logging, etc)
- An Automation Edition tier subscription allows all basic functions of the enterprise plus advanced feature set Automation Gateway, and Secure Provisioning for assigned nodes.

## Creating an Enrollment bundle

To create an Enrollment bundle in a Lighthouse instance:

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



- Click the + button. **The Enrollment Bundle Details** page displays.

ENROLLMENT BUNDLES

ENROLLMENT BUNDLE DETAILS

Name ⓘ

Token ⓘ

☐ Auto-approve node ⓘ

Subscription Selection ⓘ

Lighthouse Enterprise

Lighthouse Enterprise Automation Edition

ENROLLMENT BUNDLE NODE TAGS

Select tag ▼

Select value ▼

+

+ Add Tag

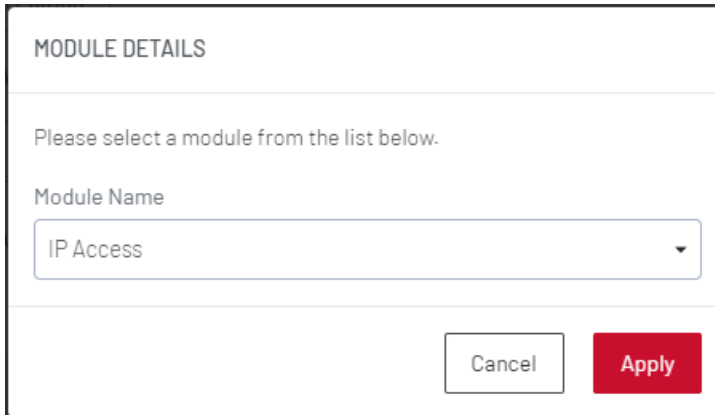
Create new tag value

TEMPLATES

ORDER	TEMPLATE TYPE	TEMPLATE NAME	ACTIONS
-------	---------------	---------------	---------

- Enter a **Name** and **Authentication Token** for the bundle in the respective fields.
- Select the number of **Tags** and **Values** to apply to any nodes that enroll using this Enrollment bundle.
- (Optional) Select the **Auto-approve** node checkbox.  
 When this is checked, a device configured using this Enrollment bundle is not placed in pending mode during the Enrollment process. Instead, it is automatically approved for Enrollment after it has been identified.

6. Select the desired **Subscription Bundle Label** - Lighthouse Enterprise Edition or Lighthouse Automation Edition. This is the desired subscription type to assign to newly enrolled nodes using this bundle (optional when more than one subscription is active on the Lighthouse).
7. You can also use this bundle to automatically activate NetOps modules for any supported nodes. Scroll down to the **NETOPS MODULES** section and press the + button to open the **MODULE DETAILS** dialog.



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

With the Enrollment bundle named, use the **Enrollment BUNDLE NODE TAGS** to populate it with the desired name-value pairs:

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

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

22.11 - REVISION 1	Managing Lighthouse Nodes	164
--------------------	---------------------------	-----

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

The templates in the table can be reordered using the arrow buttons in the far-left column of the table and are executed in the order they appear. The order buttons appear if there is more than one template in the table.

Template push operations stop if one template fails.

## Structure of an Enrollment bundle

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

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

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

```
address=192.168.88.20
api_port=4443
bundle=bne-dc
password=secret
```

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

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

## Enrolling 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.



## Enrollment via Node Web UI

If a node is behind a firewall, Lighthouse cannot initiate an Enrollment. The node must be enrolled from the Node Web UI.

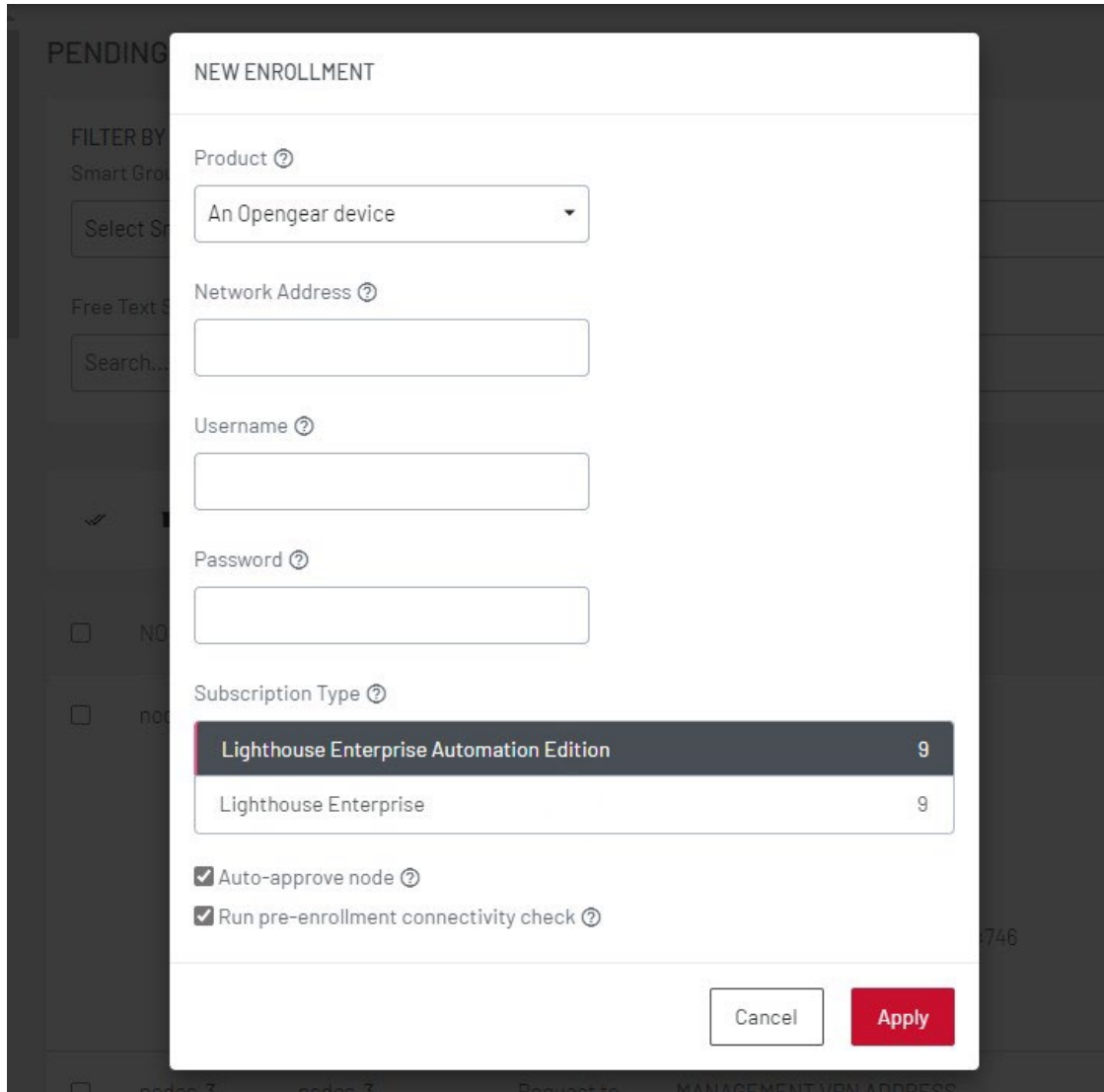
1. Log into the Node Web UI.
2. Select **Serial & Network > LIGHTHOUSE**.
3. Enter the **Server Address** of Lighthouse (which can be hostname, FQDN, or IP address)
4. Optionally, enter the **Server Port**.
5. Enter the **Enrollment Bundle** (if a specific bundle is being used), and the **Enrollment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

After enrollment, the node displays in the **Pending Nodes** page, with a **Status** of *Approval*. Click the Approve icon in the **Actions** column.

## Enrollment via Lighthouse Web UI

Enrollment via Lighthouse Web UI only works if the Node is reachable from Lighthouse. To enroll a node:

1. Select **Add Node** in the top menu bar to open a **NEW Enrollment** dialog.



**NEW ENROLLMENT**

Product ⓘ  
 An Opendgear device ▼

Network Address ⓘ

Username ⓘ

Password ⓘ

Subscription Type ⓘ

Lighthouse Enterprise Automation Edition	9
Lighthouse Enterprise	9

☒ Auto-approve node ⓘ  
☒ Run pre-enrollment connectivity check ⓘ

Cancel Apply

2. Select the **Product** type from the **Product** menu.

- . An Opendgear device
  - . A generic third-party device
  - . An Avocent ACS6000
  - . An Avocent ACS8000
  - . An Avocent ACS Classic
  - . A Cisco 2900 Series
  - . A Digi Passport
3. Enter the **Network Address**, **Username**, and **Password** of the node being enrolled. The **Username** and **Password** fields are for the login credentials required by the remote node being enrolled, not the login credentials used to login to the Lighthouse instance.
  4. Select the **Subscription Type**. Each type shows the number of available subscriptions.
    - . Enterprise Edition
    - . Automation Edition.
  5. If required, you can also select the **Auto-approve node** and **Run pre-enrollment connectivity checks**.
  6. Click **Apply**

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



## Enrolling nodes via OM, ACM, CM, and IM Web UI

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

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

### Mass Enrollment using ZTP

For mass node Enrolments 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.

## Enrollment via USB drive

An unconfigured device can be enrolled using a USB drive loaded with an Enrollment bundle. (See ["Creating an Enrollment bundle" on page 162.](#))

Download the Enrollment bundle, `manifest.og` from a Lighthouse instance as follows:

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

To enroll a node via USB drive:

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

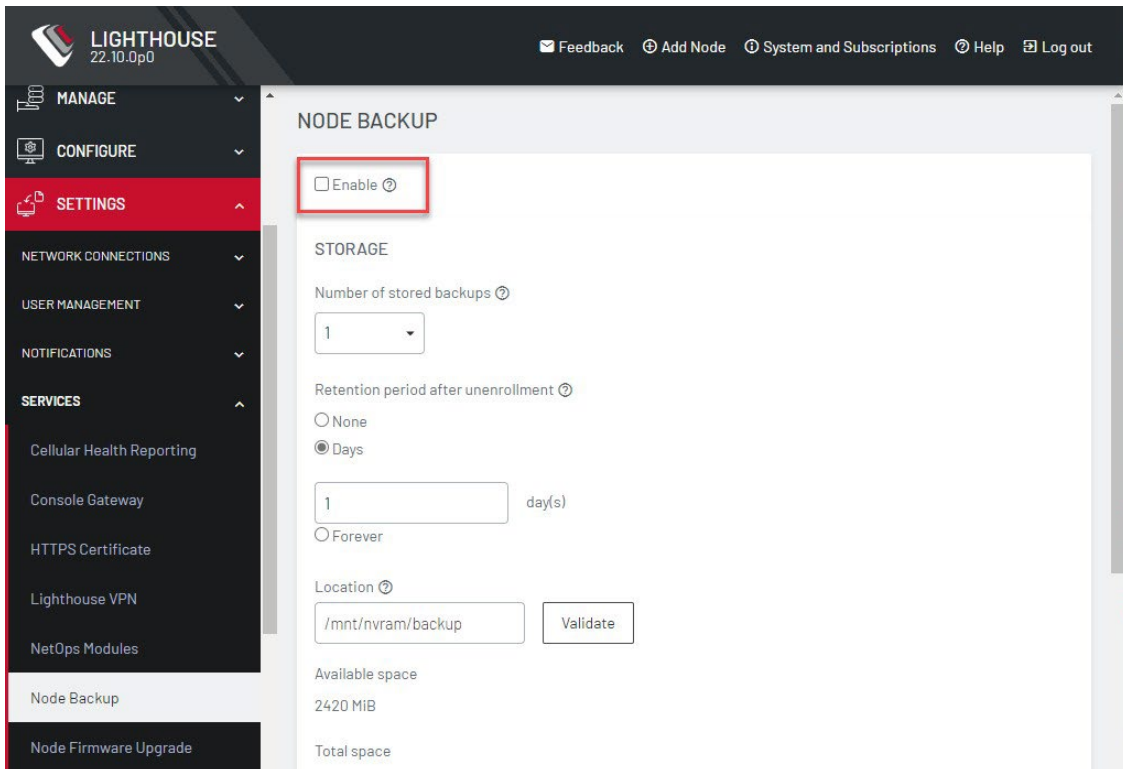
## Backing up Nodes

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

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

To set up node backup:

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



2. Select the **Enable** checkbox to turn on this service.
3. Under the **Storage** section, select the **Number of stored backups** you wish to keep.
4. In **Retention period after unEnrolment** select **None**, **Days** or **Forever**.

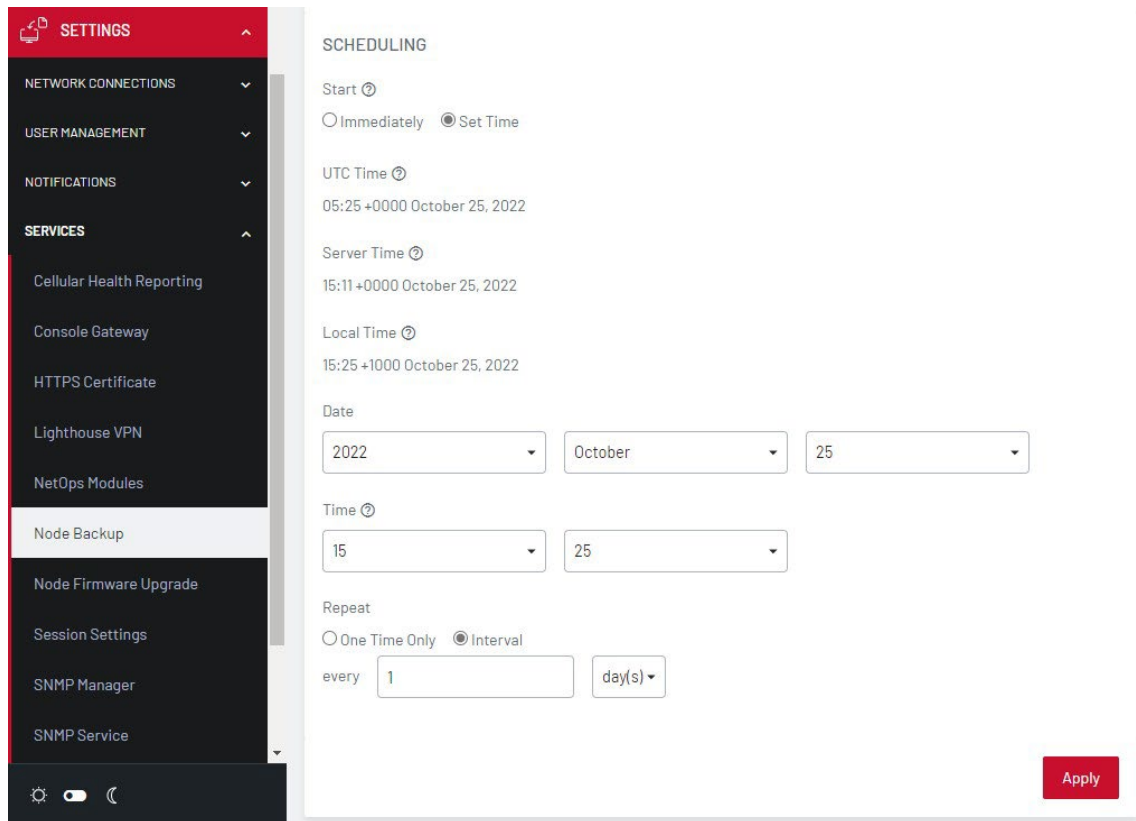


If you select **Days** enter the number of days.

5. Enter the **Location** you wish to store the backup files. We suggest you store backups at `/mnt/nvram/`.
6. Click **Validate** to make sure the location exists and has enough space to store them. The **Available space**, **Total Space** and **Estimated disk usage** MiBs display.
7. Click **Apply** or set a schedule using the **Scheduling** fields.

To setup an automated schedule for performing node backups:

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



**SETTINGS**

- NETWORK CONNECTIONS
- USER MANAGEMENT
- NOTIFICATIONS
- SERVICES**
  - Cellular Health Reporting
  - Console Gateway
  - HTTPS Certificate
  - Lighthouse VPN
  - NetOps Modules
  - Node Backup**
  - Node Firmware Upgrade
  - Session Settings
  - SNMP Manager
  - SNMP Service

**SCHEDULING**

Start ⓘ

☐ Immediately ☒ Set Time

UTC Time ⓘ

05:25 +0000 October 25, 2022

Server Time ⓘ

15:11 +0000 October 25, 2022

Local Time ⓘ

15:25 +1000 October 25, 2022

Date

2022 October 25

Time ⓘ

15 25

Repeat

☐ One Time Only ☒ Interval

every 1 day(s)

**Apply**

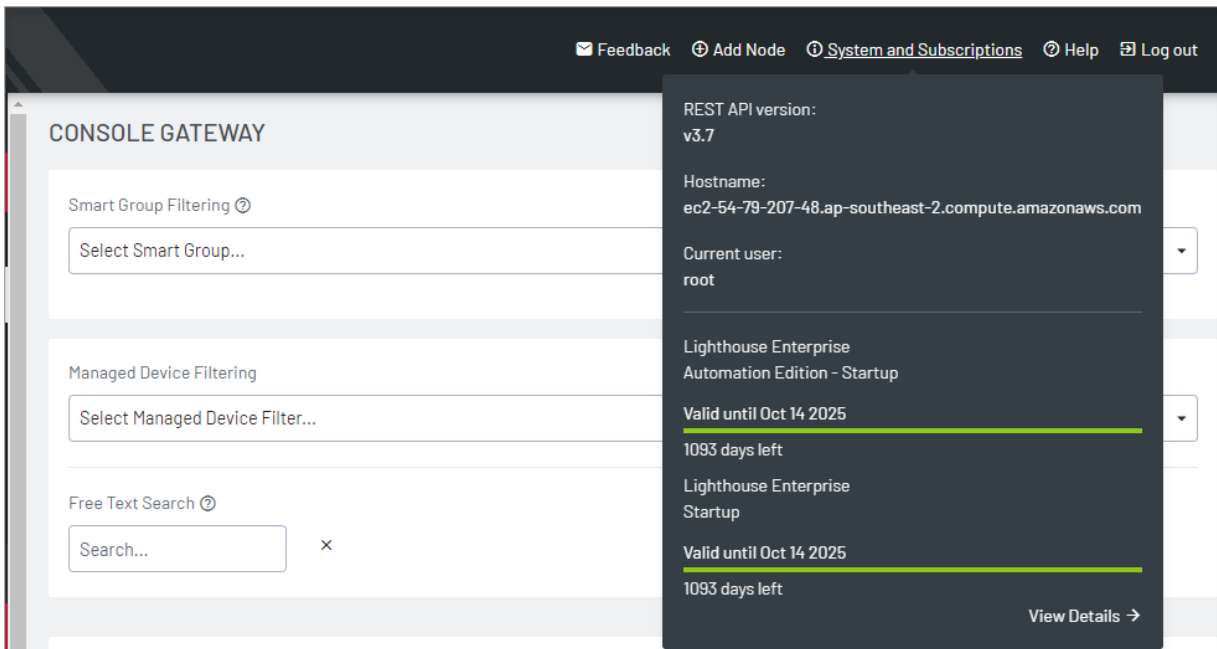
2. For the **Start** time, choose either **Immediately** or choose **Set Time** to open editable **Date** and **Time** fields.

3. Choose how often you wish to **Repeat** the backup by selecting either **One Time Only** or **Interval**.
4. If you selected **Interval**, enter the values for **Days** or **Weeks**.

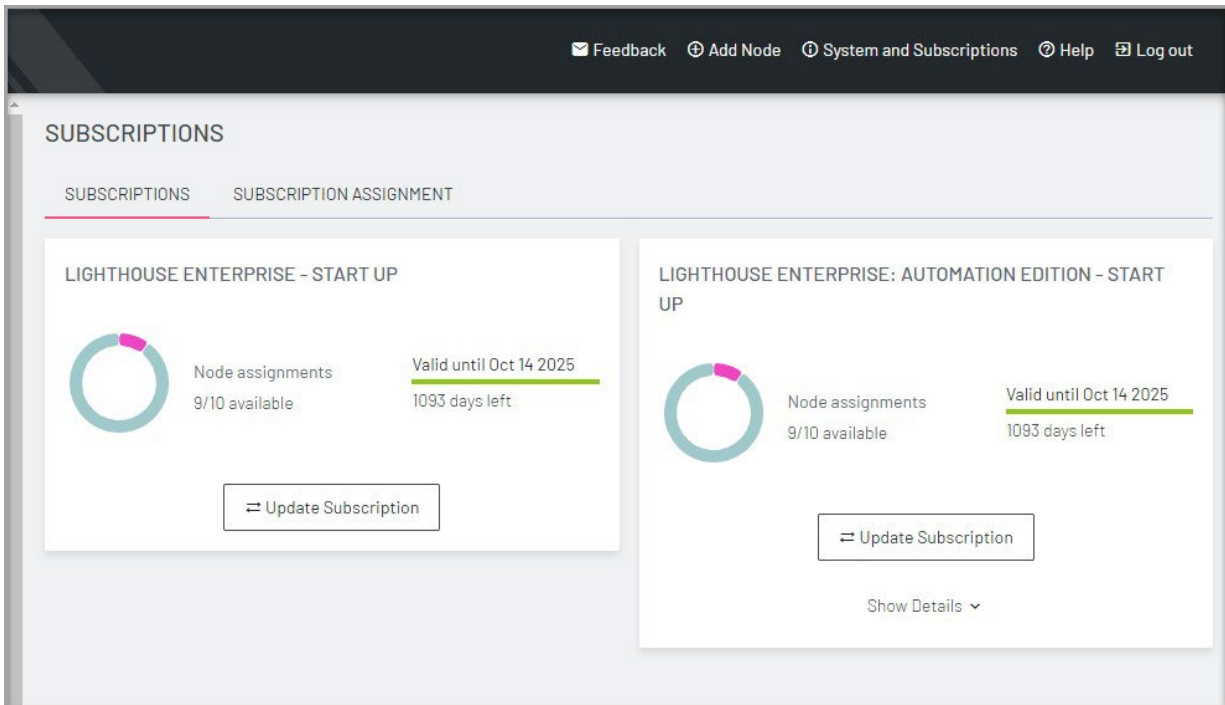
**Note:** You can modify these options by returning to **SETTINGS > SERVICES > Node Backup** at any time.

## Subscribing to Lighthouse

Lighthouse has a flexible, simplified subscription model that allows you to add extended functionality if required. The **Systems and Subscriptions** option on the main menu provides information on your subscriptions including the type of subscriptions, the validity and the number of days before expiry.



Click **View Details**. The **Subscriptions** page displays.



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 NetOps features - Automation Gateway and 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.

For more information on subscription tiers, see ["Enrollment Bundles"](#) on page 160.

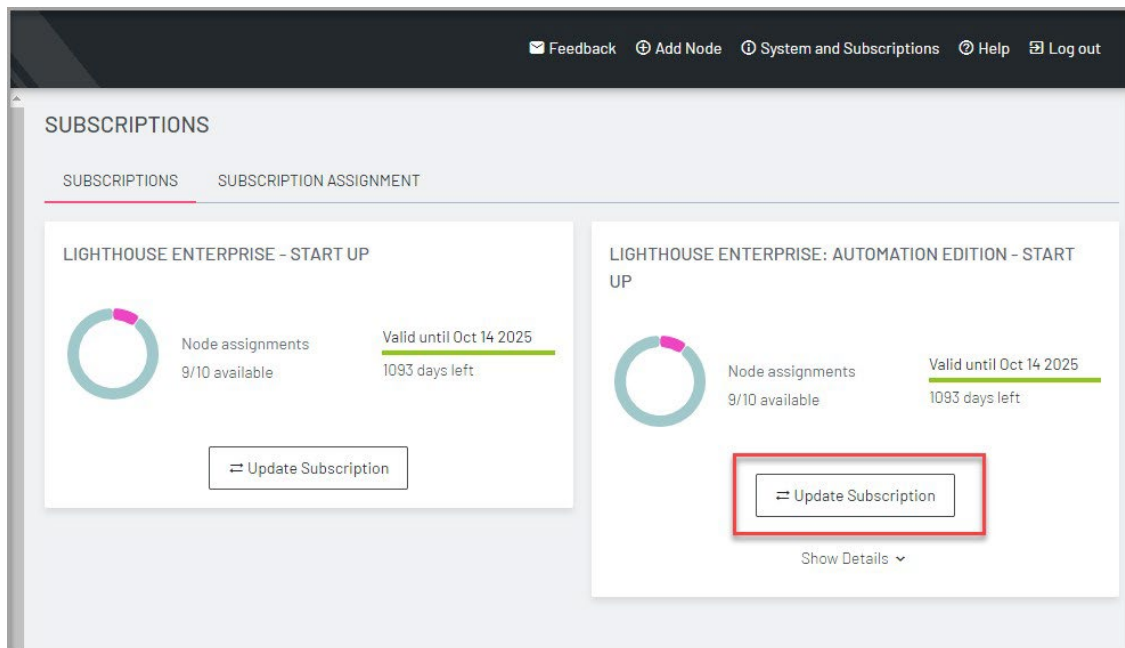
## Adding a new subscription to Lighthouse

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

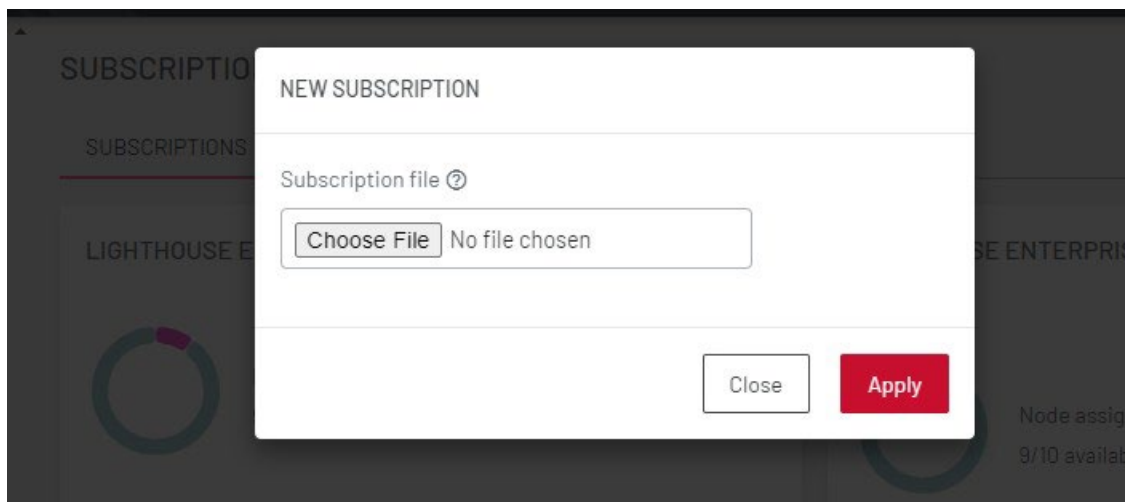
**Note:** Contact [sales@opengear.com](mailto:sales@opengear.com) with a request for the type of subscription, including the number of nodes you wish to apply the subscription to. You will receive an encrypted .zip file named ***subscription.zip***.

## To add a subscription

1. Select **Systems > Settings > Subscriptions**. The **Subscriptions** page displays.
2. Click **Update Subscription**.



3. The **New Subscription** dialog displays. Choose ***subscription.zip***, and click **Apply**.



4. The subscription will be applied to the appropriate nodes in the Lighthouse.

## Assigning subscriptions to nodes

Lighthouse has a flexible, simplified subscription model that allows users to mix and match functionality where required. Users can reassign nodes to different subscriptions in order to optimize Automation Edition features on their network. For example, if the user:

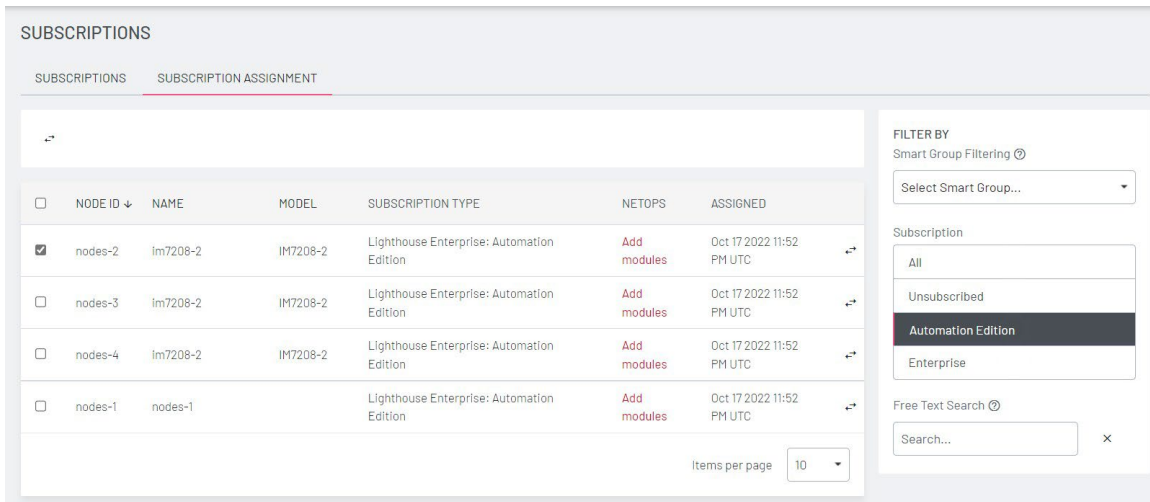
- Previously had two license tiers, but only paid to renew one license tier
- Purchased an AE license for the first time, and wants the extra functionality on some of their nodes
- Wants to swap which nodes have access to the extra functionality available in an AE license



## View Subscriptions in the Lighthouse UI

To see subscriptions:

1. Select **SETTINGS > Systems > Subscriptions**.
2. The Subscriptions page displays. Select **Subscription Assignment**.



The screenshot shows the 'SUBSCRIPTIONS' page with the 'SUBSCRIPTION ASSIGNMENT' tab selected. The table lists subscriptions with columns for selection, node ID, name, model, subscription type, netops, and assigned date. The first row is selected, and the 'Add modules' link is visible. The right sidebar shows filter options for 'Subscription' (All, Unsubscribed, Automation Edition, Enterprise) and a 'Free Text Search' field.

	NODE ID	NAME	MODEL	SUBSCRIPTION TYPE	NETOPS	ASSIGNED
<input checked="" type="checkbox"/>	nodes-2	lm7208-2	IM7208-2	Lighthouse Enterprise: Automation Edition	<a href="#">Add modules</a>	Oct 17 2022 11:52 PM UTC
<input type="checkbox"/>	nodes-3	lm7208-2	IM7208-2	Lighthouse Enterprise: Automation Edition	<a href="#">Add modules</a>	Oct 17 2022 11:52 PM UTC
<input type="checkbox"/>	nodes-4	lm7208-2	IM7208-2	Lighthouse Enterprise: Automation Edition	<a href="#">Add modules</a>	Oct 17 2022 11:52 PM UTC
<input type="checkbox"/>	nodes-1	nodes-1		Lighthouse Enterprise: Automation Edition	<a href="#">Add modules</a>	Oct 17 2022 11:52 PM UTC

Items per page: 10

**FILTER BY**  
 Smart Group Filtering ⓘ  
 Select Smart Group...  
 Subscription  
 All  
 Unsubscribed  
**Automation Edition**  
 Enterprise  
 Free Text Search ⓘ  
 Search... X

The Subscriptions page displays details of the subscriptions and allows you to search using smart group filters, or free text searches.

The page also allows you to update, replace or add new subscriptions


## To assign nodes to subscriptions

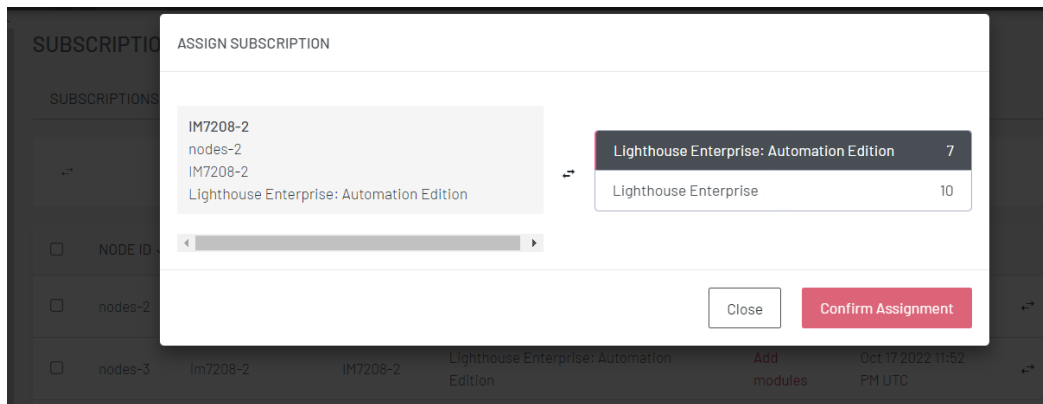
Select **SETTINGS > System > Subscriptions > Subscription Assignments**.


The ***Subscriptions Assignments*** page displays details of the nodes including:

- . The Node name
- . The Subscription Type
- . Current NetOps module deployment to assigned nodes
- . Date of Assignment


## To assign a subscription

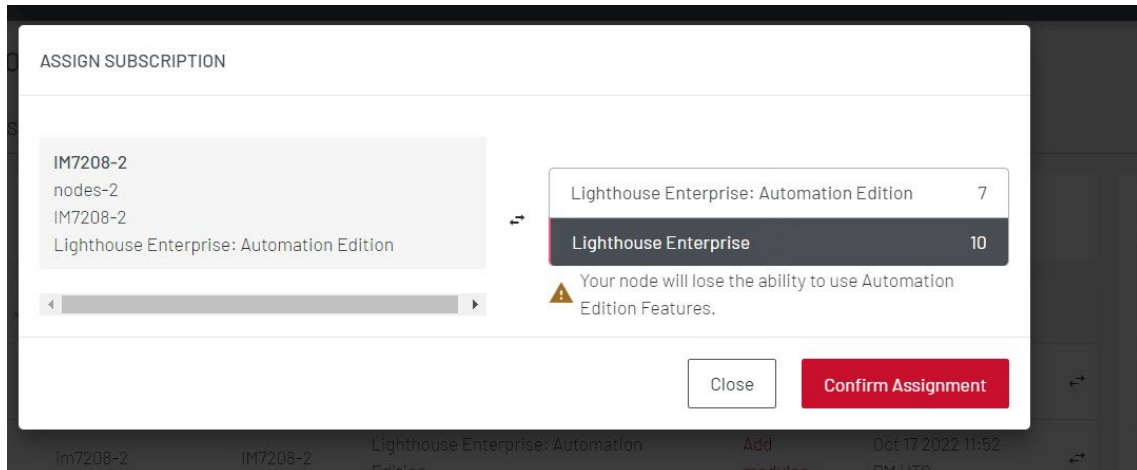
1. Select **SETTINGS > System > Subscriptions > Subscription Assignment**
2. Display the required nodes using the Filter pane, either by using the **Filters** or **Free text** options.
3. To assign a subscription to a node click the  **Assign** button. The **Assign Subscription** dialog displays.



4. Select the required subscription from the leftmost panel using the  **Assign** button.
5. Click **Confirm Assignment**

## To change a node's subscription

1. Select **SETTINGS > Systems > Subscriptions > Subscription Assignment**
2. Display the required nodes using the Filter pane, either by using the **Labels** or **Free text** options.
3. To change subscription on a node click the  **Assign** button on the row. The **Assign Subscription** dialog displays.
4. Select the required subscription from the right panel.
5. Click **Confirm Assignment** to change the node's subscription. A message box displays indicating the effect of changing the subscription.



**Note:** Another way of removing a node from a subscription is to unenrol the node. The subscription will then become available to be assigned to another node.


## Work with Nodes


After a node has been enrolled, you can connect to it directly, either to monitor it, or to run commands on it. Multiple nodes can be selected.

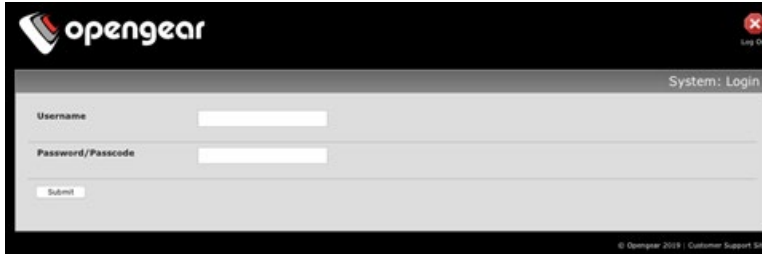
## Connect to the web-management interface of a node

After a node has been enrolled, you can connect to the enrolled node's web-management interface as follows:

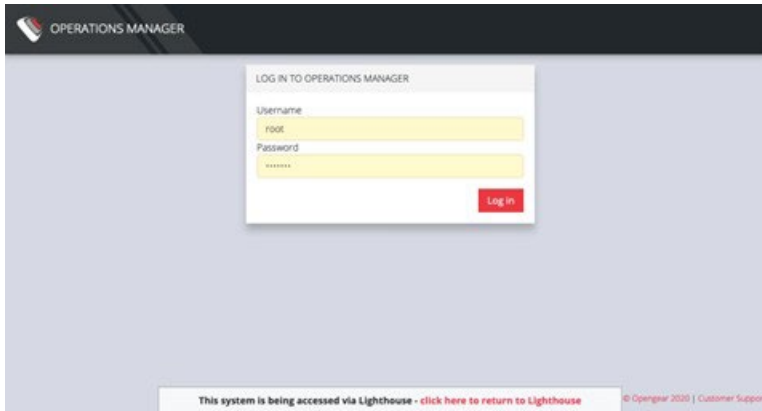
1. Select **MANAGE > NODES > Node Web UI**. The nodes display:

NODE ID	NAME	STATUS	CELL HEALTH	INFO
nodes-5	SPT-OM2248	Connected Last status change 2 weeks ago	 Cellular Interface Disabled Last status change 2 weeks ago	MANAGEMENT VPN ADDRESS 192.168.128.5 NET1 MAC ADDRESS 00:13:c6:08:07:7c NET1:STATIC1 192.168.0.1 NET1:DHCP 10.220.3.6 NET1:IP6AUTO Unavailable WWAN0:DHCP Unavailable WWAN0:IP6AUTO Unavailable NET2:STATIC9 10.10.10.1 <a href="#">Show Details</a>

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



The screenshot shows the Opengear System Login interface. At the top left is the Opengear logo. At the top right is a 'Log Out' button with a red 'X' icon. Below the header is a 'System: Login' title. The main area contains two input fields: 'Username' and 'Password/Passcode'. Below these fields is a 'Submit' button. At the bottom right, there is a small copyright notice: '© Opengear 2019 | Customer Support Site'.



The screenshot shows the Opengear Operations Manager login interface. At the top left is the 'OPERATIONS MANAGER' title. The main area features a 'LOG IN TO OPERATIONS MANAGER' box. Inside this box are two input fields: 'Username' (with the value 'root') and 'Password' (with masked characters). Below these fields is a red 'Log in' button. At the bottom of the page, there is a message: 'This system is being accessed via Lighthouse - [click here to return to Lighthouse](#)'. To the right of this message is a small copyright notice: '© Opengear 2020 | Customer Support'.

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

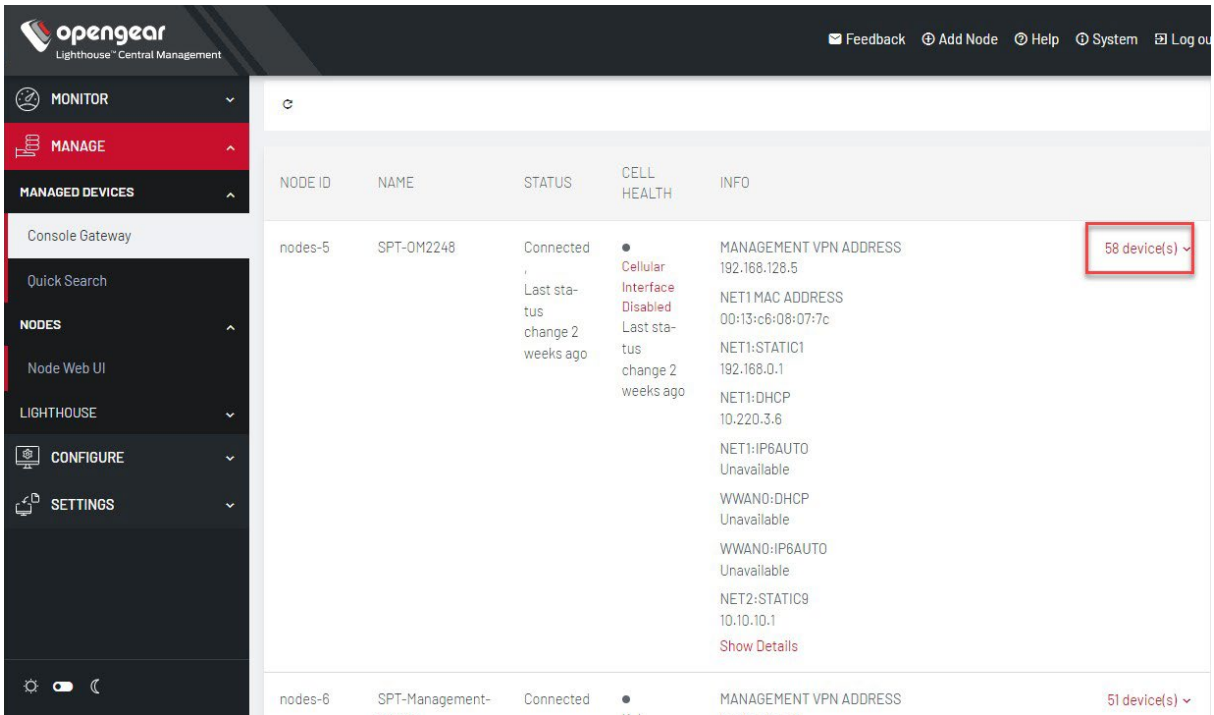
## Connecting to a node's serial ports

You can connect to the enrolled node's serial ports 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.



## Find the Serial Ports

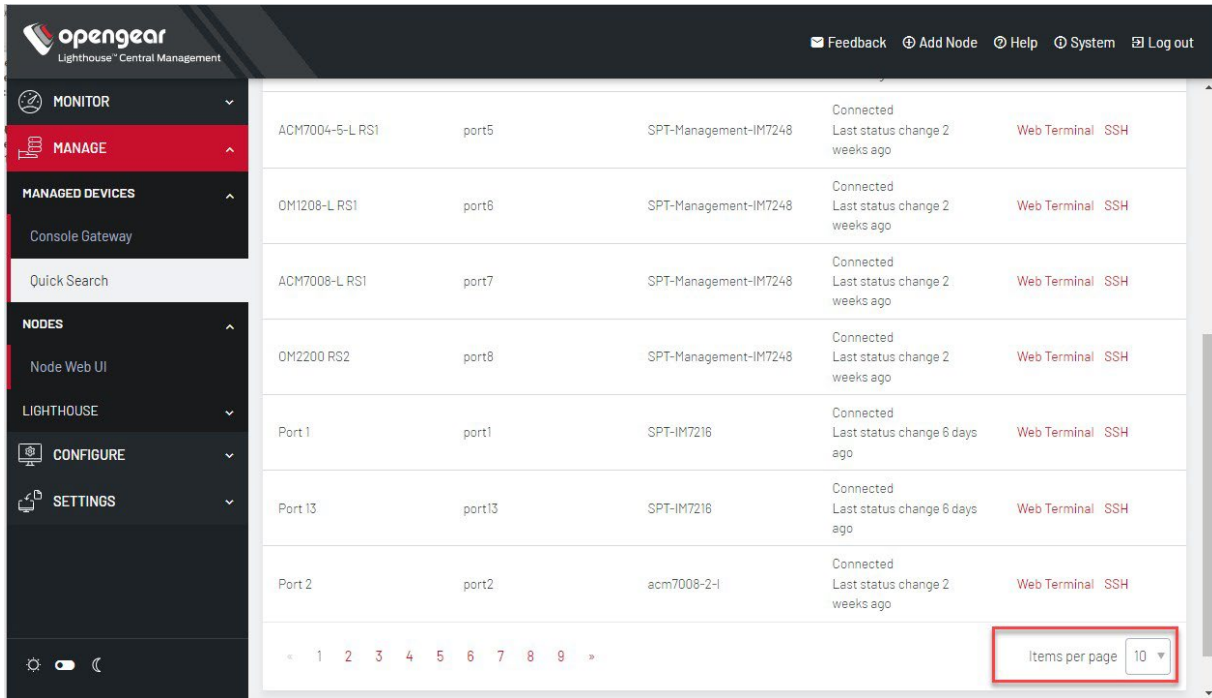
To search for serial ports on Lighthouse select **MANAGE > MANAGED DEVICES**  
**> Console Gateway**



NODE ID	NAME	STATUS	CELL HEALTH	INFO
nodes-5	SPT-OM2248	Connected Last status change 2 weeks ago	● Cellular Interface Disabled Last status change 2 weeks ago	MANAGEMENT VPN ADDRESS 192.168.128.5 NET1 MAC ADDRESS 00:13:c6:08:07:7c NET1:STATIC1 192.168.0.1 NET1:DHCP 10.220.3.6 NET1:IP6AUTO Unavailable WWAN0:DHCP Unavailable WWAN0:IP6AUTO Unavailable NET2:STATIC9 10.10.10.1 <a href="#">Show Details</a>
nodes-6	SPT-Management-123456	Connected	●	MANAGEMENT VPN ADDRESS 192.168.128.5 NET1 MAC ADDRESS 00:13:c6:08:07:7c NET1:STATIC1 192.168.0.1 NET1:DHCP 10.220.3.6 NET1:IP6AUTO Unavailable WWAN0:DHCP Unavailable WWAN0:IP6AUTO Unavailable NET2:STATIC9 10.10.10.1 <a href="#">Show Details</a>

or

Select **MANAGE > MANAGED DEVICES > Quick Search** A page displays all the connected nodes, the right most column displays the number of devices connected to the node.



Node ID	Port	SPT	Status	Last Status Change	Actions
ACM7004-5-L RS1	port5	SPT-Management-IM7248	Connected	Last status change 2 weeks ago	Web Terminal SSH
OM1208-L RS1	port6	SPT-Management-IM7248	Connected	Last status change 2 weeks ago	Web Terminal SSH
ACM7008-L RS1	port7	SPT-Management-IM7248	Connected	Last status change 2 weeks ago	Web Terminal SSH
OM2200 RS2	port8	SPT-Management-IM7248	Connected	Last status change 2 weeks ago	Web Terminal SSH
Port 1	port1	SPT-IM7216	Connected	Last status change 6 days ago	Web Terminal SSH
Port 13	port13	SPT-IM7216	Connected	Last status change 6 days ago	Web Terminal SSH
Port 2	port2	acm7008-2-1	Connected	Last status change 2 weeks ago	Web Terminal SSH

Items per page: 10

The **Items per page** drop-down on **Quick Search** page allows the user to select the number of ports per page. This setting applies to the current user session only and will be lost when user logs out.

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

## Node-centric searching

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

## Port-centric searching

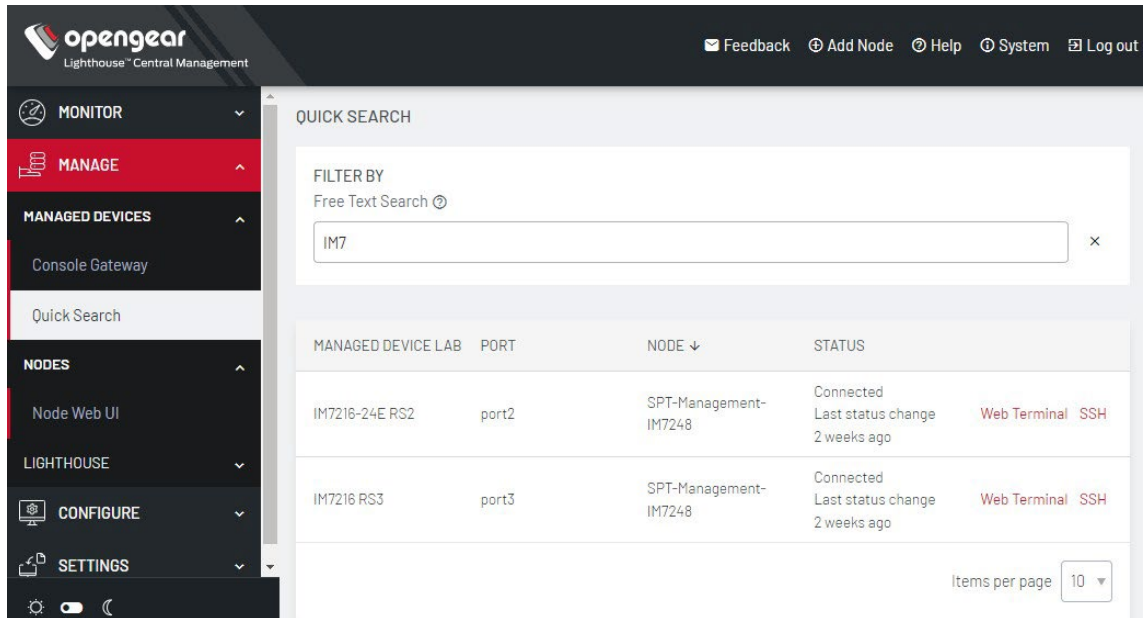
1. Select **MANAGE > MANAGED DEVICES > Console Gateway**.
2. Find the particular port by using the **Managed Device Filtering** options to restrict the listed managed devices within enrolled nodes.

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

- . Web Terminal
- . SSH

## Quick Search

1. Select **MANAGE > MANAGED DEVICES > Quick Search**.
2. Enter the managed device label, aka name, in the **Quick Managed Device Search** field. This search live-updates as user type.



The screenshot shows the Lighthouse Central Management interface. The left sidebar has a menu with 'MANAGE' selected, which is expanded to show 'MANAGED DEVICES' and 'Quick Search'. The main content area is titled 'QUICK SEARCH' and contains a 'FILTER BY' section with a 'Free Text Search' input field containing 'IM7'. Below the search field is a table with the following data:

MANAGED DEVICE LAB	PORT	NODE ↓	STATUS
IM7216-24E RS2	port2	SPT-Management-IM7248	Connected Last status change 2 weeks ago <a href="#">Web Terminal</a> <a href="#">SSH</a>
IM7216 RS3	port3	SPT-Management-IM7248	Connected Last status change 2 weeks ago <a href="#">Web Terminal</a> <a href="#">SSH</a>

At the bottom right of the table, there is a 'Items per page' dropdown menu set to '10'.

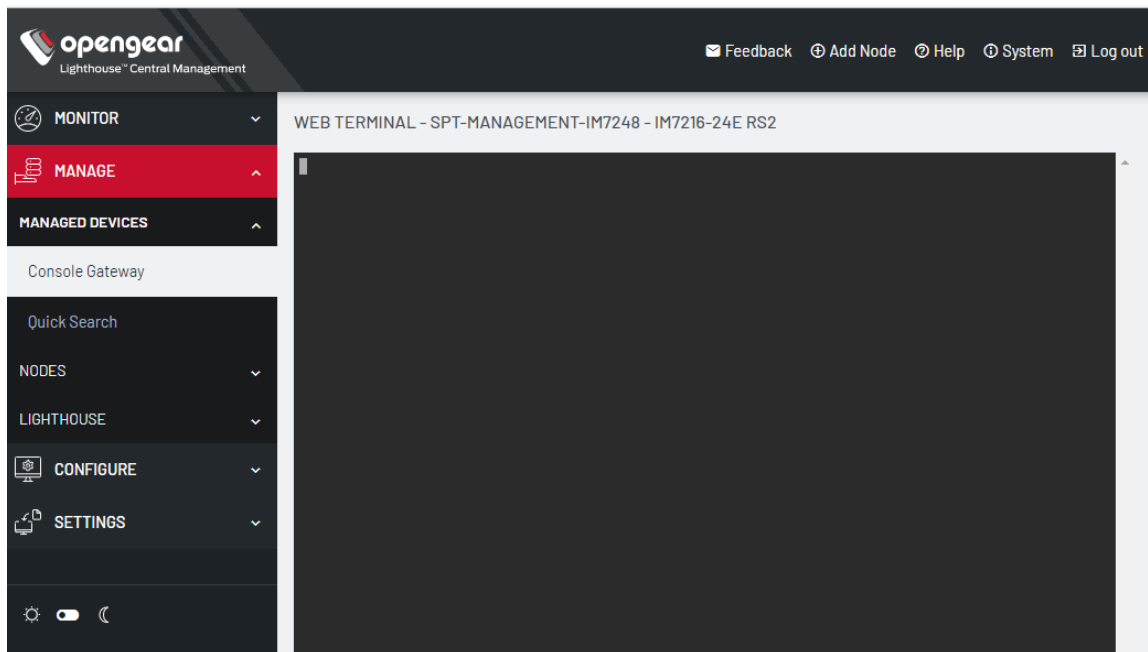
3. Use **Web Terminal** and/or **SSH** links on a particular port to access it.

## Access via HTML5 Web Terminal

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

To access a console port via the Web Terminal:

1. Locate the particular port by using one of the search techniques described in ["Connecting to a node's serial ports" on page 191](#).
2. Click the **Web Terminal** link for the particular port. A new tab opens containing the Web Terminal.



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

## Access a serial port via SSH

To access ports via SSH, use a direct SSH link from the Web UI to connect to the port.

To access a console port via a Direct SSH link:

1. Locate the particular port by using one of the search techniques discussed in ["Connecting to a node's serial ports" on page 191..](#)
2. Click the **SSH** link to connect to the URL.

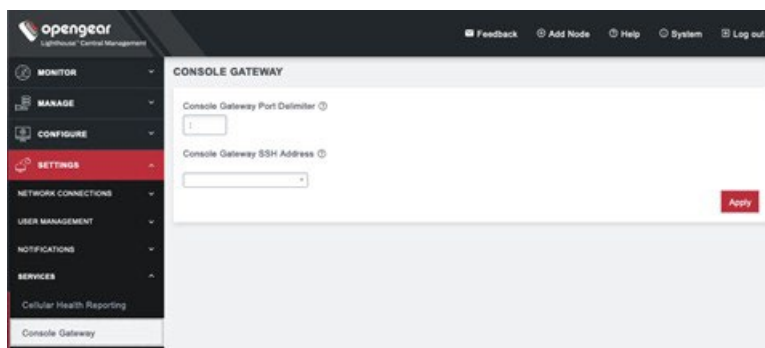
These auto-generated links use the colon (:) as the field-delimiter. The auto-generated SSH link has the following form:

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

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

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

1. Select **SETTINGS > SERVICES > Console Gateway**



2. Enter a delimited character in the **Console Gateway Port Delimiter** text-entry field. The carat, ^, is the most common alternative.

3. Use the **Console Gateway SSH Address** drop-down menu to choose an address from which to SSH. The list of available addresses contains the current network interfaces and external network addresses. The value defaults to `net1:dhcp` if it exists and `net1:static` otherwise. The additional external addresses can be added to this list using the **SETTINGS > SYSTEM > Administration** page.

To use the console chooser menu, SSH to the Lighthouse appliance with the username format `username:serial`. This connects to the Lighthouse and presents a list of nodes that the user can access. Once the user selects a node, they are presented with a list of console ports they have access to. When one is selected, the user is connected to that port. For faster access, there are username format shortcuts that give more specific lists of serial ports, or direct access without a menu.

- **username:node\_name**

When a valid node name is specified, a list of console ports that the user can access on that node is shown. If they do not have access to this node, the connection fails.

- **username:node\_name:port\_name**

When a valid node name and port name are specified, and the user has access to that node and port, the user is connected to this port. If they do not have access to that port, the connection fails.

- **username:port\_name**

When a valid port name is specified, the user is connected to first port with that port name found. If the user does not have access to this port, the connection fails.

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

## Example Console Gateway session

```
$ ssh adminuser:serial@lighthouse-name-or-ip-here  
1: cm7lxx  
Connect to remote > 1  
1: Cisco Console 2: Port 2  
Connect to port > 1  
router#
```



## Selecting nodes using shell-based tools

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

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

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

To Select all nodes

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

To running commands on selected nodes

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

**Note:** All commands are run as root.

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

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

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.

## Node organization and filtering

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

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

## Filter Nodes

Filters enable you to quickly find a particular node or set of nodes, The filters are:

- **Free Text Search**
- **Smart Group Filtering**, and
- **Managed Device Filtering**.

## Creating Smart Groups

**Smart Groups** are used within Lighthouse to group related nodes.

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

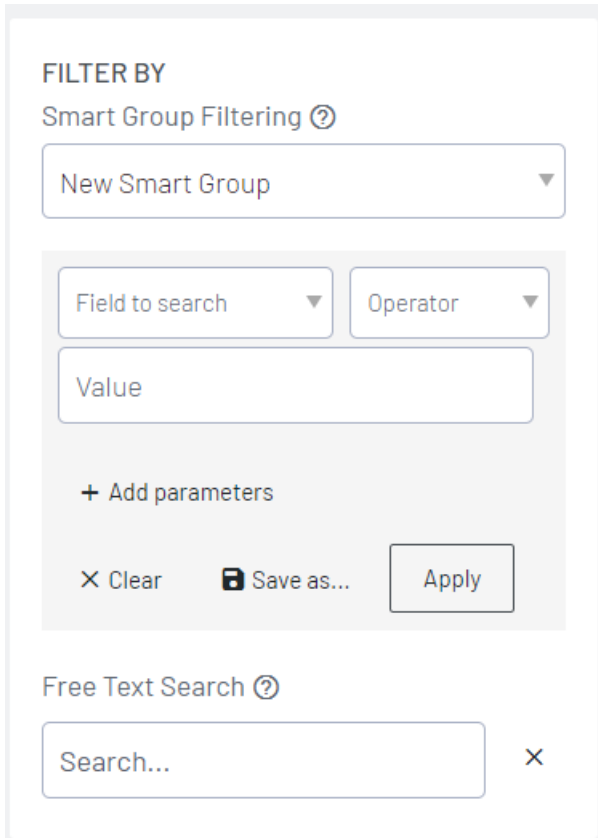
Smart Groups are dynamic, when more nodes are added to the system, the filters update automatically.

A user group can be linked to a particular **Smart Group**. When a user group is linked to a smart group in this fashion, members of the group inherit rights over all nodes in the smart group based on the group's role. "[Modifying existing groups](#)" on [page 276](#) for how to set a group's role and linked Smart Group.

To create a Smart Group:

1. Navigate to any page which displays the Smart Group search interface, for example **CONFIGURE > NODE Enrollment > Enrolled Nodes** or **MANAGE > NODES > Node Web UI**.
2. Move to the **Smart Group Filtering** and select **New Smart Group** from the displayed list

- Click the **Field to search** to select a node attribute to filter on



**FILTER BY**  
 Smart Group Filtering ?

New Smart Group ▼

Field to search ▼ Operator ▼

Value

+ Add parameters

X Clear Save as... Apply

Free Text Search ?

Search... X

Node attributes include details about the device (**Internal VPN Address, MAC Address, Name, Product, SSH Port, Firmware Version, Model, Serial Number, Node ID, Connection Status, Cell Health**), and include any **tags** that have been configured in the system. For filtering access to devices, tags are the most useful attributes to filter on. When a tag is selected, the **Field to search** text box contains tag values.

- Select the **Operator** to apply to the Value.
- Select the **Value** to be matched against.
- To enter multiple search parameters, click - **Add parameters**
- The AND OR buttons display

- > Select AND if the filter must match all values.
- > 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:** Note: Click the - button to remove the additional Search parameters if necessary.

9. Click **Apply** to see the results of the filter.

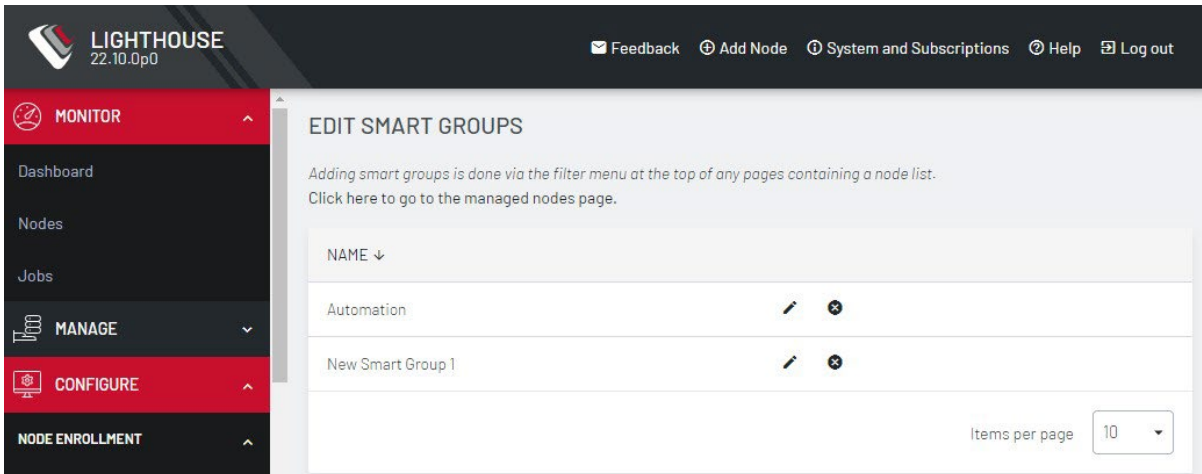
10. Click **Save As** and type in a name for your new Smart Group.

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

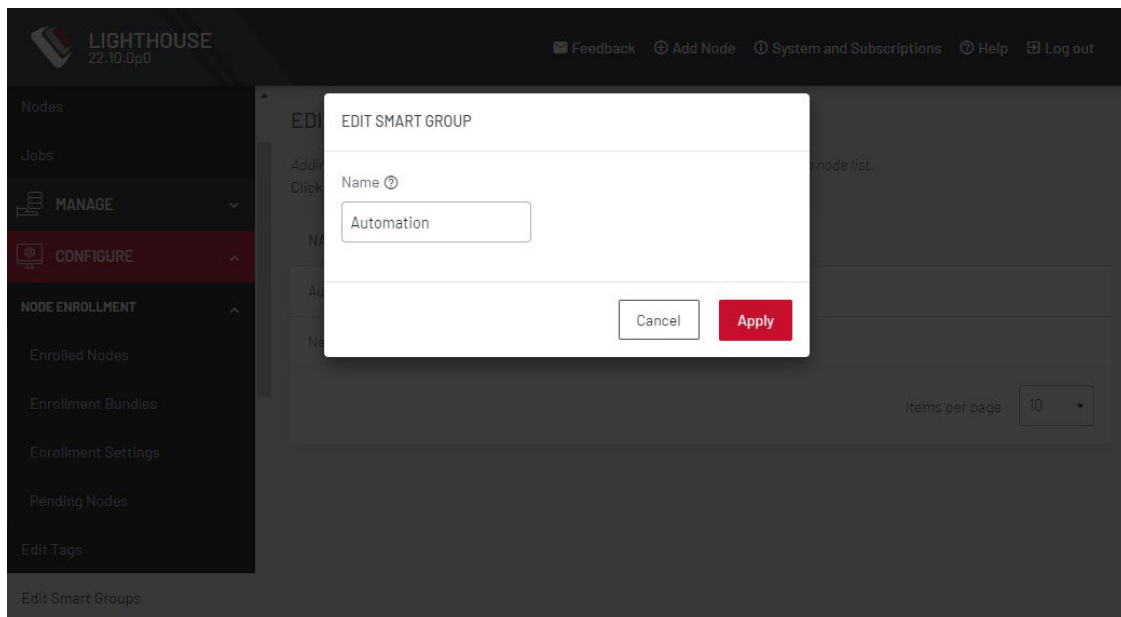
## Editing a Smart Group

To edit a Smart Group:

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



2. Click **Delete** to delete an existing Smart Group.
3. Click **Edit Group** button to change a Smart Group's name.





**To change the search parameters used by a Smart Group:**

1. Navigate to a page that displays Smart Groups for filtering (for example, **CONFIGURE > NODE Enrollment > Enrolled Nodes**).
2. Select the required Smart Group to be changed from the **Select Smart Group** drop-down menu.
3. Change the **Tag** and **Operator** values as required.
4. Click **Save as**.
5. Retain the Smart Group **Name** and click **Apply**. The changed search parameters overwrite the parameters of the ' Smart Group.

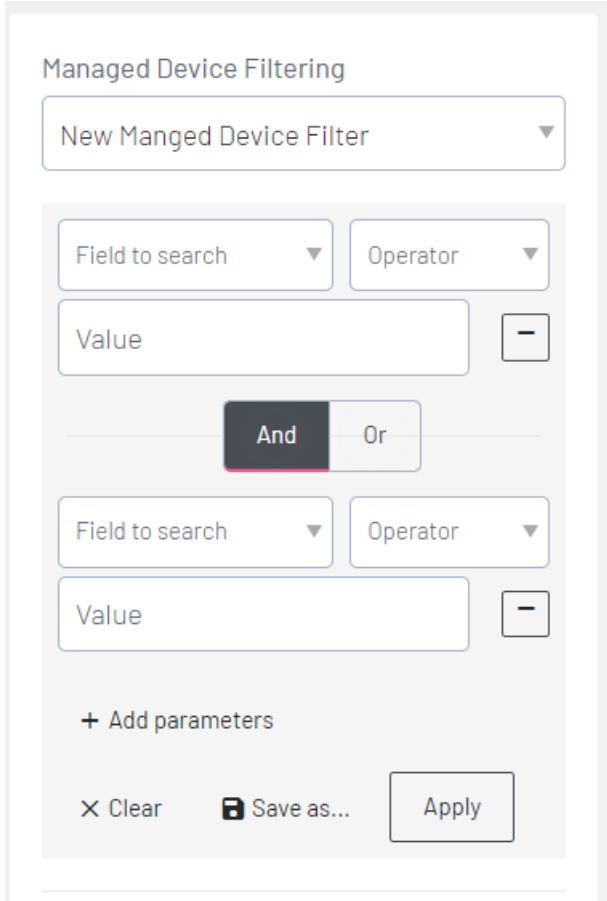
## Creating Managed Device Filters

Use **Managed Device Filters** to group managed devices on nodes. The **MANAGE > MANAGED DEVICES > Console Gateway** page makes it easier to find a particular console.

Managed Device Filters are dynamically updated when more nodes with managed devices are added. Managed Filters can be filtered by multiple parameters using either the AND and OR boolean operators.

To create a Managed Device Filter:

1. Select the **MANAGE > MANAGED DEVICES > Console Gateway** page.
2. On *the Managed Device Filtering pane* and select **New Managed Device Filter**.



The screenshot shows the 'Managed Device Filtering' interface. At the top, there's a title 'Managed Device Filtering' and a dropdown menu set to 'New Manged Device Filter'. Below this, there are two identical filter rule sections. Each section contains a 'Field to search' dropdown, an 'Operator' dropdown, and a 'Value' text input. Between the two sections, there are 'And' and 'Or' buttons. At the bottom, there are three buttons: '+ Add parameters', 'X Clear', and 'Save as...' (with a floppy disk icon), and an 'Apply' button.

3. Click **Field to search** to select a node attribute to filter on.
  - . Select **Port Label** configuration.
4. Click **Operator** to select the operator to apply to the Value.
5. Enter the **Value** to be matched against.
6. To enter multiple search parameters, click - **Add parameters**
7. The AND OR buttons display

- . Select AND if the filter must match all values.
  - . Select OR if one or the other values must match.
8. Enter the additional details in the Field to search, Operator and Value fields.

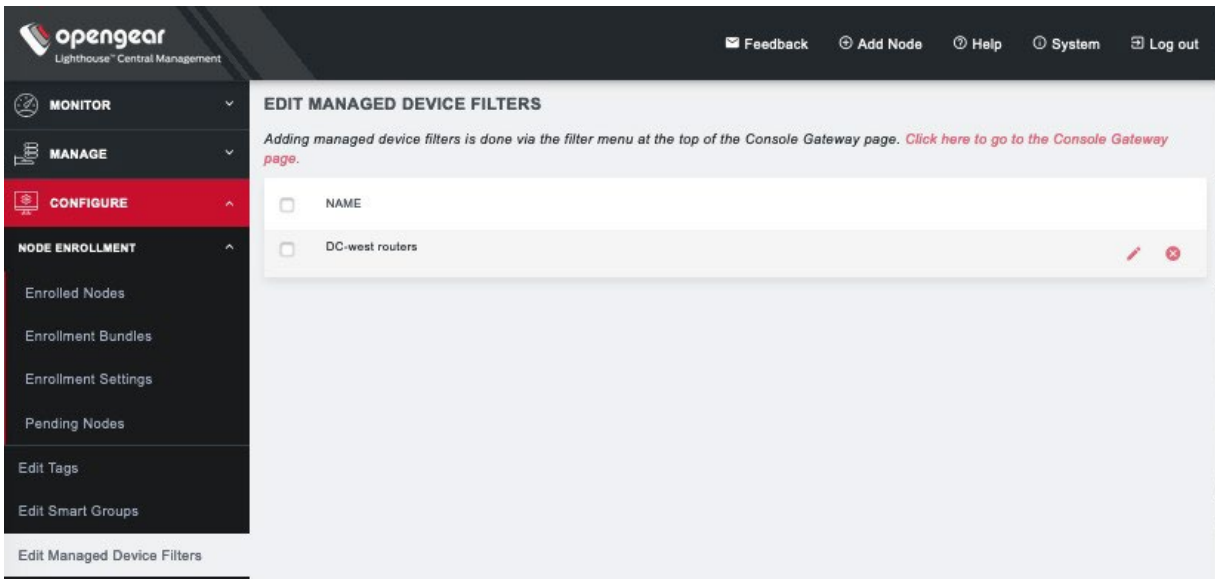
**Note:** Note: Click the - button to remove the additional Search parameters if necessary.

9. Click **Apply** to see the results of the filter.
10. Click **Save As** and enter a name for the filter.

The new **Managed Device Filter** can now be used for filtering nodes with managed devices.

## Editing an existing Managed Device Filter

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



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

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

1. Navigate to a page that displays the Managed Device Filters, such as **MANAGE > MANAGED DEVICES > Console Gateway**.
2. Select the required Managed Device Filter from the **Select Managed Device Filter** list.
3. Change the parameters (for example, **Operator** values) as required.
4. Do not change the name of the Managed Device Filter.
5. Click **Save as**.

- Click **Apply**. The modified **Managed Device Filter** overwrites the existing Managed Device Filter.

EDIT MANAGED DEVICE FILTER

Name ⓘ

DC-west routers

Cancel

Apply

**Note:** Do not change the name of the Managed Device Filter.

## Upgrade Nodes via the UI

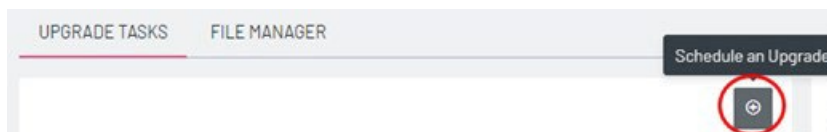
When you need to upgrade nodes to the latest firmware, for example, for security fixes, use the Node Upgrade UI to upgrade up to 5000 connected nodes per task. The Node Upgrade UI allows you to upgrade nodes either immediately or at a scheduled time, even outside normal business hours.

The Node Upgrade UI is available in the Web UI under **Settings > Services > Node 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.

Upon opening the Node Firmware Upgrade window there are two tabs accessible, plus access to the upgrade scheduling wizard, these are:

- Upgrade Tasks – a filtered dashboard where you can view scheduled 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, you need access to firmware files.

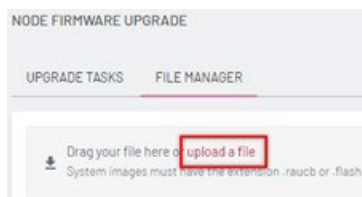


## Upload a firmware file

The Node Upgrade UI is available in the Web UI under **Settings > Services > Node Firmware Upgrade**. Use the Upload tool on the File Manager tab to upload one file at a time.

**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 will be uploaded.

1. Select the 'File Manager' tab in the Node Firmware Upgrade UI section.
2. Either select 'upload a file' to open an explorer view, then select the file. Or
3. Drag and drop the file into the upload area.



In-progress uploads can be cancelled by clicking the Close button (X).

The file upload will continue even if you click elsewhere in the Lighthouse UI, however, the upload will be cancelled if you close the website or if the HTTPS connection to Lighthouse is closed.

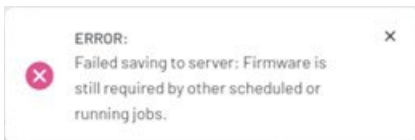
## Delete a firmware file

The Node Upgrade UI is available in the Web UI under **Settings > Services > Node Firmware Upgrade**. Use the Delete tool on the File Manager tab to delete one file at a time.

To delete a firmware file that is no longer needed:

1. Select the **File Manager** tab on the **Node Firmware Upgrade UI** page.
2. Click the delete button (■) next to the firmware file you wish to delete.

**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.



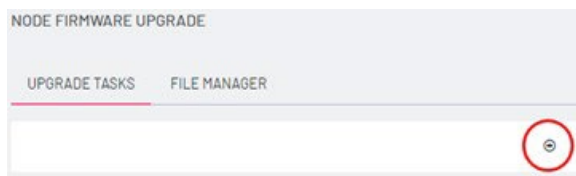
## Node Upgrade Tasks

To upgrade a node you will need to set up a task.

## Create an upgrade task

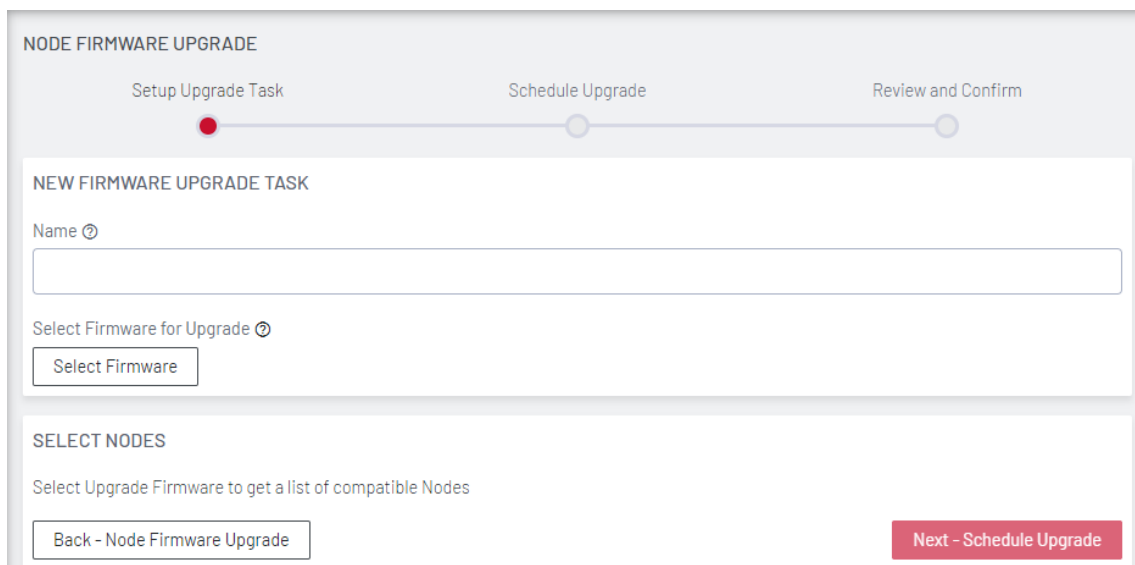
The Node Upgrade UI is available in the Web UI under **Settings > Services > Node Firmware Upgrade** Use the Upgrade tasks tab to

- 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 an Upgrade button (+) above the task information table to schedule an upgrade (or start one immediately).
2. The **Node Firmware Upgrade** wizard displays. Enter a name/title for the upgrade task.



NODE FIRMWARE UPGRADE

Setup Upgrade Task      Schedule Upgrade      Review and Confirm

NEW FIRMWARE UPGRADE TASK

Name ⓘ

Select Firmware for Upgrade ⓘ

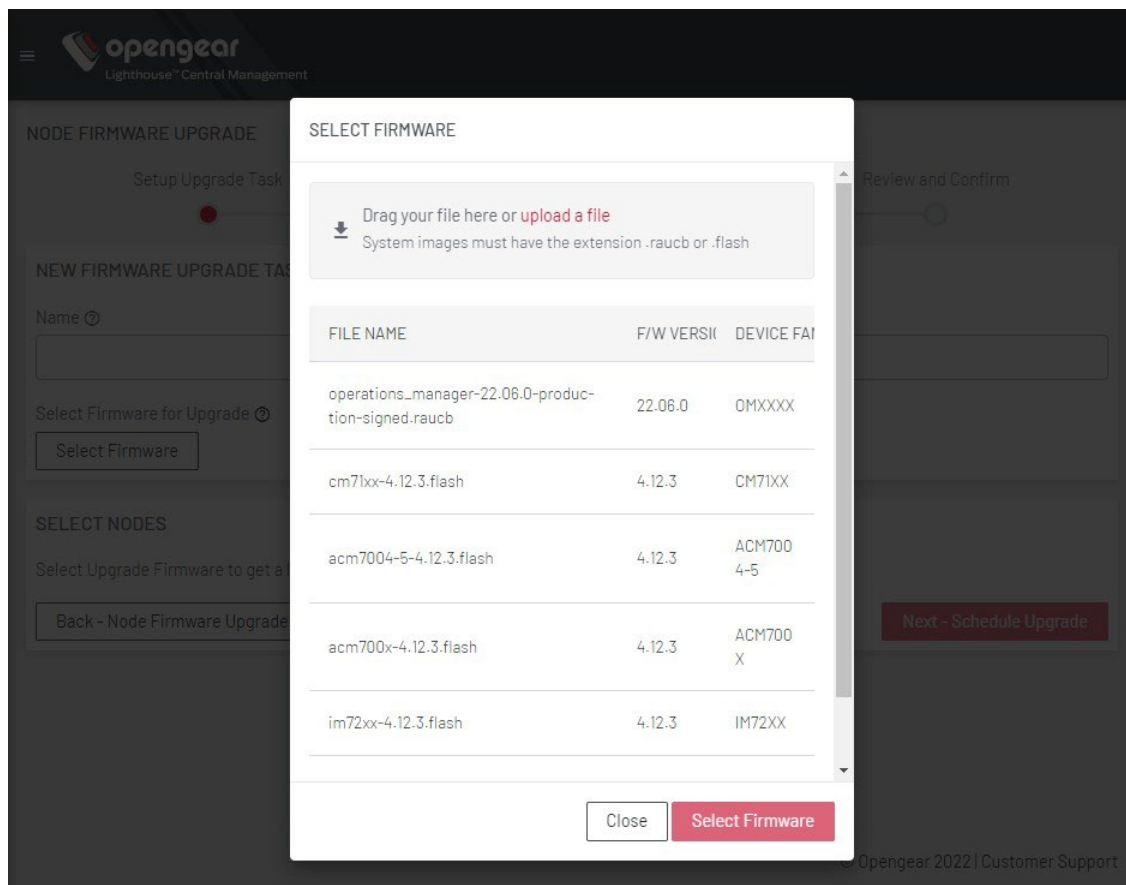
Select Firmware

SELECT NODES

Select Upgrade Firmware to get a list of compatible Nodes

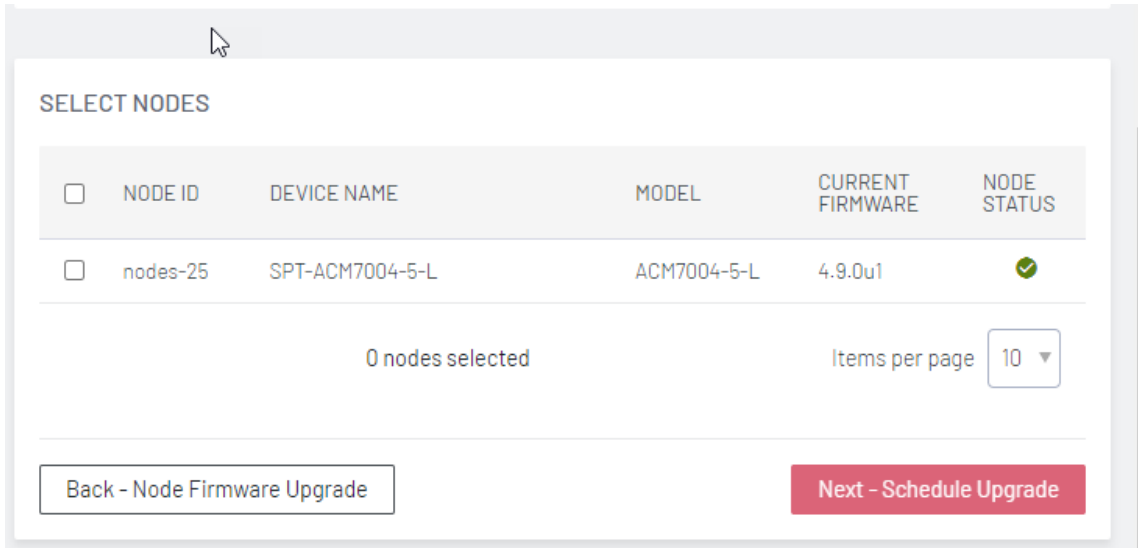
Back - Node Firmware Upgrade      Next - Schedule Upgrade

- Click the **Select Firmware** button. The Select Firmware list displays.



- Select the firmware upgrade or upload new firmware for the upgrade task.  
Click **Select Firmware** button.

5. Select the nodes to be upgraded. Select **Next - Schedule Upgrade**.



**SELECT NODES**

<input type="checkbox"/>	NODE ID	DEVICE NAME	MODEL	CURRENT FIRMWARE	NODE STATUS
<input type="checkbox"/>	nodes-25	SPT-ACM7004-5-L	ACM7004-5-L	4.9.0u1	✓

0 nodes selected      Items per page 10 ▼

[Back - Node Firmware Upgrade](#)      [Next - Schedule 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 will not be listed.

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.



**START TIME**

☐ Immediately after creation    ☒ Set time

Current Time:  
20 May 2022 03:27 UTC

ⓘ Firmware upgrade task times can vary based on latency, hardware, and node count.  
During the upgrade, each appliance will reboot and will be unreachable for several minutes.

Scheduled time:  
20 May 2022 at 15:25 UTC  
In approximately 11 hours

**Note:** The scheduled time is always in UTC.

7. If Upgrade in Failover is selected, it is selected either in bulk, by dropdown selection (a), or, individually by a toggle switch beside each table row (b):



**Note:** Selecting this option may result in considerable cell charges in the event of a failover.

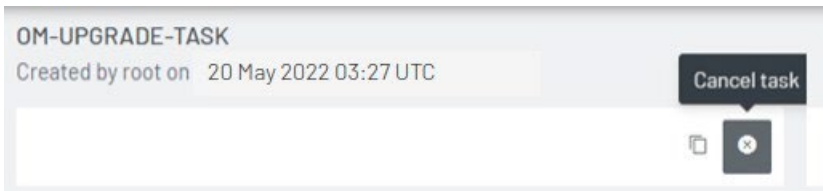
8. Click **Next – Review and Confirm** to go to the review screen. Check the schedule details are correct. To change schedule details, click **Back – Schedule Upgrade**, all data local data will be preserved while you change parameters on previous screens.
9. Select **Confirm**, and type **Yes** at the prompt, then click **Confirm** to create the task.

## Cancel an upgrade task

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

To cancel an upgrade task"

1. Select **Settings > Services > Node Firmware Upgrade**
2. In the task list, select the task name you wish you cancel, the Task Details screen displays.
3. Click the X button (top-right) to cancel the task.



**Note:** You cannot cancel upgrade tasks that have already been completed.

### Limitations:

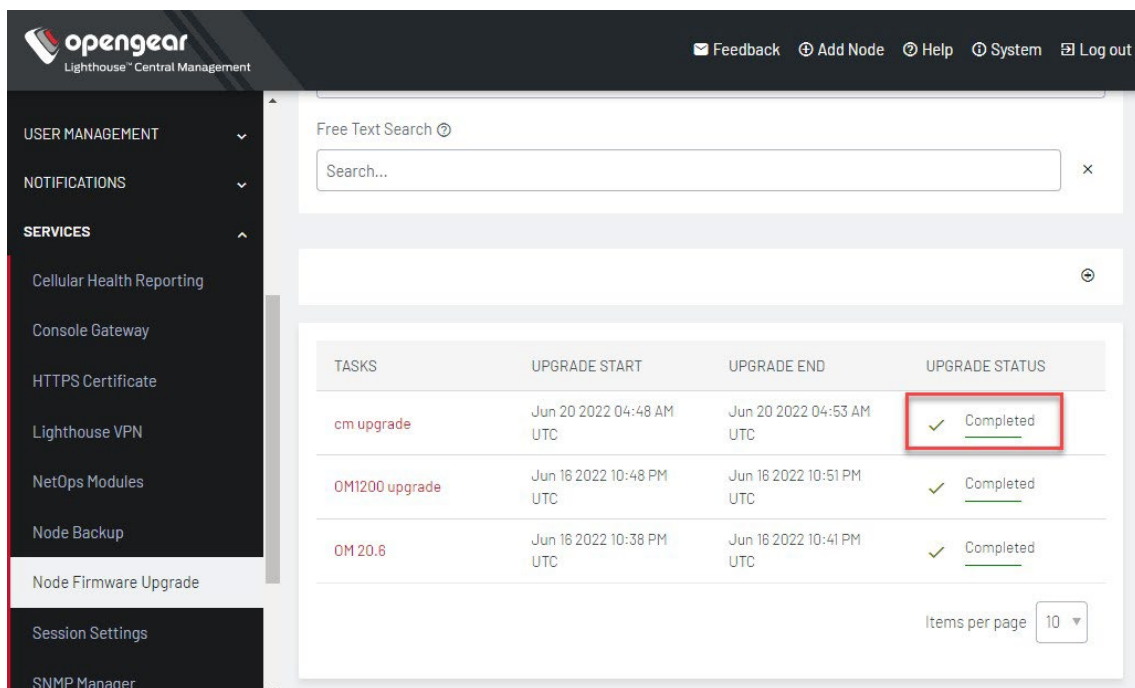
- Do not cancel an upgrade job just as it is about to begin, for example, 10 seconds before or after the start time.
- If an upgrade is cancelled while in progress, only nodes that have not yet been upgraded will be cancelled.

## Copy a scheduled task

You can copy a scheduled task in order to create a new upgrade task. The new upgrade task will use 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 **Settings > Services > Node Firmware Upgrade**



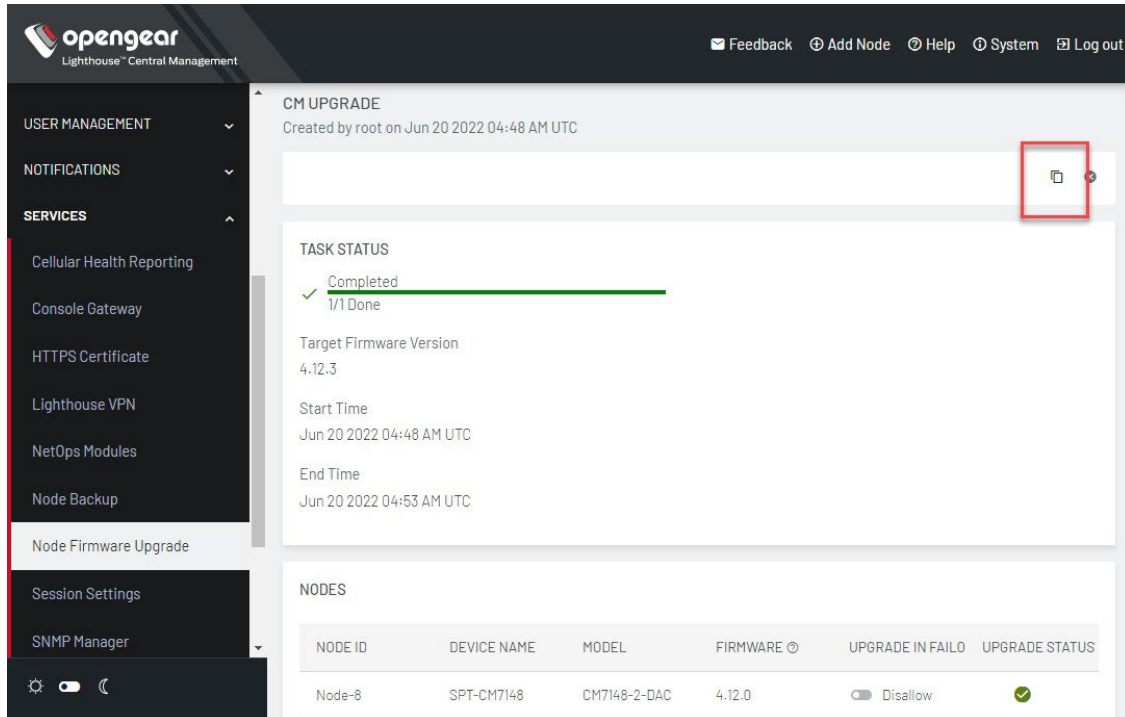
TASKS	UPGRADE START	UPGRADE END	UPGRADE STATUS
cm upgrade	Jun 20 2022 04:48 AM UTC	Jun 20 2022 04:53 AM UTC	✓ Completed
OM1200 upgrade	Jun 16 2022 10:48 PM UTC	Jun 16 2022 10:51 PM UTC	✓ Completed
OM 20.6	Jun 16 2022 10:38 PM UTC	Jun 16 2022 10:41 PM UTC	✓ Completed

Items per page 10 ▾

2. Select the task you wish you copy, by clicking the task name in the task table



- Click the Copy button to copy the task



**opengear**  
Lighthouse™ Central Management

Feedback Add Node Help System Log out

**USER MANAGEMENT** **NOTIFICATIONS** **SERVICES**

- Cellular Health Reporting
- Console Gateway
- HTTPS Certificate
- Lighthouse VPN
- NetOps Modules
- Node Backup
- Node Firmware Upgrade**
- Session Settings
- SNMP Manager

**CM UPGRADE**  
Created by root on Jun 20 2022 04:48 AM UTC

**TASK STATUS**

✓ Completed  
1/1 Done

Target Firmware Version  
4.12.3

Start Time  
Jun 20 2022 04:48 AM UTC

End Time  
Jun 20 2022 04:53 AM UTC

**NODES**

NODE ID	DEVICE NAME	MODEL	FIRMWARE	UPGRADE IN FAILO	UPGRADE STATUS
Node-8	SPT-CM7148	CM7148-2-DAC	4.12.0	Disallow	✓

- A new task is created with the same nodes.

## Delete an upgrade task

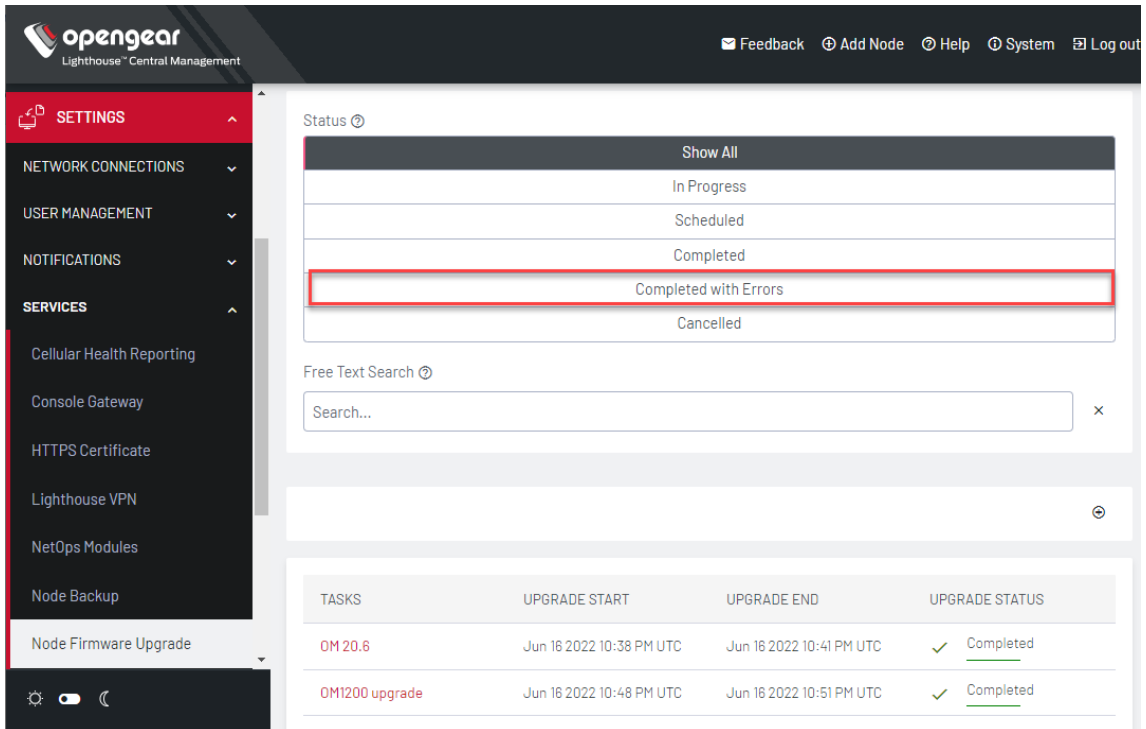
Upgrades cannot be permanently deleted as they can offer a valuable insight into the health of the nodes when problem-solving and provide the version path that they have traversed over their lifetime.

If the number of jobs is becoming unmanageable, or jobs need to be deleted for security measures, the support team will be able to 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 **Settings > Services > Node Firmware Upgrade**
2. Select the task you wish you retry, by clicking the task name in the task table. You can filter the task list by using the **Completed with Errors** filter.



TASKS	UPGRADE START	UPGRADE END	UPGRADE STATUS
OM 20.6	Jun 16 2022 10:38 PM UTC	Jun 16 2022 10:41 PM UTC	✓ Completed
OM1200 upgrade	Jun 16 2022 10:48 PM UTC	Jun 16 2022 10:51 PM UTC	✓ Completed

3. In the task detail screen, click the **Repeat for failed upgrades** button.



**TASK STATUS**

Completed  
3/3 Done

Repeat for failed upgrades

**Note:** If the relative Firmware file has been deleted, the **Repeat for failed upgrades** button is not displayed.

4. The **Node Firmware Upgrade** page displays. See ["Create an upgrade task" on page 218](#).

## Node Upgrade Runtime Behaviour

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

## Promoting 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.

## Downgrading and Skipping 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.Q3 to 21.Q4, it will ignore any nodes that are already at 21.Q4 or 22.Q1.

### Skipping versions

If a node is scheduled to be upgraded from 21.Q3 directly to 22.Q1 (skipping 21.Q4), it will upgrade the node even if it has been manually upgraded to 21.Q4 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 will be based upon 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 will be skipped.

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

## Node Connection interrupted

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 need 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.

## Unenrolling Nodes at Upgrade

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

## 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 timezone, 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 will not be run.

## Configuring Lighthouse

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

## Create Templates

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

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

The available group roles are:

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

## Creating new user and group templates

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

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

The available group roles are:


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

To create a new users and groups template:

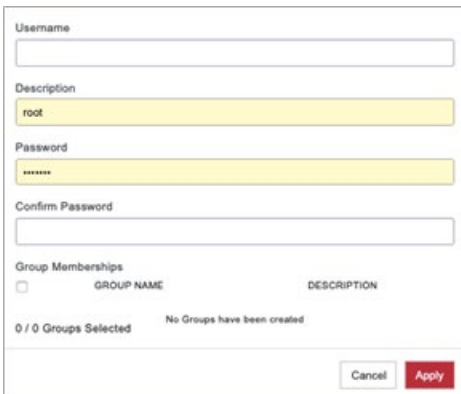
1. Select **CONFIGURE > CONFIGURATION TEMPLATING > Users and Groups Templates**.
2. Click the **+** button. The **New Users and Groups Template** page displays.



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


 A dialog box titled "GROUP DETAILS". It contains three input fields: "Group Name" (a single-line text box), "Description" (a multi-line text box), and "Role" (a dropdown menu currently showing "Node Administrator"). At the bottom right are two buttons: "Cancel" and "Apply".

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


 A dialog box titled "User Details". It contains four input fields: "Username" (a single-line text box), "Description" (a single-line text box with "root" entered), "Password" (a single-line text box with masked characters), and "Confirm Password" (a single-line text box). Below these is a "Group Memberships" section with a checkbox and a table with columns "GROUP NAME" and "DESCRIPTION". The table is empty, and below it says "0 / 0 Groups Selected" and "No Groups have been created". At the bottom right are two buttons: "Cancel" and "Apply".



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

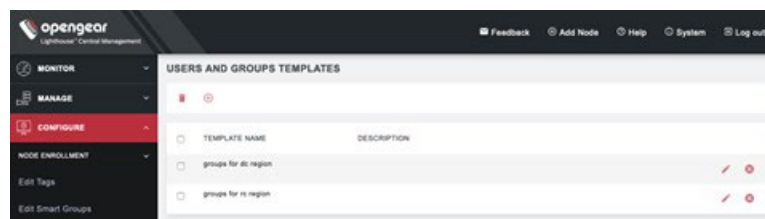
**Note:** When a users and groups template is pushed to a node, all custom groups on that node are replaced by groups defined in the template. If no users are in the new template, existing users will remain on the node. To push users, the selected nodes need to be running firmware version 4.3.0 or later.

## Modifying existing users and groups templates

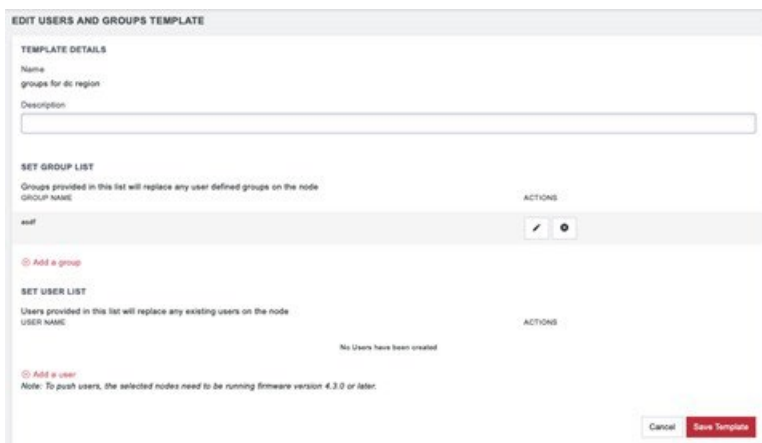
Use the **Edit Users and Groups Template** dialog to modify a template's **Description**, **Group List**, and **User List**.

To modify a template:

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



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



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

## Deleting users or groups from a template

To delete users or groups from a template:

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

## Deleting users and groups templates

To delete a template:

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

The template is deleted.

## Create Authentication Templates

Only users assigned to the *Lighthouse Administrator* role can access

**CONFIGURE > CONFIGURATION TEMPLATING > Authentication Templates**  
and create authentication templates.

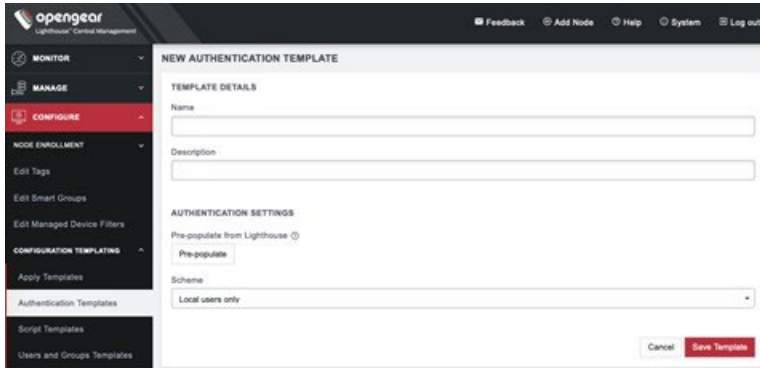
## Creating new authentication templates

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

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

To create a new authentication template:

1. Select **CONFIGURE > CONFIGURATION TEMPLATING > Authentication Templates**.
2. Click the **+** button. The **New Authentication Template** page loads.



3. Enter a **Name** and **Description** for a template in the **Template Details** section.
4. Select a desired **Scheme** or click **Pre-populate** to pre-populate a template with the current Lighthouse remote authentication configuration.

5. Enter or update authentication settings if required. See 8.10 Configuring AAA for an example.
6. Click **Save Template**.

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

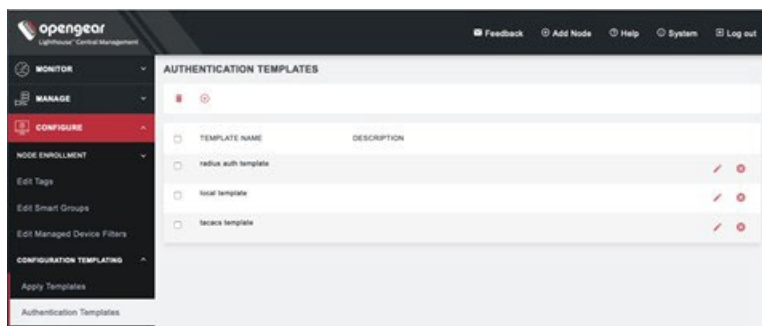
**Note:** The authentication templates do not support the full list of settings that the Opengear console servers support. However, templates can be applied, and then additional settings configured manually.

## Modifying existing authentication templates

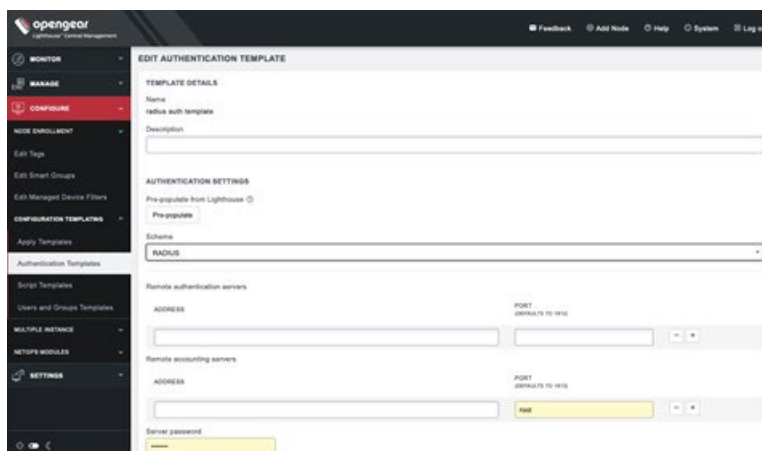
The **Edit Authentication Template** dialog allows the template's *Description* and **Authentication Settings** to be set and changed.

To modify an existing authentication template:

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



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



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



## Deleting authentication templates

To delete an authentication template:

1. Select **CONFIGURE > CONFIGURATION TEMPLATING > Authentication Templates**.
2. Click **Delete** next to the template to be removed. The **Confirmation** dialog displays.
3. Click **Yes** in the **Confirmation** dialog.

The authentication template is deleted.

## Create 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 and
- can't be more than 1MB in size.

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

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

## Creating new script templates

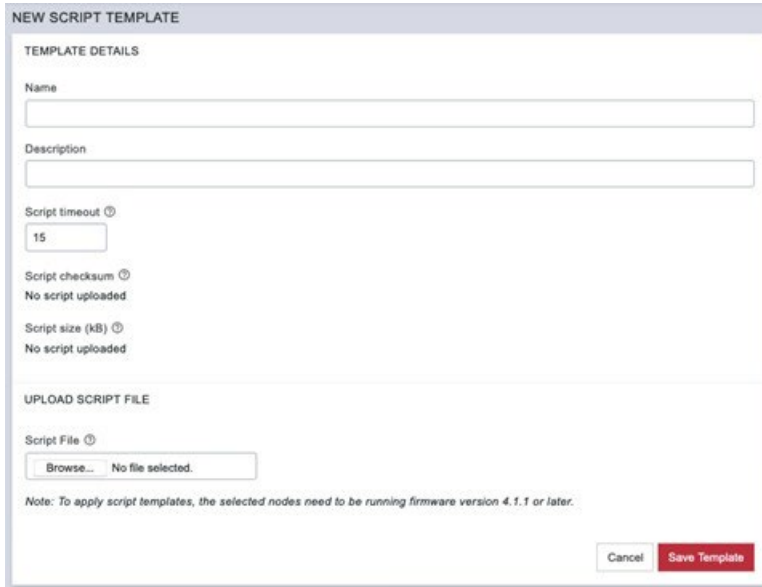
Script Templates allow the user to upload arbitrary shell scripts to be run on a node. A script may set additional configuration settings not available in other templates or store additional files onto the node such as certificates, for example. The uploaded script must have a .sh extension and can't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

Lighthouse Administrators can create script templates via **CONFIGURE > CONFIGURATION TEMPLATING > Script Templates**

To create a new script template:

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

- Click the **+** button. The **New Script Template** dialog loads.



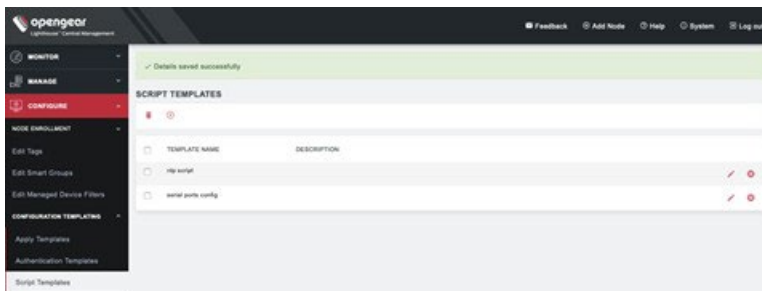
- Enter a **Name** and **Description** for a template in the **Template Details** section.
- To select a script to upload, click **Choose file**.
- Click **Save Template**.

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

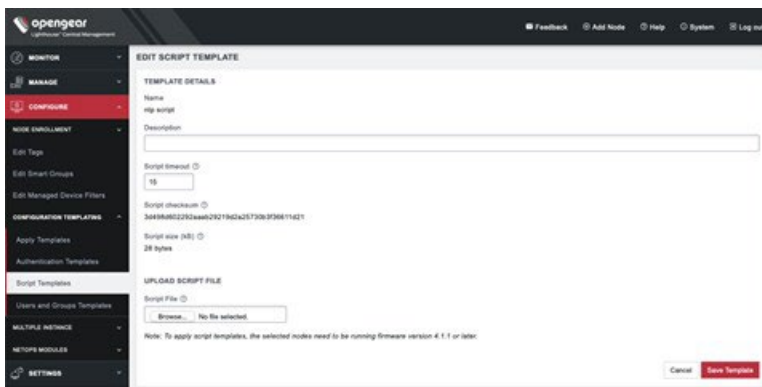
## Modifying existing script templates

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

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



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



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

## Deleting script templates

To delete a script template completely:

1. Select **CONFIGURE > CONFIGURATION TEMPLATING > Script Templates**.
2. Click **Delete** next to the template to be removed. The **Confirmation** dialog displays.
3. Click **Yes** in the **Confirmation** dialog.

The script template is deleted.

## Use 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 **CONFIGURE > CONFIGURATION TEMPLATING > Apply Templates** and execute templates affecting any node.

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

## Apply Templates

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

**Apply Templates** consists of four stages, each one a step in the overall wizard.

The steps are:

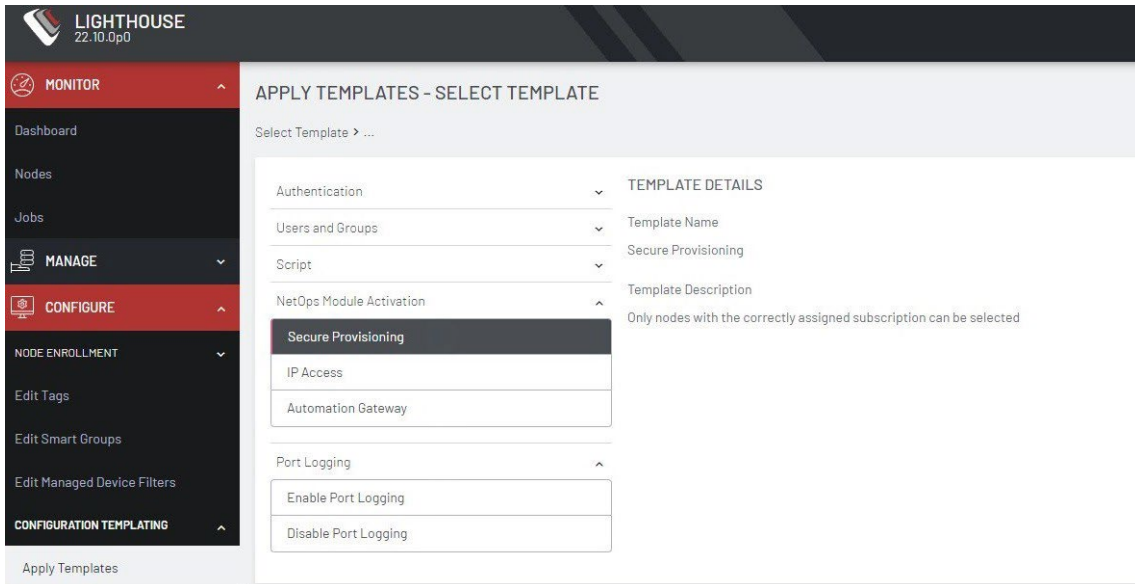
1. Select Template.
2. Select Nodes.
3. Preflight. This test run simulates what happens if the template is pushed to the selected nodes.
4. Execution.

To apply a template:

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



2. Select a template from the existing template tree. **Template Details** populates with details from the selected template.



3. Click **Next — Select Nodes**. The **Select Nodes** stage loads.
4. Select nodes from the list of enrolled nodes. **Smart Group Filtering** and **Free Text Search Filtering** can be used to narrow down the results.

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

5. Scroll to the bottom of the page and click **Next — Preflight**. The **Preflight** stage loads. This stage requires manual refresh to retrieve updated **Preflight Result** and **Details**.

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

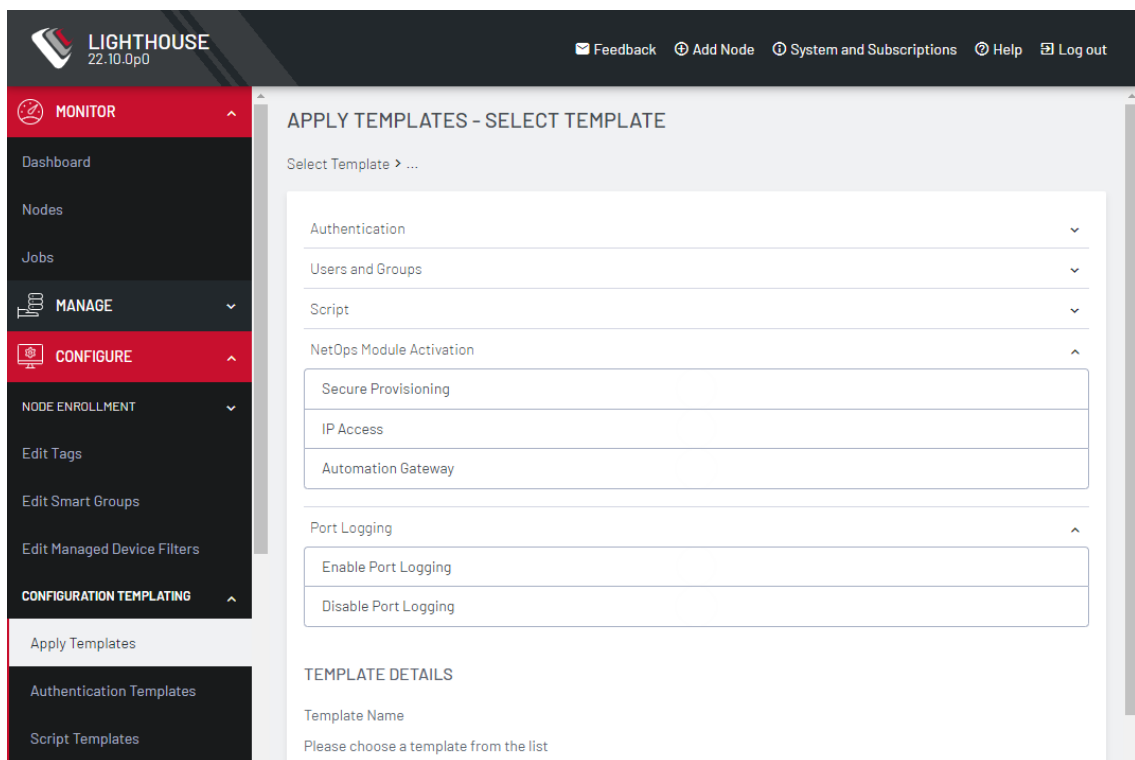
6. Select desired nodes for template execution and click **Next — Push Configuration**. The **Configuration Status** stage loads. This stage requires manual refresh to retrieve updated **Push Result** and **Details**.

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

## Manually Activate NetOps modules via Template

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

1. Select **CONFIGURE > CONFIGURATION TEMPLATING > Apply Templates**
2. Click **Secure Provisioning** or **IP Access** or **Automation Gateway** under **NetOps Module Activation**.



**LIGHTHOUSE** 22.10.0p0

Feedback Add Node System and Subscriptions Help Log out

**MONITOR**

Dashboard

Nodes

Jobs

**MANAGE**

**CONFIGURE**

NODE ENROLLMENT

Edit Tags

Edit Smart Groups

Edit Managed Device Filters

**CONFIGURATION TEMPLATING**

Apply Templates

Authentication Templates

Script Templates

**APPLY TEMPLATES - SELECT TEMPLATE**

Select Template > ...

Authentication

Users and Groups

Script

NetOps Module Activation

Secure Provisioning

IP Access

Automation Gateway

Port Logging

Enable Port Logging

Disable Port Logging

**TEMPLATE DETAILS**

Template Name

Please choose a template from the list

3. Click **Next – Select Nodes**
4. Choose the desired **NetOps Console Server** 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 **Next – Preflight**. Refresh to ensure the preflight check has succeeded.
6. When preflight is complete, click **Next - Push Configuration**.

## Configuration & 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 Opendaylight Technical Support.

### Generate a support report via the Lighthouse interface

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

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

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

2. Click **Download support report**.

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

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

It contains two files:

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

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

## Generate a support report via the local terminal

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

1. Select **MANAGE > LIGHTHOUSE > Local Terminal**.
2. At the [hostname] login: prompt, enter an administrator username and press Return.
3. At the password: prompt, enter the administrator's password and press Return.
4. At the bash shell prompt, enter  
`support-report -z > /tmp/support.zip`  
and press Return

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

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

Here are two options for copying the file from Lighthouse:

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

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

For Windows users, WinSCP on Win10 also works.

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

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

## 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+,

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

## Role Description

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

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



## Work with Groups

User groups are used to organize user accounts and their associated privileges. A user group is associated with a set of nodes and managed devices, and users who are members of the group will have access to those devices through Lighthouse. A group may also be associated with one or more roles, which provide permissions to access operational features in Lighthouse.

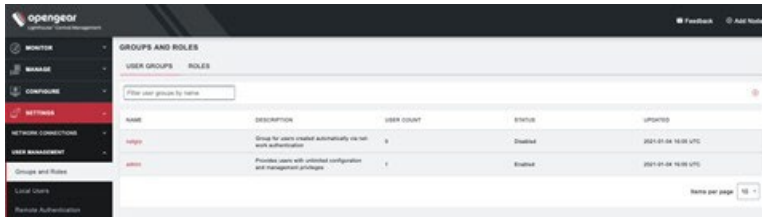
A user may be a member of multiple groups.

## Creating New Groups and Roles

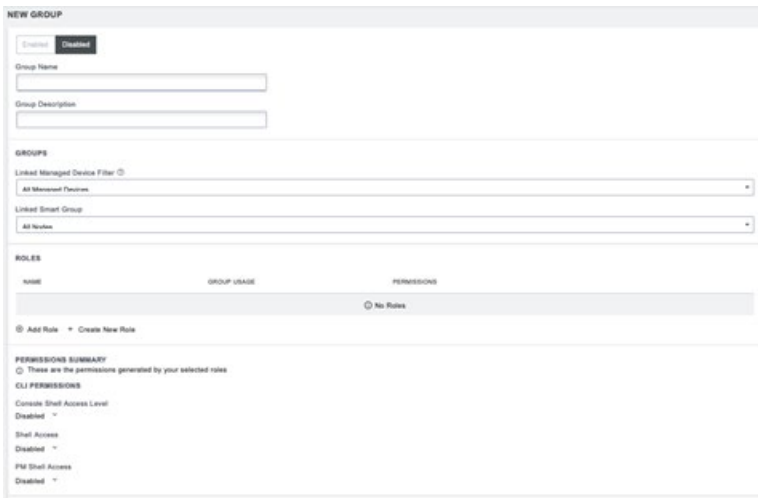
Create new user groups and roles with specific permissions for your Lighthouse instance.

## Create a new group

1. Select **SETTINGS > USER MANAGEMENT > Groups and Roles**. Select the **User Groups** tab.



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



**NEW GROUP**

Enabled Disabled

Group Name

Group Description

**GROUPS**

Linked Managed Device Filter

All Managed Devices

Linked Smart Group

All Nodes

**ROLES**

NAME	GROUP USAGE	PERMISSIONS
No Roles		

Add Role Create New Role

**PERMISSIONS SUMMARY**

These are the permissions generated by your selected roles

**CLI PERMISSIONS**

Console Shell Access Level

Disabled

Shell Access

Disabled

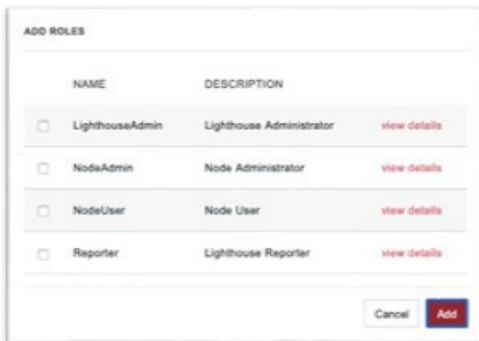
FW Shell Access

Disabled

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

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

4. The **CLI Permissions** section displays Command Line Interface(CLI) permissions based on the roles you have assigned to this group. To change the permissions, you can edit or add new roles with the desired CLI Permissions. See ["Create a new Role" on page 273](#).
5. Click **Enabled** to enable group.
6. If desired, you can select a **Linked Managed Device Filter** and **Linked Smart Group** to associate with this group.
7. Add one or more roles by clicking **Add Role** and checking the desired roles.



ADD ROLES		
NAME	DESCRIPTION	
<input type="checkbox"/> LighthouseAdmin	Lighthouse Administrator	<a href="#">view details</a>
<input type="checkbox"/> NodeAdmin	Node Administrator	<a href="#">view details</a>
<input type="checkbox"/> NodeUser	Node User	<a href="#">view details</a>
<input type="checkbox"/> Reporter	Lighthouse Reporter	<a href="#">view details</a>

Cancel Add

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

REPORTER  
[Back](#)

CLI PERMISSIONS SUMMARY

Console Shell Access Level  
 Disabled

Shell Access  
 Disabled

PM Shell Access  
 Disabled

OPERATION PERMISSIONS SUMMARY

FEATURE	FULL ACCESS	READ ONLY	DENY
Actions	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Events	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Subscriptions	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Logging	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Netops	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Nodes & Configuration	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Service Settings	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Smartgroups & Tags	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>

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

**Note:** See [Available Operations Permissions](#) for a list of all options.

OPERATION PERMISSIONS

FEATURE	FULL ACCESS	READ ONLY	DENY
Actions	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Logging	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Netops	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Nodes & Configuration	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Service Settings	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Smartgroups & Tags	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
System	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Users & Permissions	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>

[Cancel](#) [Save Group](#)

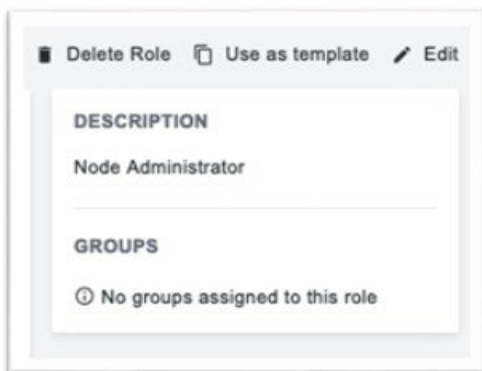
9. Click **Save Group**.

## Available Roles:

- **Lighthouse Administrator:** Members of groups with this role have **Full** access to all nodes and managed devices.
- **NodeAdmin:** Has no shell access. Has **Read Only** access to Netops Modules, all Nodes & Configuration Operations, Cell Health, Smart Groups, Tags, and Jobs.
- **NodeUser:** Has **PM Shell** access. Has **Read Only** access to Nodes & Devices (Base) and Tags.
- **Lighthouse Reporter:** Has no shell access. Has **Read Only** access to all Operations.

You can also create a custom role that allows you to modify **CLI Permissions** and **Operations Permissions** by clicking "[Create a new Role](#)" on [page 273](#) on the New Group page.

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



## Available Operations Permissions:

- **Logging**

Port Logging – Manage port logging settings.

Syslog – Manage system syslog settings.

- **Netops**

Netops Modules

- **Nodes & Configuration**

Nodes & Devices (Base) – Access to dashboard, nodes, managed devices, node Enrollment, console gateway, and Node web UI.

Nodes & Devices (Advanced) – Access to jobs, pending nodes, smart groups, and managed device filters.

Nodes Firmware Management

Template Push – Manage templates and push templates to nodes.

- **Service Settings**

LHVPN

Cell Health

Console Gateway

Date & Time

HTTPS

Netops – Install Netops modules and manage local Netops repositories.

Node Backup

Session Settings

SNMP

SSH

Syslog

- **Smartgroups & Tags**

Bundles – Manage and use bundles.

Smart Groups – Manage and use smart groups.

Tags – Manage and use tags.

- **System**

Admin & Licensing – Manage access settings for Lighthouse and license settings.

Backup & Restore

Jobs

Multi-instance – Manage multi-instance settings and control state of instances.

Network Interfaces – Manage network interface settings.

System Upgrade & Reset

- **Users & Permissions**

Authentication – Manage authentication settings including methods, policies, and restrictions.

Groups & Roles – Create and edit groups and roles. May not assign them to users.

Users – View, manage, create, and delete users.



CREATE RULE

Rule Name

Rule Description

CLI PERMISSIONS

Console Shell Access Level

Admin

Readonly

Disabled

Shell Access

Admin

Readonly

Disabled

FW Shell Access

Admin

Readonly

Disabled

OPERATION PERMISSIONS

FEATURE	FULL ACCESS	READ-ONLY	DENY
Actions	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Logging	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Network	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Nodes & Configuration	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Service Settings	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Snapgroups & Tags	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
System	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Users & Permissions	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Cancel

Create Rule

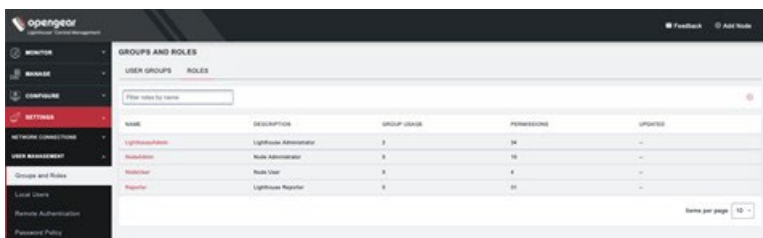
**Note:** When a new group is given the **Lighthouse Administrator** role, members of the group have access to the `sudo` command. Groups or users with the **Lighthouse Administrator** role are added to the admin group, which is in the list of allowed `sudoers`. On first boot of a new Lighthouse instance,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

## Create a new Role

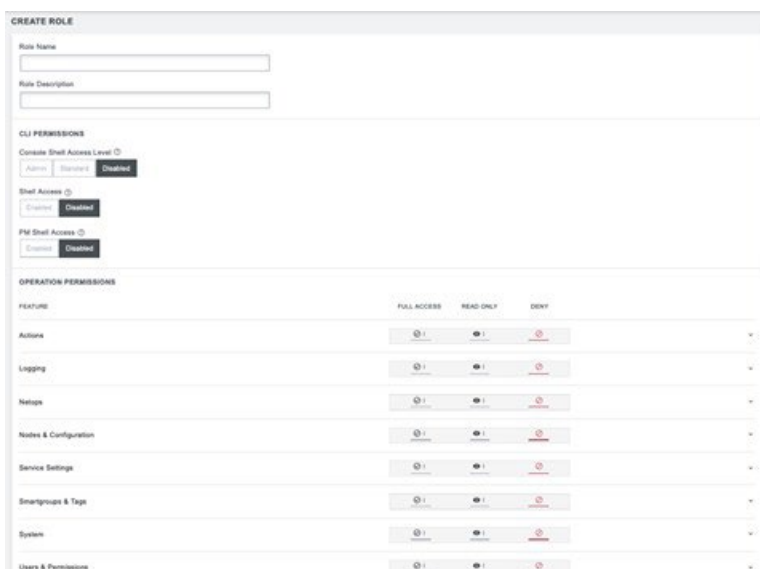
You can create a new role in two ways, either from the Roles tab or by using an existing role as a template.

**To create a new role from the Roles tab:**

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



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



**CREATE ROLE**

Role Name:

Role Description:

**CLI PERMISSIONS**

Console Shell Access Level: ☐ Admin ☐ Standard ☐ Disabled

Shell Access: ☐ Enabled ☐ Disabled

PM Shell Access: ☐ Enabled ☐ Disabled

**OPERATION PERMISSIONS**

FEATURE	FULL ACCESS	READ-ONLY	DENY
Actions	<input type="radio"/> Full <input type="radio"/> Read-Only <input type="radio"/> Deny		
Logging	<input type="radio"/> Full <input type="radio"/> Read-Only <input type="radio"/> Deny		
Network	<input type="radio"/> Full <input type="radio"/> Read-Only <input type="radio"/> Deny		
Nodes & Configuration	<input type="radio"/> Full <input type="radio"/> Read-Only <input type="radio"/> Deny		
Service Settings	<input type="radio"/> Full <input type="radio"/> Read-Only <input type="radio"/> Deny		
Smartgroups & Tags	<input type="radio"/> Full <input type="radio"/> Read-Only <input type="radio"/> Deny		
System	<input type="radio"/> Full <input type="radio"/> Read-Only <input type="radio"/> Deny		
Users & Permissions	<input type="radio"/> Full <input type="radio"/> Read-Only <input type="radio"/> Deny		

4. Enter a **Role Name** and **Role Description**.

5. Modify the **CLI Permissions** for as desired.

**Console Shell Access:** Ability to connect to nodes' command lines via Lighthouse's SSH.

**Shell Access:** Ability to access Lighthouse's command line as administrator.

**PM Shell Access:** Ability to connect to serial ports via SSH.

6. You can also control the **Operation Permissions** for the new role independently. Specify

**Full Access,**

**Read Only,** or

**Deny.**

Click to the right of each Operation row to see all options.

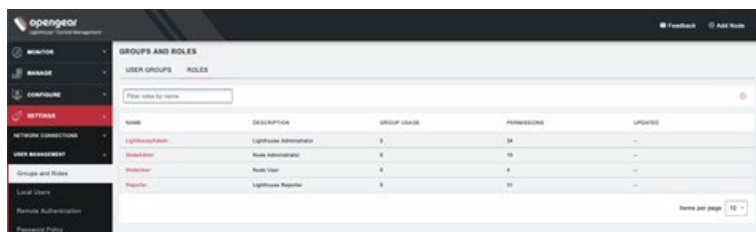
**Note:** See [Available Operations Permissions](#) for a list of all options.

7. Click **Save Role**.

### To create a new role from an existing role:

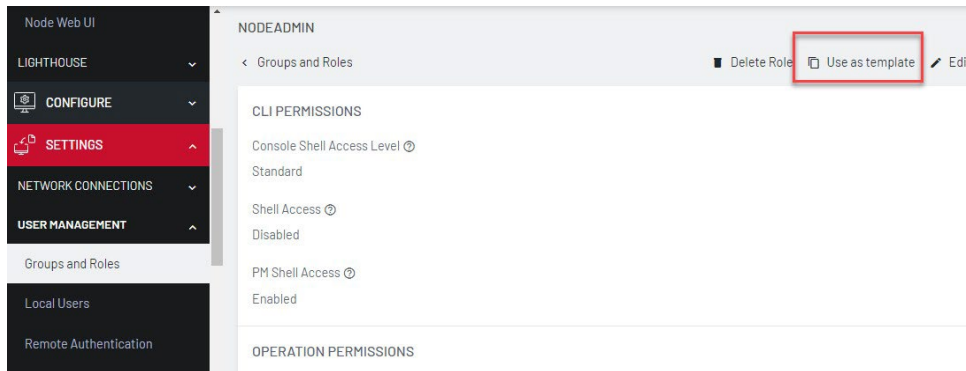
A new role can also be based on an existing role with the **Use as template** link on the upper right of a role's detail page.

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



GROUPS AND ROLES				
USER GROUPS		ROLES		
Filter roles by name				
NAME	DESCRIPTION	GROUP ID	PERMISSIONS	UPDATED
LighthouseAdmin	Lighthouse Administrator	1	34	—
LighthouseNodeAdmin	Node Administrator	2	16	—
LighthouseNodeUser	Node User	3	8	—
LighthouseReporter	Lighthouse Reporter	4	21	—

3. Select the role you wish to copy from the list.

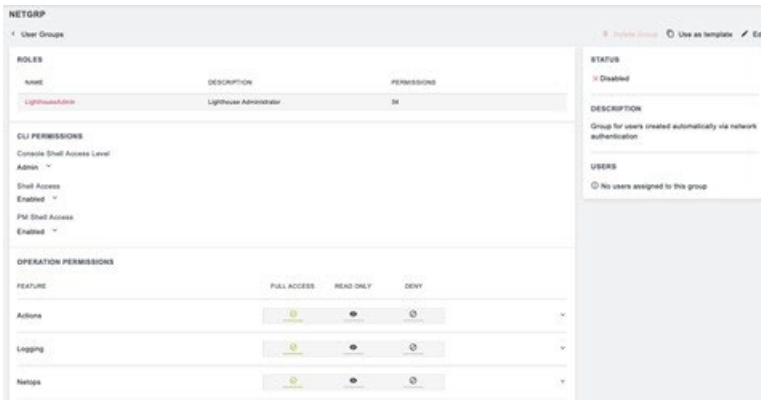


4. Click **Use as template**. The Create Role tab displays with all the settings of the existing role. Make changes if necessary.
5. Click **Save Role** to create the new role.

## Modifying existing groups

To modify an existing group:

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



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

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

See [Creating Smart Groups](#) for details regarding creating and using **Smart Groups** and **Creating Managed Device Filters** for details regarding creating and using **Managed Device Filters**.

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

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

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

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

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

## Work with Users

You can create new users, edit existing users, delete users and alter groups and permissions.

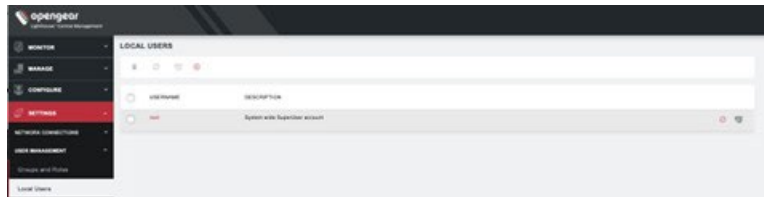


## Creating new local users

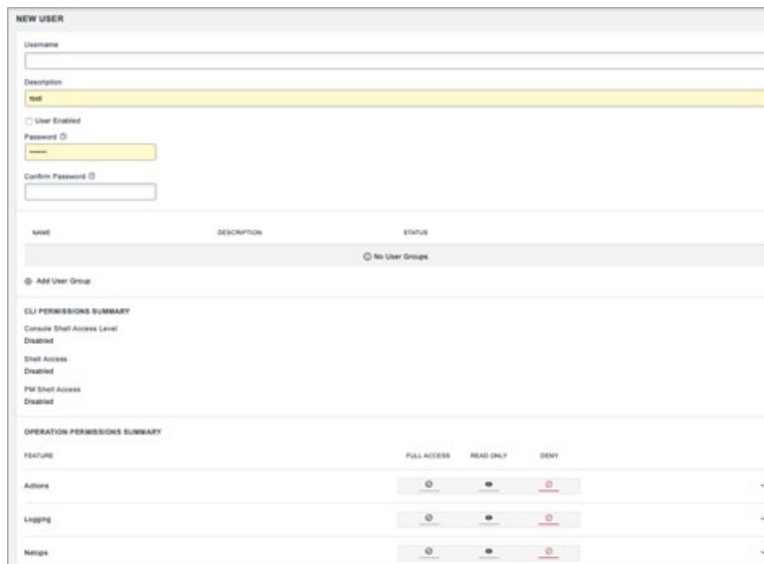
To create a new local user:

1. Select **SETTINGS > USER MANAGEMENT > Local Users**.

By default, the root user is the only user listed.



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



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

Click **Save User**.

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

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

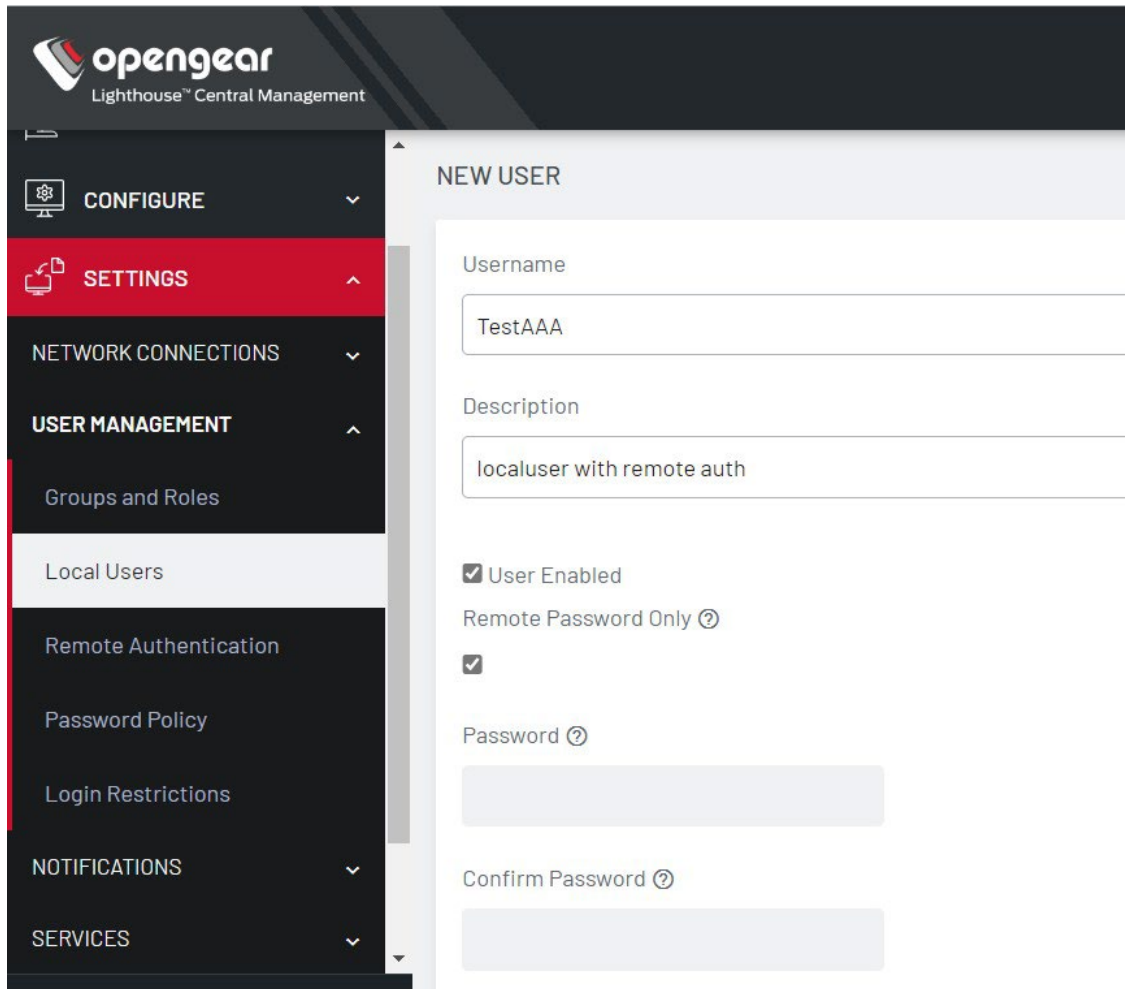
If the created user is set to disabled, the `configurator_users` message does not appear as they have not been added to the passwords file.

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

## Create New Local Users for Remote Authentication

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

1. Select **SETTINGS > USER MANAGEMENT > Remote Authentication**
2. Choose a **Scheme** and enter appropriate settings.
3. Select **SETTINGS > USER MANAGEMENT > Local Users**
4. Click the **+** button. The **New User** dialog loads.



The screenshot shows the OpenGear Lighthouse Central Management web interface. On the left is a dark sidebar with a navigation menu. The 'SETTINGS' menu item is highlighted in red. Under 'USER MANAGEMENT', 'Local Users' is selected. The main content area displays the 'NEW USER' dialog. It contains the following fields and options:

- Username:** A text input field containing 'TestAAA'.
- Description:** A text input field containing 'localuser with remote auth'.
- User Enabled:** A checked checkbox.
- Remote Password Only:** A checked checkbox with a help icon.
- Password:** A password input field (currently empty).
- Confirm Password:** A password input field (currently empty).

5. Enter a **Username** and **Description**.
6. Select the **Remote Password Only** checkbox.
7. Select the **Enabled** checkbox. The Password and Confirm Password fields are disabled.
8. Click **Save User**.

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

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

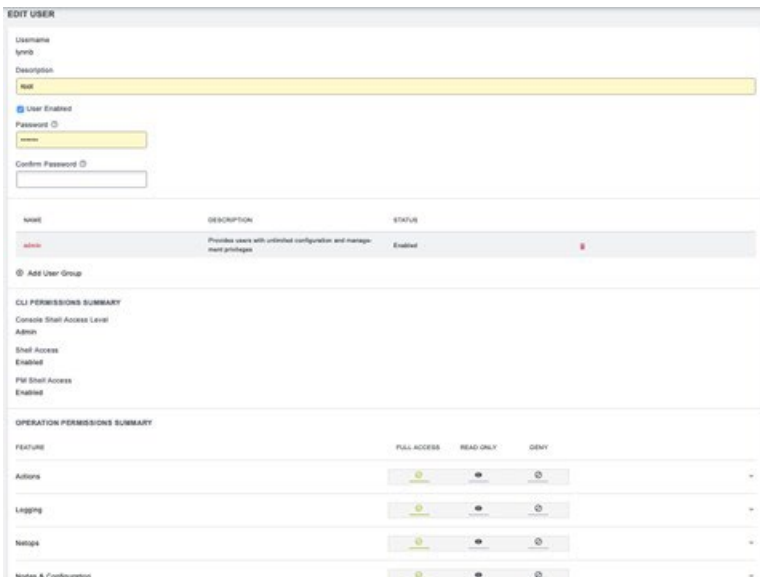
If the created user is set to disabled, the `configurator_users` message does not appear as they have not been added to the passwords file.

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

## Modify existing users

To edit settings for an existing user:

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



**EDIT USER**

Username  
 Name

Description  
 Add

☒ User Enabled

Password

Confirm Password

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

**CLI PERMISSIONS SUMMARY**

Console Shell Access Level  
 Admin

Shell Access  
 Enabled

FW Shell Access  
 Enabled

**OPERATION PERMISSIONS SUMMARY**

FEATURE	FULL ACCESS	READ ONLY	DENY
Actions	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
Logging	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
Networks	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
Nodes & Configuration	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>

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

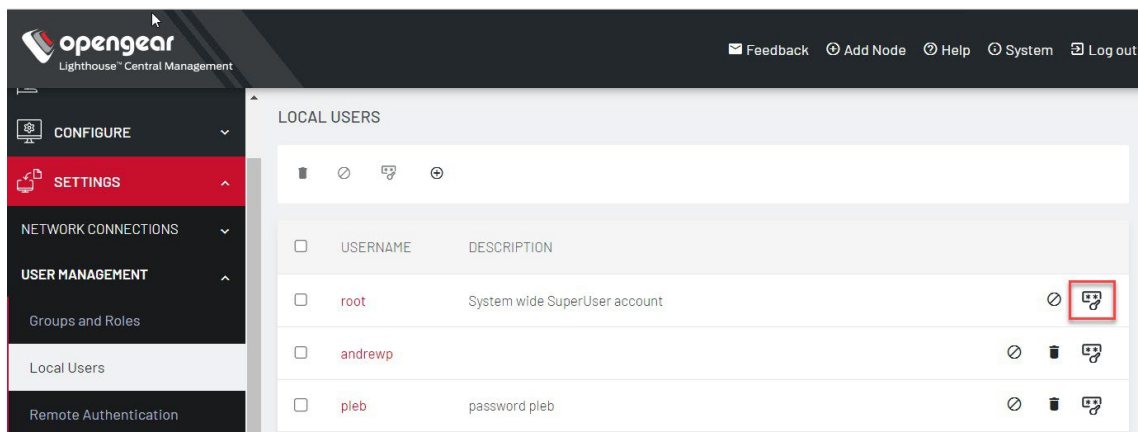
**Note:** Disabled users cannot login to Lighthouse using either the Web-based interface or via shell-based logins (that is `sshusername-`

disabled@lighthouse-name-or-ip). The user and the  
/home/username-disabled directory still exist in the Lighthouse VM file  
system.

## Expire user password

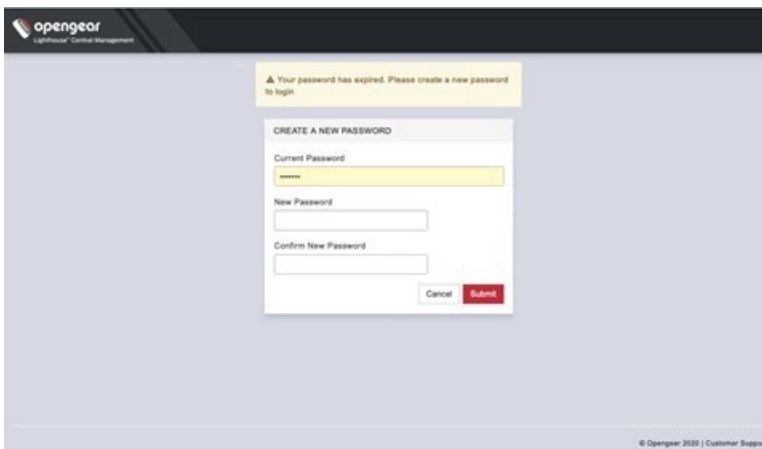
You can set user passwords to expire. To ensure a user's password is set to expire:

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



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

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

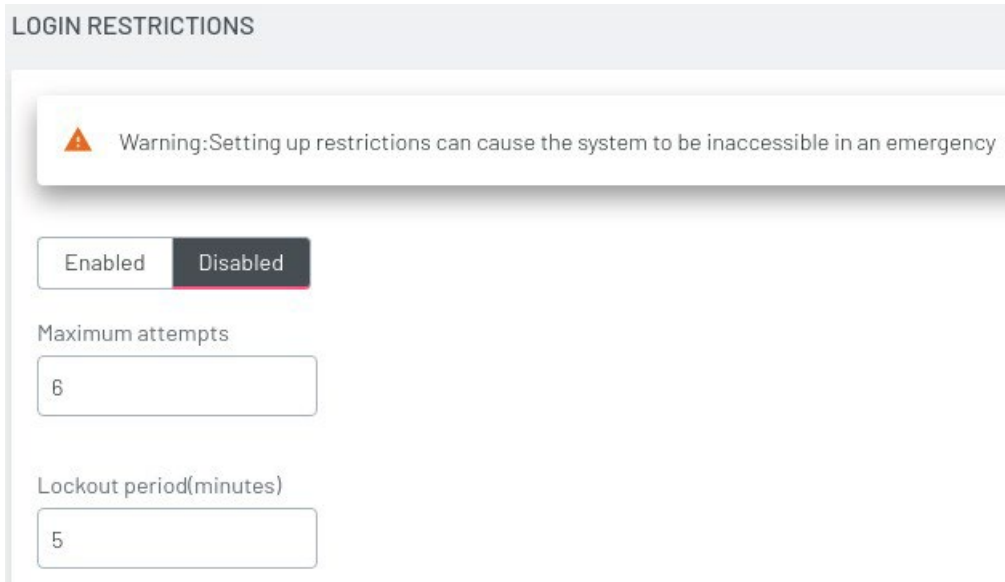


## Setting Login Restrictions

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

**Note:** This does not affect SSH or Console logins.

1. Select SETTINGS > USER MANAGEMENT > Login Restrictions.

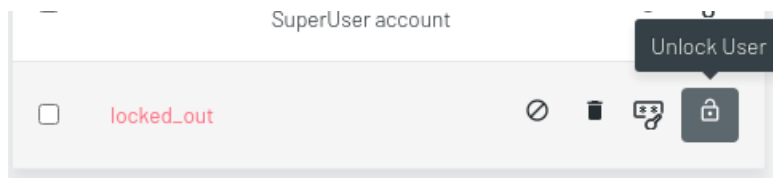


2. Enter a value for either or both of the following:
  - **Maximum attempts** – choose a number of attempts a user can enter an incorrect password before being locked out.
  - **Lockout period** – enter a number of minutes until a user can try to login again after reaching maximum incorrect login attempts.

After a user has been locked out, an administrator can unlock the account.



1. Go to **SETTINGS > USER MANAGEMENT > Local Users**.
2. Click the **Unlock** button associated with the locked out user's row in the User table.



## Disabling a Lighthouse root user

To disable a root user:

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

To 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.

## Delete and Disable Users

You can delete users and disable root users.

To delete a user:

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

### Disabling a Lighthouse root user

To disable a root user:

Make sure that another user exists that is in a group that has the Lighthouse Administrator role.

1. Select **SETTINGS > USER MANAGEMENT > Local Users**
2. Click **Disable** in the **Actions** section of the root user.
3. Click **Yes** in the **Confirmation** dialog.

To re-enable root user:

1. Log in with an existing user id that is in a group with the Lighthouse Administrator role.
2. Click **Enable** in the **Actions** section of the root user.

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

## 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 21.Q4 SAML is only supported for authentication to the lighthouse Web GUI.

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

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

Lighthouse's SAML integrates with the following identity providers:

- . OKTA
- . Azure Active Directory (Azure AD)

- . One Login
- . Auth0

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

Common values you will need are:

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

lighthouse.example.com or 192.168.1.10

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

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

## Generic IdP Setup

This section describes how to integrate Lighthouse with your Generic Identity Provider (IdP) Application.

In case Lighthouse's supported IdPs doesn't 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.

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

**Note:** The `{provider}` in the steps must exactly match one of our provider strings 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 if you only access each via a single address you could create a separate application integration per lighthouse.
6. If your IdP has the option then set the initiator to the `Service Provider`
  7. Set your IdP to sign the **Assertion** for SAML



## Generic IdP SAML Attribute

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

In most IdPs this is done by adding an Attribute Statement or Parameter configuration in your application integration. This parameter should be set as a multi-value parameter, 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. i.e. custom user properties

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

## Lighthouse Setup

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

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

**Note:** 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
```

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

## Examples of Specific IdP Setups

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

Okta

## Create an Application

You need to create an application that Okta will be doing authentication on behalf of.

**Note:** You'll need to know what the addresses of your Lighthouses before creating the application.

1. In the Okta web console go to **Applications - > Applications**
  - a. Click **Create App Integration**
  - b. Select **SAML 2.0**
2. Give the application a name: 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. Tick both:
    - i. Use this for Recipient URL and Destination URL
    - ii. Allow this app to request other SSO URLs
  - b. Fill out the **Requestable SSO URLs** with the SSO URLs for every lighthouse address you want to be able to sign in with. i.e. IP addresses and DNS address for both your primary and secondary lighthouses.  
Example:
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 Onelogin application.
2. Go to More Actions > SAML Metadata. This is the metadata xml file that you will need to configure lighthouse.

## Configure Lighthouse

1. Copy the 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} create -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 **User groups** 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 desire. See [“Creating new user and group templates” on page 238](#).
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 wish to add to the group.

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

## Onelogin

Create an Application. You need 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 will need to configure lighthouse.

## Configure Lighthouse

1. Copy the metadata xml to your primary lighthouse.
2. Using `saml-idp-metadata` on your primary lighthouse, configure each of your lighthouses to use your IdP., 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.

22.11 - REVISION 1	Managing Lighthouse Users	304
--------------------	---------------------------	-----

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

1. Login to Lighthouse as a local user (or any non-SAML user) for example, root.
2. Create the Usergroups with the Roles and permission that you desire. See ["Creating New Groups and Roles" on page 265](#).
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 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

22.11 - REVISION 1	Managing Lighthouse Users	307
--------------------	---------------------------	-----

After this initial setup, you will be able to login as a SAML user. If you do not already have your own Usergroups 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 Usergroups with the Roles and permission required. See ["Creating New Groups and Roles" on page 265.](#)

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**.
7. Go to **Enterprise applications**.
8. Open your App, that is, Lighthouse.
9. Go to **Users and groups**.
10. Click **Add user/group**.
11. Select a user and one of your App roles then click **Assign**.

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

## 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 **Settingstab**.
  - 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 wish 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 setup the 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 Lighthouses for AUTH0 for IdP

In this step you will use the downloaded file to configure each of your lighthouses to use your Auth0 IdP.

1. Copy the downloaded metadata.XML file to your primary lighthouse. See ["Configure AUTH0 Metadata for IdP" on the previous page](#).
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}
```

See ["SAML Config for SSO" on page 291](#)

## Configure AUTH0 for IdP

After you have added Lighthouse as an **Enterprise application** to AUTH0, you need to 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 Usergroups 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.
10. 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 will need to regenerate it then re-export your IdP metadata and update your Lighthouse configurations. If your IdP supports sending expiry notifications to your admin, we recommend you enable these notifications.

## Making 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 need to quickly restrict a user's access, consider altering the permissions of or deleting that user's usergroups on Lighthouse, see [""Modifying existing groups" on page 276](#). You can also set a low Web Session Timeout. See [Examine or modify Lighthouse Session Settings](#).



## SAML SSO Usergroups

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

## SAML SSO Users

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

## Configuring AAA

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

To begin, select **SETTINGS > USER MANAGEMENT > Remote Authentication**.

REMOTE AUTHENTICATION

SETTINGS

Scheme LDAP

Mode LDAPDownLocal

Remote authentication servers

ADDRESS

PORT  
(DEFAULTS TO LDAP/LDAPS STANDARD PORTS)

- +

LDAP base DN ⓘ

LDAP bind DN ⓘ

Bind DN password

Confirm password

LDAP username attribute ⓘ

LDAP group membership attribute ⓘ

☐ Ignore referrals ⓘ

SSL

Server protocol ⓘ LDAP over SSL preferred

☐ Ignore SSL certificate errors ⓘ

CA certificate ⓘ  
 No file selected.

Apply

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

- LDAP/Local
- Local/LDAP

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

3. Add the **Address** and optionally the **Port** of the LDAP server to query.
4. Add the **LDAP Base DN** that corresponds to the LDAP system being queried.  
For example, if a user's distinguished name is **cn=John Doe,dc=Users,dc=ACME,dc=com**, the **LDAP Base DN** is **dc=ACME,dc=com**
5. Add the **LDAP Bind DN**. This is the distinguished name of a user with privileges on the LDAP system to perform the lookups required for retrieving the username of the users, and a list of the groups they are members of.
6. Add and confirm a password for the binding user.
7. Add the **LDAP username** attribute. This depends on the underlying LDAP system. Use **sAMAccountName** for Active Directory systems, and **uid** for OpenLDAP based systems.
8. Add the **LDAP group membership** attribute. This is only needed for Active Directory and is generally **memberOf**.
9. If desired, check **Ignore referrals** option. When checked, LDAP will not follow referrals to other remote authentication servers when logging users in to Lighthouse. If multiple remote authentication servers exist on the network, checking this option may improve login times.
10. Under the **SSL** section, choose the desired **Server 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
11. If desired, check **Ignore SSL certificate** errors to ignore any SSL certificate errors.
  12. **CA Certificate** is used to upload an SSL Certificate which will verify any LDAP servers you specify on the page.

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

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

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

## Radius Configuration

To configure RADIUS:

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

REMOTE AUTHENTICATION

SETTINGS

Scheme

RADIUS

Mode

RADIUSDownLocal

Remote authentication servers

ADDRESS

PORT (DEFAULTS TO 1812)

-

+

Remote accounting servers

ADDRESS

PORT (DEFAULTS TO 1813)

root

-

+

Server password

\*\*\*\*\*

Confirm server password

Apply

1. In the **Settings** section, select **RADIUS** from the **Scheme** drop-down menu.
2. Choose the desired **Mode** from the drop-down menu.
  - RADIUSDownLocal
  - Radius
  - RADIUS/Local
  - Local/RADIUS

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

3. Add the **Address** and optionally the **Port** of the RADIUS authentication server to query.
4. Add the **Address** and optionally the **Port** of the RADIUS accounting server to send accounting information to.
5. Add the **Server password**, also known as the RADIUS Secret.

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

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

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

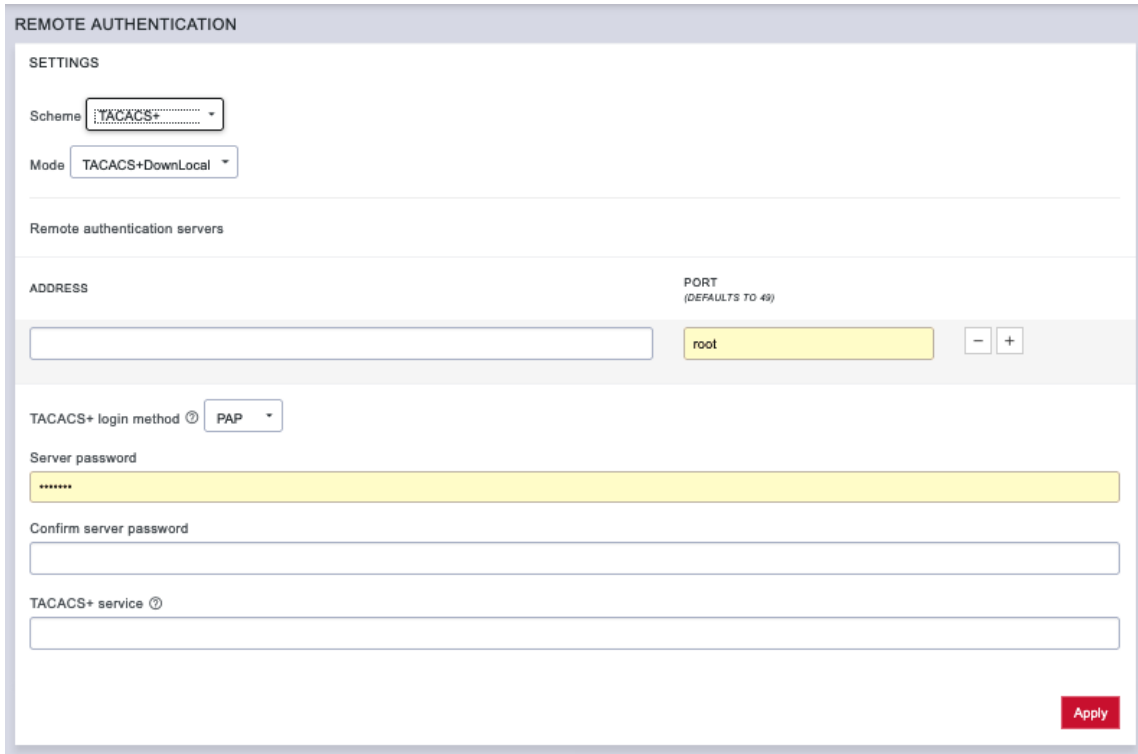
**Note:** The **Framed-Filter-ID** attribute must be delimited by the colon character.



## TACACS+ configuration

To configure TACACS+:

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



**REMOTE AUTHENTICATION**

**SETTINGS**

Scheme: **TACACS+**

Mode: **TACACS+DownLocal**

Remote authentication servers

ADDRESS:

PORT (DEFAULTS TO 49): **root**

TACACS+ login method: **PAP**

Server password:

Confirm server password:

TACACS+ service:

**Apply**

2. Select **TACACS+** from the **Scheme** drop-down menu.
3. Choose the required **Mode** from the drop-down menu.
  - TACACSDownLocal:
  - TACACS: Default behavior
  - TACACS/Local
  - Local/TACACS

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

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

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

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

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

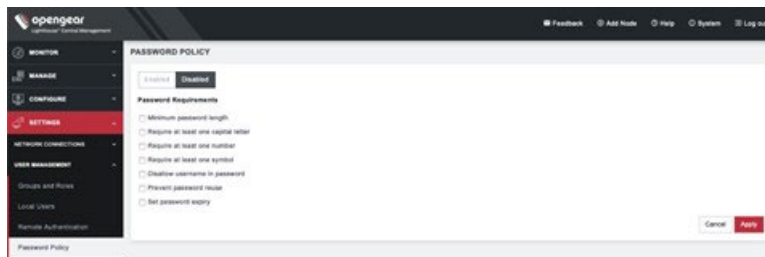
To do this with Cisco ACS, see [Setting up permissions with Cisco ACS 5 and TACACS+ on the Opendgear Help Desk](#).

## Setting password 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.

To set a password policy:

1. Select **SETTINGS > USER MANAGEMENT > Login Restrictions**,
2. Choose **Enabled**, and
3. Click **Save**. Click **Password Policy**.



Choose one or more options from the following:

- Minimum password length – from 1 to 128
- Require at least one capital letter
- Require at least one number
- Require at least one symbol
- Disallow username in password
- Prevent password reuse – choose a number of days or select Always

- Set password expiry – select a number of days until passwords expire. At next login, the user will need to reset the password.

## Password fields in Lighthouse

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

## Managing Automation

The objective of this section of the guide is to familiarize you with NetOps Automation and the modules that comprise IP Access, Secure Provisioning and Automation Gateway.

## Introduction

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 NetOps modules automate the configuration and operation of network infrastructure in data center and remote edge locations.

The platform comprises the following:

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

## About NetOps

NetOps modules require a Lighthouse with appropriate licensing and modules to be uploaded and active. You will also need at least one OGCS or NGCS node enrolled and able to use these modules.

Lighthouse Enterprise contains the IP Access module, whereas, Lighthouse Enterprise: Automation Edition contains IP Access, Secure Provisioning and Automation Gateway.

The NetOps modules are built as Docker containers and accessed through the Lighthouse GUI.



## 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.

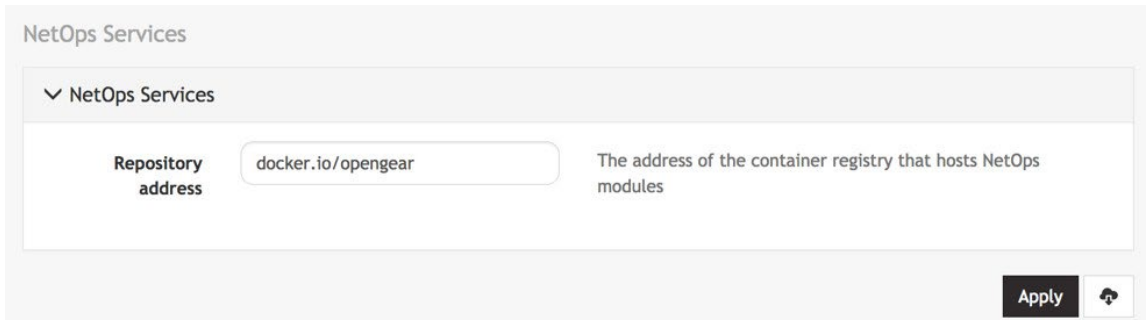
## NetOps Module Management

The NetOps modules, IP Access and Secure Provisioning, Automation Gateway, are stored in a private repository. Periodically, module updates and new modules may become available.

**Note:** You can upgrade NetOps Modules without upgrading Lighthouse software.

## Upgrade Modules from the UI


1. Launch an HTTPS browser session to Lighthouse Login using root and the secure password set earlier.
2. You may also log in as a Lighthouse Administrator user, if you have configured one.
3. From the menu, select **SETTINGS -> Services -> NetOps Modules**.



NetOps Services

▼ NetOps Services

Repository address:  The address of the container registry that hosts NetOps modules

Apply 

4. Click the **Synchronize** widget.
5. From the menu, select **CONFIGURE NODES -> NetOps Modules -> Manage Modules**.
6. Click the **Redeploy** icon.

## Upgrade Modules from the CLI

Log in to the Lighthouse CLI as root.

Or

Log in as a Lighthouse Administrator and become root with: `sudo -i`

Replace default with your root password and run:

PROCEDURE for Lighthouse CLI

Replace **root** and **default** with a Lighthouse Administrator or root credentials, then run the following:

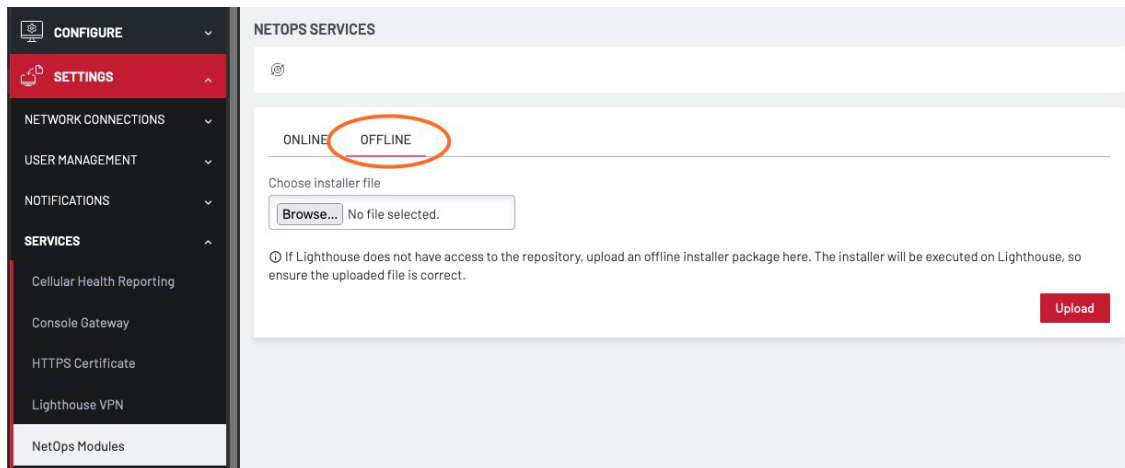
```
USERNAME=root
PASSWORD=default
/etc/scripts/netops_sync_handler
token=$(curl -k -L -d '{
  "username":"'${USERNAME}'", "password":"'${PASSWORD}'"
}'
  "https://127.0.0.1/api/v3.0/sessions/" | python -c 'import sys, json;
  print json.load(sys.stdin)["session"]')
curl -k -L -H "Content-Type: application/json"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

## Update NetOps Modules Without Docker Access

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 script. This script has a built-in payload with everything required to update NetOps Modules on Lighthouse.

## Use a Pre-built Offline Installer (via GUI)

1. Download the offline installer file (named *netops\_modules\_\*.tar.gz*) from this link: [offline installer file](#)
2. Browse to **Settings > Services > NetOps Modules** and click **Offline**.



3. Click **Browse** and select the downloaded file.
4. Click **Upload**.

**Note:** This process can take up to 15 minutes. There are notifications detailing the process steps in the lower right hand corner as well as log entries in the system log.

## Use a Pre-built Offline Installer (via CLI)

1. Download the offline installer file (named *netops\_modules\_\*.tar.gz*) from this link: [offline installer file](#).
2. Copy the offline installer to Lighthouse using scp, WinScp or similar, into the **/mnt/nvram** directory.
3. 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
```

4. Deploy the upgrade to nodes:
  - a. Log in to the Lighthouse web UI as a Lighthouse Administrator or the root user
  - b. From the menu, select **CONFIGURE NODES > NetOps Modules > Manage Modules**.
  - c. In the **Status** column, confirm the module versions now match the latest release versions.
  - d. Click the **Redeploy** icon.

**Note:** Note: Using the offline install requires 8GB of free space in the **/mnt/data** partition.

## Upgrade NetOps Modules

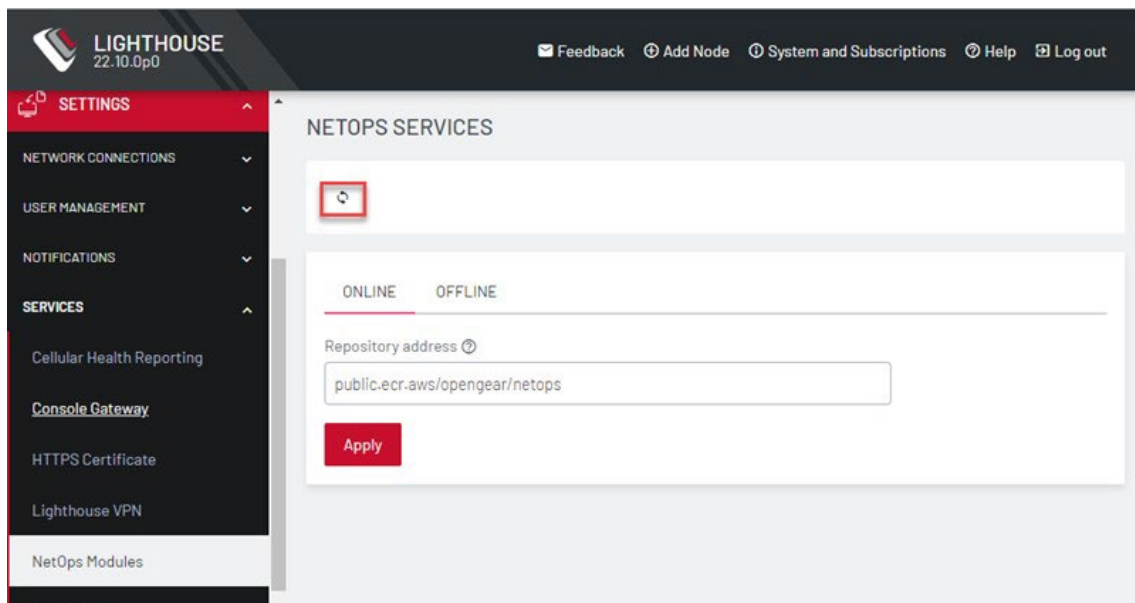
NetOps Modules are released independently of Lighthouse software or Operations Manager firmware.

NetOps releases are uploaded to Opendgear's file server, where they can be fetched by Lighthouse then deployed to all activated nodes by Lighthouse.

Node upgrades may be carried out through either the Lighthouse UI or the CLI.

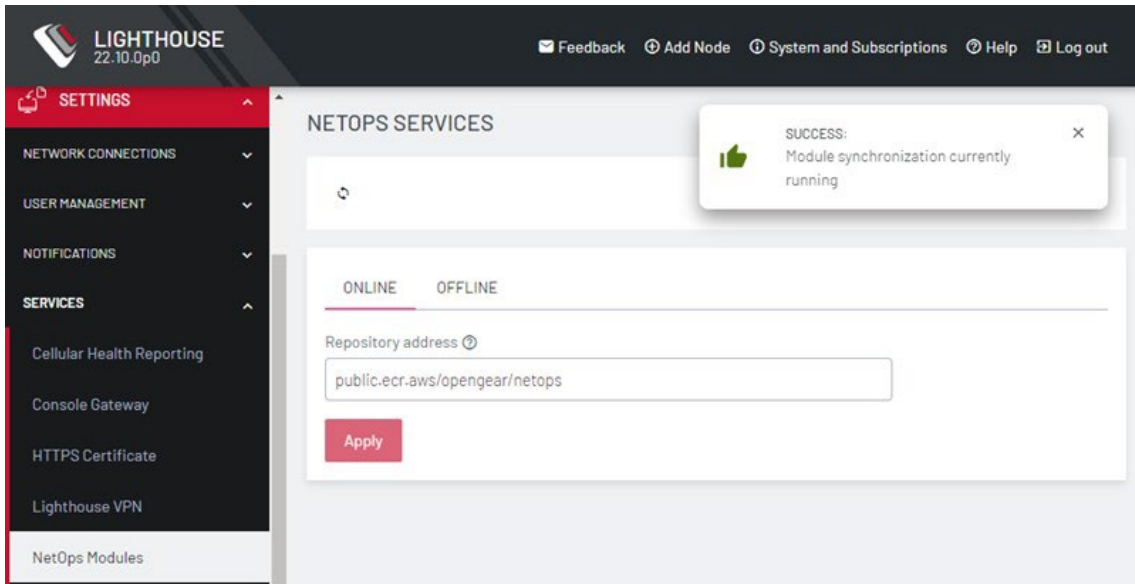
### PROCEDURE for Lighthouse UI

1. Log in to the Lighthouse web UI as a Lighthouse Administrator or the root user.
2. From the menu, select **SETTINGS > Services > NetOps Modules**. Select either Online or Offline.





- Click the  **Synchronize** icon. A message displays with the status.



- From the menu, select **CONFIGURE NODES > NetOps Modules > Manage Modules**.
- Click the **Redeploy** icon.

## PROCEDURE for Lighthouse CLI

Replace **root** and **default** with a Lighthouse Administrator or root credentials, then run the following:

```

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

IP Access is activated for both Enterprise Edition and Automation Editions.

Secure Provisioning and Automation Gateway are activated from the Automation Edition license.

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.

## Activate the NetOps Module on Lighthouse

### 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 **SETTINGS > System > Subscriptions**.

## PROCEDURE

1. Ensure you have a valid NetOps Module licence installed under **SETTINGS > 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 progress under **CONFIGURE NODES > NetOps Modules > Manage Modules**.
3. Refresh your browser and note new module-specific menu items are now available under **CONFIGURE NODES**, for example, **Secure Provisioning** or **IP Access**.

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. Check and apply **CONFIGURE NODES > NetOps Modules > Manage Modules > Module Name > Always Activate**.
2. To activate a new node, enroll the node into Lighthouse, , see "[Activate the Secure Provisioning Module on Lighthouse](#)" on page 411.

## Activate NetOps on Selected Nodes (Automatic)

You may selectively activate the module on a subset of nodes using Enrollment Bundles. Only nodes enrolling using one of these bundles will be automatically activated.

### PROCEDURE

1. Navigate to **CONFIGURE NODES > NetOps Modules > Manage Modules > *Module Name* > Always Activate**.
2. Uncheck **Always Activate**.
3. Click **Apply**.
4. Select **CONFIGURE NODES > Node Enrollment > Enrollment Bundles** and add a new bundle (you may also edit an existing bundle).
5. Enter a bundle **Name** and **Token**, and choose whether or not to **Auto-Approve** enrollment, see ["Activate a NetOps Module" on page 346](#).
6. Scroll down to **NetOps Modules** and add the **Module Name**.
7. Enroll the node to Lighthouse, specifying the **Enrollment Bundle Name** and **Token**.

**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)

### PROCEDURE

To activate nodes manually after enrollment, use the following steps:

1. Ensure **CONFIGURE NODES > NetOps Modules > Manage Modules > Module Name > Always Activate** is unchecked and applied.
2. Select **CONFIGURE NODES > Configuration Templating > Apply Templates**.
3. Under **NetOps Module Activation**, choose the **Module Name** and click **Next**.
4. Select the nodes to activate, then click **Next**.
5. **Refresh** to ensure preflight checking is successful, then click **Next**.
6. **Refresh** to ensure activation is successful.

**Note:** Under the NetOps *licensing* arrangement, once 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

To deactivate and remove a NetOps Module from a given node, 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), and *node\_id* is the internal node ID, for example, *nodes-1*.

Follow the procedure below to remove NetOps Modules from a given node:

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
```

## 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.

Managed devices 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 unroutable 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 all Operations Manager product lines (OM120x and OM22xx), and CM8100 currently, and will be included in future products as well.

## 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 managed devices, 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. Once 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 managed device 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.

## PROCEDURE

The basic steps to setup and use this feature are as follows:

1. Activate the **Automation Gateway** NetOps Module on an enrolled node, that is on the same network as the remote device that you wish to reach.
2. Log in to the 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.
3. Click **CONFIGURE > AUTOMATION GATEWAY > Devices**.
4. In the **FILTER BY** menu, select *HTTPS* or *HTTP* from the **Services** drop down.
5. 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.

6. To initiate an Automation Gateway session, click the device's web UI icon.

**DEVICE ACCESS**
Rescan Devices

DEVICE INFO

DISCOVERED SERVICES

No devices have been identified.

FILTER BY

Device Name

IP Address

Node Name

MAC Address

Services

7. 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.

8. 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.

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 Opendgear
# 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 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 **CONFIGURE > Automation Gateway > Devices**. The discovery process can be restarted using the "Rescan" button on the **Automation Gateway > Devices** 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 }'
```

## IP ACCESS

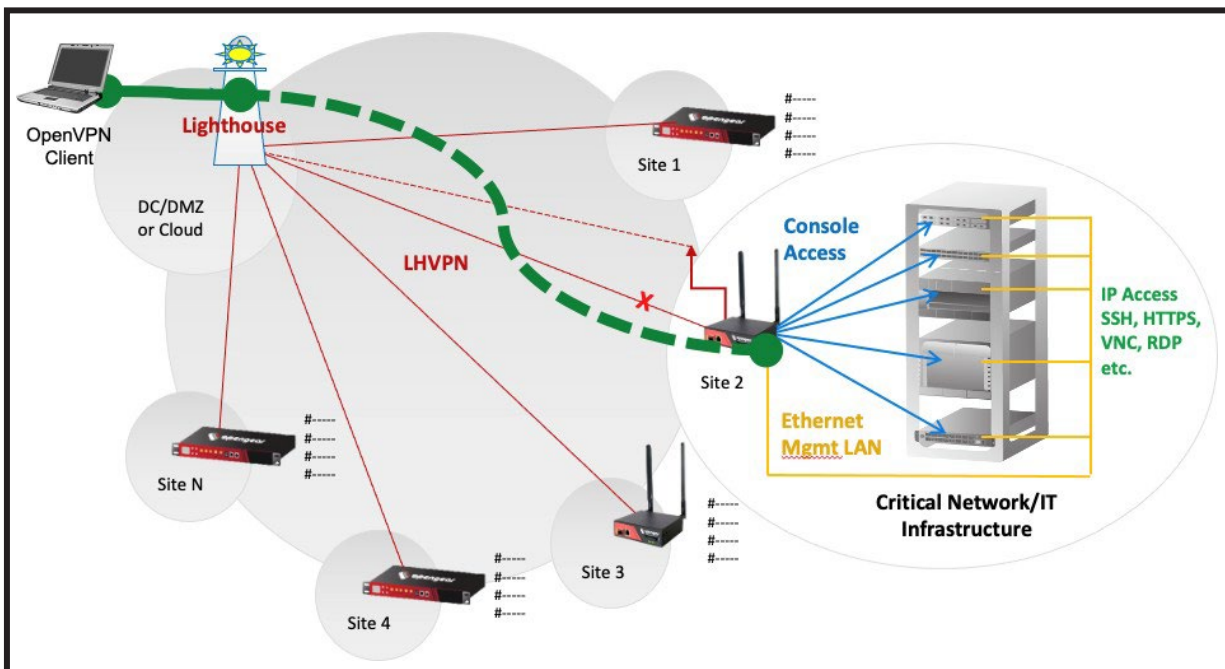
The Lighthouse IP Access feature allows an engineer to reach hosts on a remote site via an OpenVPN client through Lighthouse, over the Lighthouse VPN fabric, without physically traveling to the site. If IP Access is enabled for Lighthouse, it can be managed from using the **Configure > IP Access** menu option on the Lighthouse web UI.

IP Access adds client VPN capability to Lighthouse. Network engineers, firewall and server administrators can launch a VPN client connection to Lighthouse, be authenticated, then automatically connected to the remote site management network. The client PC has a secure VPN tunnel to the remote equipment the user needs to work on, providing the same TCP/IP access they would get if they traveled 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 (NGCS). 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 wish 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.

Other vendors/models are not currently supported.

Select **Configure > IP Access** to view the available nodes in Lighthouse. The **Node Access** page displays.


**opengear**  
 Lighthouse™ Central Management

[Feedback](#)
[Add Node](#)
[Help](#)
[System](#)
[Log out](#)

[Edit Managed Device Filters](#)

[CONFIGURATION TEMPLATING](#)

[MULTIPLE INSTANCE](#)

[NETOPS MODULES](#)

[DEVICE PROVISIONING](#)

[AUTOMATION GATEWAY](#)

**IP ACCESS**

[Node Access](#)

[Client Certificates](#)

[Client Status](#)

[Advanced Options](#)

[Network Access Policies](#)

## NODE ACCESS

Smart Group Filtering

Free Text Search

NAME	NODE ID	STATUS	ACTIONS
SPT-Management-IM7248	nodes-6	Enabled	
SPT-IM7216	nodes-7	Enabled	
SPT-CM7148	nodes-8	Enabled	
im7216-2	nodes-9	Enabled	



## Enable IP Access in Lighthouse

This topic walks through the steps required to activate and enable the IP Access feature.

## Enable NetOps Automation

After deploying the Lighthouse, sync the latest NetOps Modules from Docker Hub:

1. Log in to the Lighthouse web UI as a Lighthouse Administrator or the root user.
2. From the menu, select **SETTINGS > Services > NetOps Modules**.
3. Click the **Synchronize** icon.

**Note:** If Lighthouse cannot contact the Docker Hub, you may be able to use the offline installer.

## Activate the IP Access Module

NetOps Modules must be activated on Lighthouse and a per-node basis.

1. Log in to the Lighthouse web UI as root or a Lighthouse Administrator, and upload the Enterprise Edition or Enterprise Automation Edition licence file under **SETTINGS > System > Subscriptions > New**.
2. Click **CONFIGURE > NetOps Modules > Manage Modules** and wait until Lighthouse activation is complete.

To activate on the node you wish to access IP networks via, use the following steps:

1. Ensure **CONFIGURE > NetOps Modules > Manage Modules > IP Access > Always Activate** is unchecked and applied.
2. Select **CONFIGURE > Configuration Templating > Apply Templates**.
3. Under **NetOps Module Activation** select **IP Access** and click **Next**.
4. Select the nodes to to activate and click **Next**.
5. **Refresh** to ensure preflight checking is successful, then click **Next**.
6. **Refresh** to ensure activation is successful.

**Note:** Any locally attached subnet and any static routes configured on the Management LAN (OGCS) or interfaces in the LAN zone (NGCS) will get pushed to the client.

See also: ["Activate a NetOps Module" on page 346](#).

## Network Ports Used for IP Access

IP Access OpenVPN clients connect on UDP port 8194, inbound to Lighthouse.

The remainder of the connection is bridged over the existing Lighthouse VPN between the node and Lighthouse, therefore no additional ports are utilized.

## Generate a Certificate and Export Client Configuration

Clients connect to Lighthouse via an OpenVPN client, which in turn connects them to the Management LAN network of a particular node. IP Access provides a convenient means to configure the OpenVPN client by generating the configuration files that may be imported directly into your OpenVPN client of choice.

1. Log in to the Lighthouse web UI as root or a Lighthouse Administrator, and click **CONFIGURE > IP Access > Client Certificates**. Enter a **Certificate Name** and click **Create**.
2. When the certificate is created, download an associated OpenVPN client configuration by clicking **Export**.

**Note:** Deleting a client configuration file from Lighthouse revokes that client certificate and any associated client configurations using that certificate will no longer be permitted to connect.

## Connect the VPN Client

The final step is to establish the VPN connection that allows IP Access to the Management LAN (and optionally other networks) behind a node.

1. Import the client configuration from the previous step into your preferred OpenVPN client and start the VPN connection.
2. When prompted to authenticate the VPN connection, you must also specify your Lighthouse credentials and the node that you want to establish IP access via.
3. Specify the node by adding *:node-name* to your Lighthouse username, for example, authenticating with the username *james:my-acm7004-5* will authenticate as Lighthouse user *james* and connect the VPN to the IP network(s) behind *my-acm7004-5*.

**Note:** To be permitted connection, the Lighthouse user must have at least Node User rights for the specified node.

**Note:** The IP Access NetOps module creates a Layer 2 TAP mode tunnel which is not supported by Android or iOS operating systems.

## Password Authentication During VPN Connection

During VPN connection, the client is prompted to enter a username and password. These credentials are used to authenticate the user, and also to specify the remote node to establish IP access through.

Specify the node by adding *:node-name* to your Lighthouse username, for example, authenticating with the username *james:my-acm7004-5* will authenticate as Lighthouse user *james* and connect the VPN to the IP network(s) behind *my-acm7004-5*.

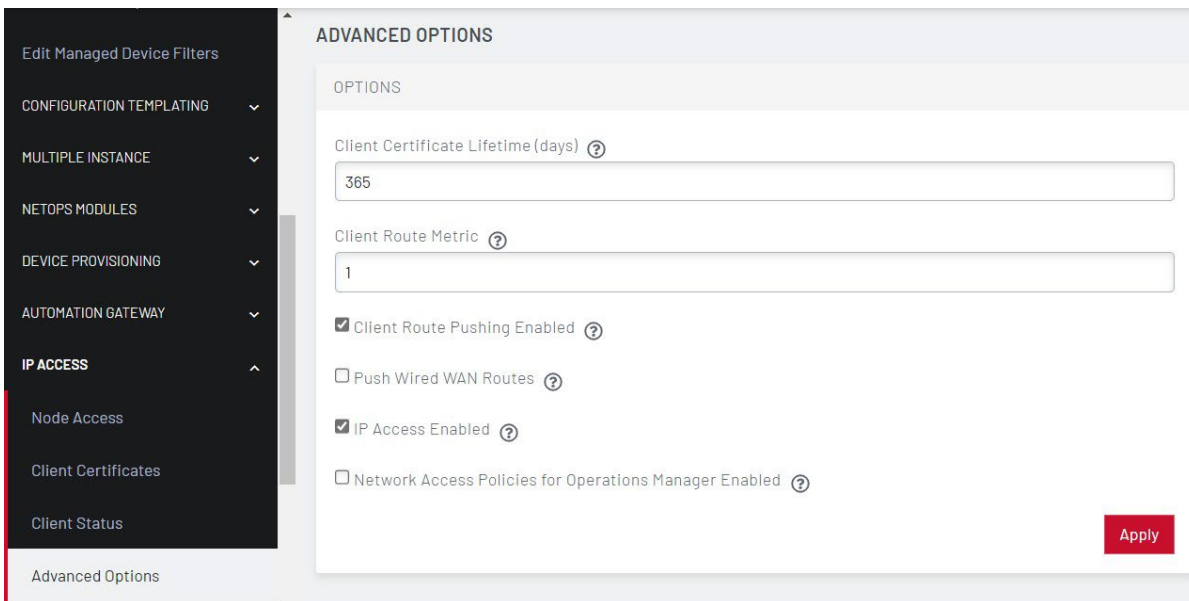
For authentication to succeed, the Lighthouse user must be one of:

- Lighthouse Administrator role, or root
- Node Administrator role with access to the node you are connecting through
- Node User role with access to the node you are connecting through

## 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.



Edit Managed Device Filters  
 CONFIGURATION TEMPLATING  
 MULTIPLE INSTANCE  
 NETOPS MODULES  
 DEVICE PROVISIONING  
 AUTOMATION GATEWAY  
**IP ACCESS**  
 Node Access  
 Client Certificates  
 Client Status  
 Advanced Options

### ADVANCED OPTIONS

OPTIONS

Client Certificate Lifetime (days) ?

Client Route Metric ?

☒ Client Route Pushing Enabled ?

☐ Push Wired WAN Routes ?

☒ IP Access Enabled ?

☐ Network Access Policies for Operations Manager Enabled ?

Apply



## Connecting to WAN Zone

By default, IP access connects the client to the management LAN zone.

If IP Access is required for the WAN zone, select **Configure > IP ACCESS > Advanced Options > Push Wired WAN Routes**. This is a global configuration, and will affect all node that are enabled for IP Access.

In this case, the customer must have deployed Opendgear 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.

**Note:** On NGCS this option connects the client to the WAN zone and any interfaces that belong to it, by default this is also NET1.

## 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 NGCS 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.

## Understanding Access Policies

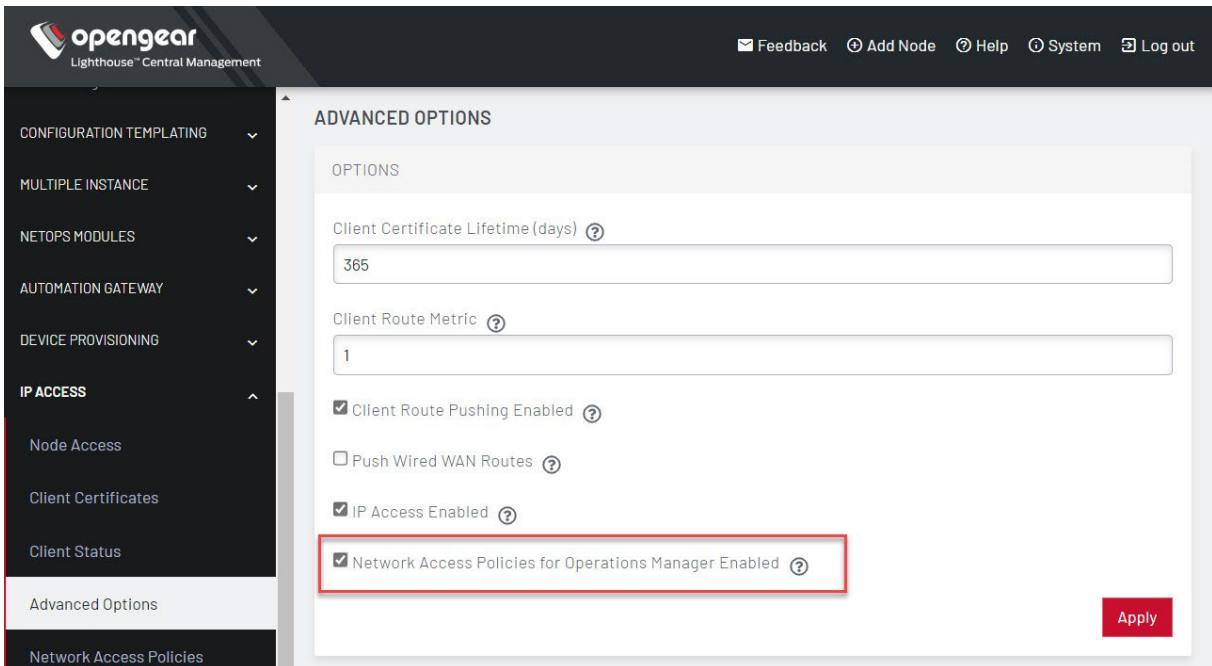
Putting it all together, when a user authenticates to Lighthouse, they are mapped into one or more group(s), which map into firewall zone(s), which allow authenticated users to reach the appropriate network interfaces(s), including switch ports or VLANs, via IP Access.

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 that 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.

## Setting up Network Access Policies

To enable this feature, go to **IP Access > Advanced Options** and select “Network Access Policies for Operations Manager Enabled”, then hit **Apply**.



**opengear**  
Lighthouse™ Central Management

Feedback Add Node Help System Log out

**CONFIGURATION TEMPLATING** ▾  
**MULTIPLE INSTANCE** ▾  
**NETOPS MODULES** ▾  
**AUTOMATION GATEWAY** ▾  
**DEVICE PROVISIONING** ▾  
**IP ACCESS** ▴  
 Node Access  
 Client Certificates  
 Client Status  
 Advanced Options  
 Network Access Policies

**ADVANCED OPTIONS**

OPTIONS

Client Certificate Lifetime (days) ?  
 365

Client Route Metric ?  
 1

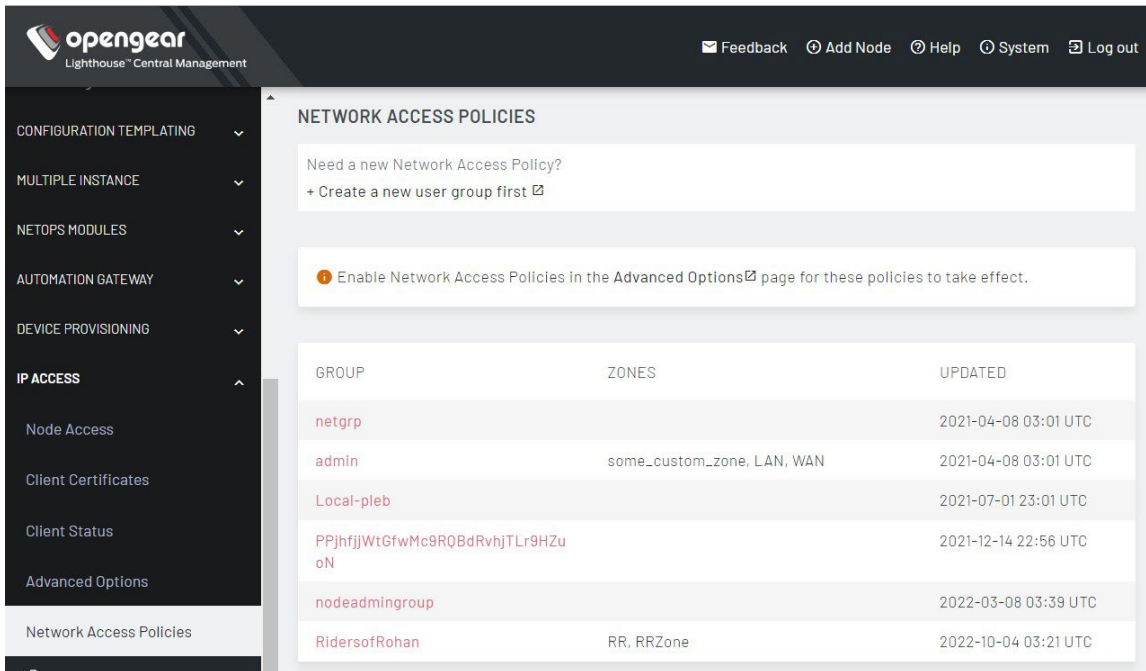
☒ Client Route Pushing Enabled ?  
☐ Push Wired WAN Routes ?  
☒ IP Access Enabled ?  
☒ Network Access Policies for Operations Manager Enabled ?

Apply

## To set Access policies

A policy must be configured for each group whose members will use IP Access.

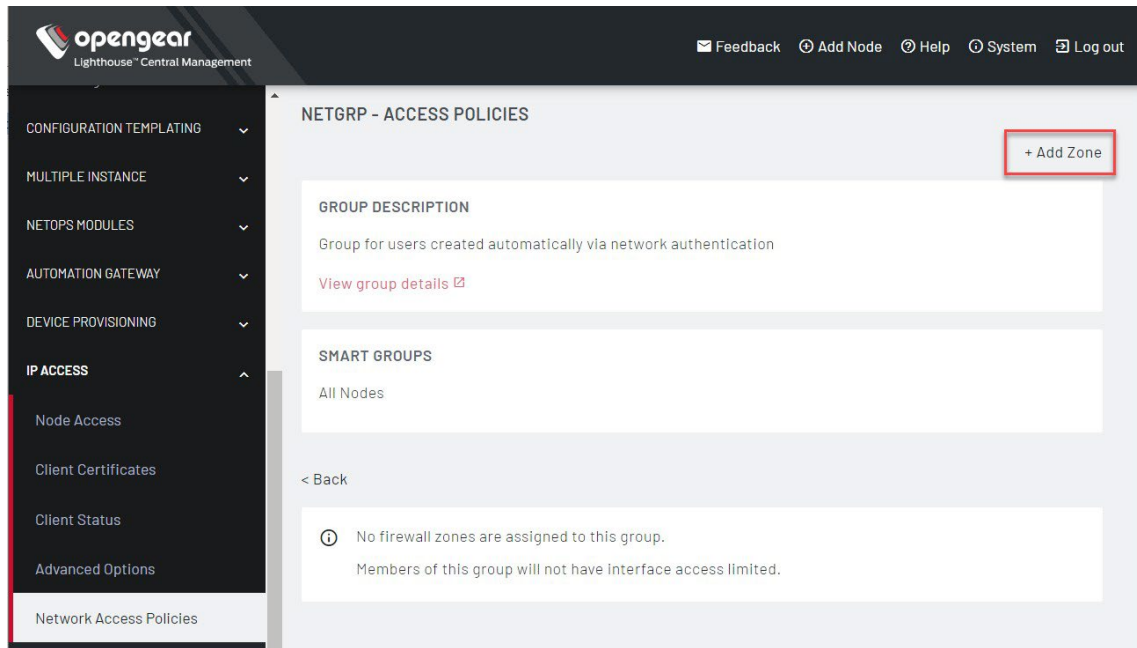
1. Select **Configure > IP Access > Network Access Policies**. The **Network Access Policies** page displays. The group to zone mapping column names **ZONES** is empty by default.



GROUP	ZONES	UPDATED
netgrp		2021-04-08 03:01 UTC
admin	some_custom_zone, LAN, WAN	2021-04-08 03:01 UTC
Local-pleb		2021-07-01 23:01 UTC
PPjhjjjWtGfwMc9RQBdRvhjTLr9HZuoN		2021-12-14 22:56 UTC
nodeadmingroup		2022-03-08 03:39 UTC
RidersofRohan	RR, RRZone	2022-10-04 03:21 UTC

2. Click on the group name to edit the group policy.

- Click on **+ Add Zone** to add one or more firewall zones for this group.



- Select the firewall zone and click Add.

**ADD FIREWALL ZONE**

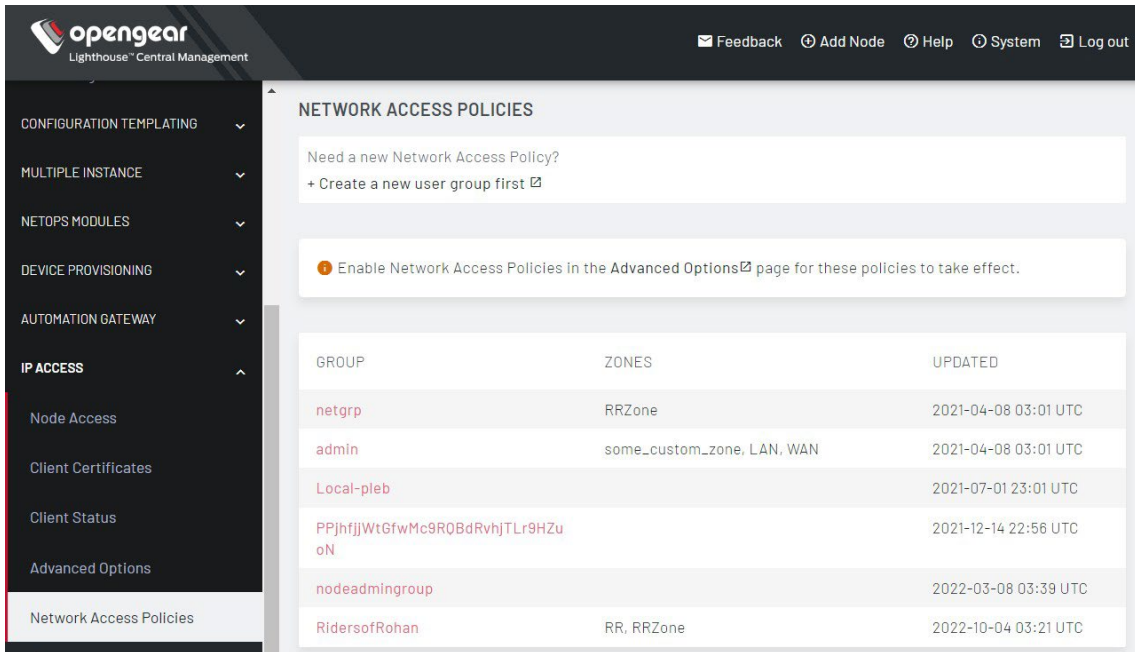
NAME	NODES	
<input type="checkbox"/> LAN	1	<a href="#">View details</a>
<input type="checkbox"/> Lighthouse VPN	1	<a href="#">View details</a>
<input type="checkbox"/> WAN	1	<a href="#">View details</a>
<input type="checkbox"/> Cellular	0	<a href="#">View details</a>
<input type="checkbox"/> RR	0	<a href="#">View details</a>
<input type="checkbox"/> RRZone	0	<a href="#">View details</a>

+ Manually specify zone

Cancel

Add

- The Network Access Policies page now displays the group with the Firewall zone.



**opengear**  
Lighthouse® Central Management

Feedback Add Node Help System Log out

**CONFIGURATION TEMPLATING**

**MULTIPLE INSTANCE**

**NETOPS MODULES**

**DEVICE PROVISIONING**

**AUTOMATION GATEWAY**

**IP ACCESS**

Node Access

Client Certificates

Client Status

Advanced Options

**Network Access Policies**

### NETWORK ACCESS POLICIES

Need a new Network Access Policy?  
+ Create a new user group first

Enable Network Access Policies in the Advanced Options page for these policies to take effect.

GROUP	ZONES	UPDATED
netgrp	RRZone	2021-04-08 03:01 UTC
admin	some_custom_zone, LAN, WAN	2021-04-08 03:01 UTC
Local-pleb		2021-07-01 23:01 UTC
PPjhjjWtGfwMc9RQbDRvhjTLr9HZuoN		2021-12-14 22:56 UTC
nodeadmingroup		2022-03-08 03:39 UTC
RidersofRohan	RR, RRZone	2022-10-04 03:21 UTC



## Accessing 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 Smart Group matching. Note that this works regardless of Local or Remote user authentication on Lighthouse; user authorization (which determines access to Nodes and managed devices, and now firewall zones) is always derived from group membership.

## Firewall Zones

In the **IP Access Network Access Policy** settings, 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 NGCS 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.

## Troubleshooting 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
```

```
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.

## Using 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
```

The Routing table displays:

```
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
```



```
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 route 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.

### Examples

```
ping 192.168.2.4
curl -k https://192.168.2.4/
```

## 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 managed devices 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.

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

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

## 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:

- . Eliminates config drift
- . Enforces config reproducibility
- . Central audit trail of all configuration changes
- . Disaster recovery becomes as simple as resetting all devices to reprovision

### Cons:

- . Requires a longer maintenance window as the device is reset and reboots
- . Patches cannot be applied to running configuration

## 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 managed device configuration files and firmware images, and the node provisioning (ZTP) services required to deliver the files to managed devices.

Secure Provisioning is configured by defining the resources to provision managed devices with, and defining how these resources should be distributed around your network.

- **Device Resource Bundles** contain the files needed to provision one or many managed devices:
  - **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**:
  - A **Static Node Inventory List** - a predefined, static list of nodes to distribute to
  - A **Dynamic Node Inventory List** - evaluates a Smart Group 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 Managed Devices 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
- Opendaylight

Advanced users may add support for additional devices using custom DHCP configuration.

## 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 managed devices, 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:

## 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 managed device. This may be a full, final configuration, or a baseline configuration to allow the managed device 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 managed devices of the selected Device Type. Bundles may be targeted to specific managed devices by specifying one or more device MAC addresses (including range and reverse match), or in some case by specifying one or more device serial numbers.



## 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 Smart Group of nodes

**Note:** You may combine distribution methods.

*Topic continued on next page:*

## Create Device Configuration

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

For example, 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
   (__)  | |   Opendgear NetOps Automation
   (__)  | |
   (__)__|. |

EOF
!
end
```

A 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          I
  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
   (__)  |  |          Opendgear NetOps Automation
   (__)  |  |
   (__)_.|_|
^
!
end
```

## Activate the Secure Provisioning Module on Lighthouse

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.

Installing the Secure Provisioning license automatically activates Secure Provisioning on Lighthouse, at which point the NetOps Automation platform installs the central Secure Provisioning software components on Lighthouse.

- Install the Enterprise Automation Edition license under **SETTINGS > System > Subscriptions**.
- Install an applicable legacy license, or, apply an Automation Edition subscription, to enable Secure Provisioning under **SETTINGS > System > Subscriptions**.
  - It will take a few minutes for the Secure Provisioning to activate on Lighthouse, view progress under **CONFIGURE NODES > NetOps Modules > Manage Modules**.
- Refresh your browser and note new menu items are now available under **CONFIGURE > Secure Provisioning**.

Secure Provisioning may now be selectively activated on nodes automatically as they enroll, or activated on nodes manually after enrollment.

## Install the Node

1. Connect the NET1 Ethernet to a network port via which node can reach the Lighthouse VM Connect power to the node.
2. By default, the node requests a DHCP address and has a static address of 192.168.0.1/24.
3. Test you can reach the node address via ping, SSH and HTTPS, and note this address for the following step.

## Configure a Per-node Module Activation Policy

The process of automatically or manually activating Secure Provisioning on a node prepares it to become a Secure Provisioning server, securely over Lighthouse VPN.

Activating a node for Secure Provisioning consumes an activation from the license. Deactivation returns the activation to the available pool. To deactivate and remove a NetOps Module from a given node, see ["Deactivate \(remove\) a NetOps Module" on page 352](#).

**Note:** Operations Manager activation deploys the Secure Provisioning container to the node, which may take several minutes.

## Option A. Automatically Activate All Nodes Upon Enrollment

This is the default policy. When a license is present and activations are available, all nodes are activated for Secure Provisioning as they enroll. Nodes that have been previously enrolled must be manually activated.

1. Ensure CONFIGURE > NetOps Modules > Manage Modules > Secure Provisioning > Always Activate is checked and applied.
2. To activate a node, enroll it into Lighthouse.

## Option B. Automatically Activate Select Nodes Upon Enrollment

You may selectively activate Secure Provisioning on a subset of nodes using Enrollment Bundles. Only nodes enrolling using one of these bundles will be automatically activated.

1. Uncheck **CONFIGURE > NetOps Modules > Manage Modules > Secure Provisioning > Always Activate** and click **Apply**.
2. Select **CONFIGURE > Node Enrollment > Enrollment Bundles** and add a new bundle (you may also edit an existing bundle) Enter a bundle **Name** and **Token**, and choose whether or not to **Auto-Approve** enrollment.
3. Scroll down to **NetOps Modules** and add **Secure Provisioning** then **Apply**.
4. When enrolling the node to Lighthouse, specify the **Enrollment Bundle Name** and **Token**.

**Note:** Lighthouse-initiated manual enrollment (i.e. clicking the **Add Node** button in the Lighthouse web UI) does not support bundles, you must use a node-initiated enrollment method.



## Option C. Manually Activate Nodes After Enrollment

1. Select **CONFIGURE NODES > Configuration Templating > Apply Templates**.
2. Under **NetOps Module Activation**, select **Secure Provisioning** and click **Next**. Select the nodes to activate and click **Next**.
3. Refresh, to ensure preflight checking is successful, then click **Next**.
4. Refresh, to ensure activation is successful.

## Enroll the Node Into Lighthouse

1. Launch an HTTPS browser session to Lighthouse.
2. Login using root and the secure password set earlier.

**Tip:** You may also login as a Lighthouse Administrator user, if you have configured one.

3. At the top of the UI, click **Add Node**.
4. Select An Opendgear appliance, the second option in the **Product** dropdown list.
5. Enter the Operations Manager's Network Address, Username (root) and Password (default) **Check Auto-approve node** then **Apply**.
6. From menu, select **CONFIGURE NODES -> Node Enrollment -> Enrolled Nodes** then click the **Refresh** icon to check enrollment has completed.
7. If you are using the Automatically activate select nodes upon enrollment policy, you must manually activate the node after enrollment.

The node now has a secure Lighthouse VPN (OpenVPN) tunnel back to Lighthouse, over which all communications are now secured.

## Connect Target Device

Secure Provisioning currently supports provisioning devices from these vendors:

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

**Note:** Additional devices may be supported using custom DHCP configuration.  
To request built-in support for additional devices, contact customer support.

## PROCEDURE

1. Connect a supported managed device'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 managed device.
5. Ensure the managed device is in ZTP mode, this typically requires the device to have its configuration erased/reset to factory defaults.

## UI-based Workflow

Each NetOps Module provides a simple web UI for configuration and status monitoring. This UI is designed primarily for manual operation, evaluation and testing. For comprehensive automation, refer to the CLI-based workflow section that follows. It may be useful to familiarize yourself with the system using the UI before adopting a CLI-based workflow.

**Note:** 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.





## UI Workflow

Launch an HTTPS browser session to Lighthouse Login using root and the secure password set earlier.

**Tip:** You may also login as a Lighthouse Administrator user, if you have one configured

## Create Device Resource Bundle

1. In the menu, navigate to **CONFIGURE > Secure Provisioning > Device Resources**.

DEVICE RESOURCE BUNDLES					
<div>⚠ There are resource changes that have not yet been pushed to nodes. <button>Push now</button></div>					
<div>⊕ <span>⌂</span></div>					
NAME	DEVICE TYPE	RESOURCES	PROVISION AFTER	PUSH TO NODES	ACTIONS
example_router	cisco_xr	Configuration File: demo_xrv.cfg		ExampleDynamic	 
example_switch	arista	Configuration File: demo_arista.cfg		ExampleStatic	 

- Click the **Add (+)** button. The **Modify Device Resource** page opens.

## MODIFY DEVICE RESOURCE

### DEVICE RESOURCE DETAILS

Name

Device Type

Configuration File ?

No file chosen

Image File ?

No file chosen

MAC Addresses ?

Serial Numbers ?

### ADVANCED

*Advanced features are only available on Operations Manager nodes.*



3. Choose a **Name** to identify this bundle, for example, demo\_bundle1.
4. Select the **Device Type** corresponding to a target managed device.

**Note:** Each device type may display different resource fields documented below, this is dependent on the device-specific ZTP featureset

5. The **Configuration File** is the initial configuration file for the device to load via ZTP. Select a previously uploaded file or click the **Browse** button to upload a new file.
6. The **Image File** is the initial software image for the device loaded via ZTP. Select a previously uploaded **Image File** or click the **Browse** button to upload a new file.
7. Optionally, target this bundle at devices matching the specified MAC Addresses, click the **Add** and **Remove** buttons when specifying multiples.

**Note:** Each MAC address is 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)

8. Optionally, target this bundle at devices matching specified **Serial Numbers**, click the **Add** and **Remove** icons when specifying multiples.
9. Click **Save**.

## Define Resource Distribution

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

## Define a Static Node Inventory

1. Navigate to **CONFIGURE > Secure Provisioning > Resource Distribution**.
2. Under **Static Node Inventory List** click the **Add +** button. The **New Node Inventory** window opens, displaying inventory details.
3. Choose, and enter a Name to identify this bundle, for example, `branchinventory_`
4. From the Resource Distribution list, select the bundles to distribute.

NEW NODE INVENTORY

INVENTORY DETAILS

Name

Resource Distribution

<input type="checkbox"/>	NAME	DEVICE TYPE	RESOURCES
<input type="checkbox"/>	example_router	cisco_xr	Configuration File: demo_xrv.cfg
<input type="checkbox"/>	example_switch	arista	Configuration File: demo_arista.cfg

FILTERING

Smart Group Filtering

Free Text Search

5. Select the **Nodes** to which these bundles are to be distributed.

**Note:** The **Free Text Search** and **Smart Group** filter options are to help locate nodes for inclusion in the static inventory, these filters are *not* applied dynamically to the inventory going forward.



**LIGHTHOUSE**  
Central Management

FILTERING

Smart Group Filtering

Select Smart Group...

Free Text Search

Select Nodes

<input type="checkbox"/>	NODE ID	NAME	INFO
<input type="checkbox"/>	nodes-2	nodes-2	MODEL OM2216 FIRMWARE VERSION 21.Q3.0 ENROLLMENT BUNDLE Global MANAGEMENT VPN ADDRESS 192.168.128.3 NET1 MAC ADDRESS 00:13:c6:08:0e:1a SERIAL NUMBER 22162005085943

Cancel

Apply

6. Click **Apply**.

## Define a Dynamic Node Inventory

1. Create a Lighthouse Smart Group, as documented in the *Creating Smart Groups* topic in the [Lighthouse User Guide](#).
2. In the Lighthouse UI, navigate to **CONFIGURE > Secure Provisioning > Resource Distribution**.
3. Under **Dynamic Node Inventory List** click the **Add +** button. The **Modify Node Inventory** window opens, displaying inventory details.
4. Choose, and enter a Name to identify this bundle, for example, LabInventory\_
5. Select the **Smart Group** to link to this inventory,

**Note:** this **Smart Group** search is dynamically evaluated to a list of nodes each time resources are pushed.

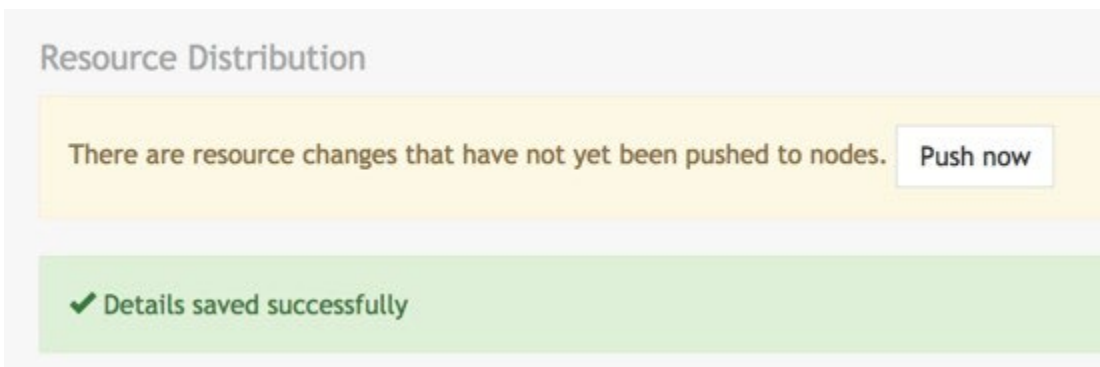
6. Select the bundles to distribute, listed in the **Resource Distribution** table.
7. Click **Apply**.

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

## Push Resources

Any changes to the resource bundle or distribution configuration made in the previous steps are not applied to nodes immediately, they must be explicitly pushed to nodes.

When changes are detected, a **Push now** button is displayed above the Secure Provisioning UI pages.



Resources may always be resynchronized by clicking **CONFIGURE > Secure Provisioning > Resource Distribution > Push Resources**.

The push operation distributes resource bundles as configured, then as needed restarts DHCP services on remote nodes. Push status is displayed above the Secure Provisioning UI pages.

You may also monitor the syslog output on Lighthouse for low-level output from the Ansible playbook.

## 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 i.e. 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 managed devices without needing 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 .yaml or .yml extension, using the following format:

#### **nom-prov.yml**

```
device_resources:  
  
  demo_arista:  
  
    device_type: arista  
  
    config_file: 'demo_arista.cfg'  
  
    image file: 'arista eos.sw1'
```



```
node_inventory:
  MyNodes:
    static:
      - nodes-1

deployment:
  MyNodes:
    - demo_arista
```

**Note:** Note that 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 one or more resources, ie. *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 Smart Group, as documented in the *Creating Smart Groups* section of the *Lighthouse User Guide*.
2. Create a key named **smartgroup** with a value of the Smart Group name, this Smart Group search is dynamically evaluated to a list of nodes each time resources are pushed

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

- 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 > Smart Group

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 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.

Once 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

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 managed device 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 once a Managed Device is considered provisioned. A device is considered to be in a provisioned state once 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](#)" on page 430.

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 **sshpas** command.

## Monitor the ZTP Progress of a Managed Device

The current provisioning state of managed devices 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"]

Sample messages showing an Opendgear ACM7004 device being provisioned:

```
[NetOps-DOP device="00:13:C6:EF:00:08"] Received DHCP request from  
device with vendor ID Opendgear/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:

- . 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

## Using 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 needs 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.

### Available variables:

- . {{ nom\_remote\_server }}
- . Address of provisioning interface on the node
- . Example: 10.0.0.1
- . {{ nom\_remote\_interface }}
- . Name of provisioning interface on the node
- . Example: net2
- . {{ nom\_remote\_netmask }}
- . Netmask of provisioning interface on the node (and netmask assigned in DHCP offers)
- . Example: 255.255.255.0
- . {{ nomremotenetmaskcidr }}\_
- . CIDR format netmask (prefix length) of provisioning interface on the node
- . Example: 24



- . {{ nom\_remote\_ntp\_server }}
- . Address of NTP server assigned in DHCP offers (same as nom\_remote\_server)
- . Example: 10.0.0.1
- . {{ nom\_remote\_dns\_server }}
- . Address of DNS server assigned in DHCP offers (same as nom\_remote\_server)
- . Example: 10.0.0.1
- . {{ nom\_device\_ipv4\_address }}
- . This feature is only supported by Operations Manager nodes.
- . DHCP address assigned to target device
- . Example: 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.
- . Example: ssh-rsa  
AAAAB3NzaC1yc2EAAAADAQABAAQDHO50MVT4A9NI7vjDIS+  
76LgDBtEDm+J1gaQPNiIV62CGA6cm5wASDiddY/KZkHA
- . root@platypus.employee.bne.opengear.com (*example only*).

- {{ nom\_device\_hostname }}
- This feature is only supported by Operations Manager nodes.
- Hostname advertised by target device
- Example: router
- {{ nom\_device\_mac\_address }}

**Note:** This feature is only supported by Operations Manager nodes.

- MAC Address of target device
- Example: 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 once a Managed Device is considered provisioned. A device is considered to be in a provisioned state once 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 nomremotessshpubkey 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 otherwise 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 dependent 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 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 dependent 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
```

## Troubleshooting Secure Provisioning

Secure Provisioning consists of several Docker containers:

- The central-dop container runs on Lighthouse, hosting git repository
- The dop-ui container runs on Lighthouse, serving the Secure Provisioning web UI
- The remote-dop 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.

*See the troubleshooting commands on the following page.*

## Troubleshooting Commands

The following are useful commands for troubleshooting issues with the NetOps modules:

Command
<code>docker ps</code>
View running Docker containers.
<code>docker exec -ti container-name bash</code>
Spawn a bash shell inside a container.
<code>docker logs container-name</code>
View logs of a container.
<code>docker exec -ti deployment ansible-playbook -vvv /ansible/dop_2.0.0.yml</code>
Manually run module deployment in verbose mode, on Lighthouse.
<code>/etc/scripts/post-receive</code>
Manually push ZTP resources from Lighthouse to Operations Manager nodes (inside central-dop container).
<code>/etc/scripts/netops_ui_handler</code>
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.



## Lighthouse CLI, Serial Port and REST API logging

Lighthouse offers command line interface (CLI) and REST API logs.

**Note:** Logging is disabled by default.

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

When you enable logging, you do not need to restart or log out and in again. There are a few caveats:

- CLI logging only works for Interactive (human-controlled) terminals. Commands generated by automated scripts will be logged.
- Commands such as `ssh` or `telnet` do not produce logs for the commands sent over the connection.
- Requests can be logged for all endpoints, however, only endpoints implemented in Lipy can have responses logged.

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

## Using ogconfig-cli to enable logging

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

- `system.logging_cli_enabled` - Enable/disable logging commands entered in the Lighthouse Terminal.
- `system.logging_rest_enabled` - Enable/disable basic logging for the REST API. These logs report the following information about every REST API call:

- Time
- Request type (GET/POST/PUT/DELETE)
- HTTP status code
- Username
- Source IP Address
- Endpoint

`system.logging_rest_request_enabled` - Enable/disable request logging for every REST API call. In addition to the basic logging, also logs the request body that was provided by the client, if any.

: Requires `system.logging_rest_enabled` to be enabled.

## Add node and port to Lighthouse logs

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

When system logging is enabled, user and node selection and user and Port selection is logged. To enable node and port logging on Lighthouse:

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

```
Oncroot@lighthouse:~# ogconfig-cli
ogcfg> set system.logging_cli_enabled true
root-1-system_logging_cli_enabled: Integer <True> ogcfg> push
OK
ogcfg> exit
```

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

## Example logs

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

```
2020-03-17 15:29:37,237 INFO [root:117][waitress] POST 400 (root |  
192.168.1.1) - /api/v3.4/system/licenses/file  
2020-03-17 15:30:23,034 INFO [root:117][waitress] GET 200 (root |  
192.168.1.1) - /api/v3.4/users?page=1&per_page=10
```

Here is an example of logs with request or response logging

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

```
2020-05-11T05:45:09.567214+00:00 lighthouse rest_api_log[2465]: PUT  
200 (root | fd07:2218:1350:4b:a438:f8ff:fe4f:65fc) -  
/api/v3.4/system/cli_session_timeout REQUEST={"system_cli_session_  
timeout":{"timeout":0}}  
2020-05-11 05:45:18,999 INFO [lipy.logging.rest_api:62][waitress] GET  
200 (root | fd07:2218:1350:4b:a438:f8ff:fe4f:65fc) -  
/api/v3.4/users?page=1&per_page=10 RESPONSE={'users': [{'username':  
'root', 'description': 'System wide SuperUser account', 'enabled':  
True, 'id': 'users-1', 'no_password': False, 'expired': False,  
'locked_out': False, 'rights': {'delete': True, 'modify': True},  
'groups': ['groups-2']}], 'meta': {'total_pages': 1}}
```

## Enable logging

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

```
root@lighthouse:~# ogconfig-cli
ogcfg> set <value> true
root-1-<value>: Integer <True>

ogcfg> push
OK
ogcfg> exit
```

Replace <value> with the desired setting:

- system.logging\_cli\_enabled
- system.logging\_rest\_enabled
- system.logging\_rest\_request\_enabled (requires system.logging\_rest\_enabled)
- system.logging\_rest\_response\_enabled (requires system.logging\_rest\_enabled)

### To check if logging is enabled:

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

```
root@lighthouse:~# ogconfig-cli ogcfg> print system
```

This will produce output with Boolean values:

- system.logging\_cli\_enabled (bool): false
- system.logging\_rest\_enabled (bool): false

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

## Disable logging

```
root@lighthouse:~# ogconfig-cli ogcfg> set <value> false
root-1-<value>: Integer <False>
ogcfg> push OK
ogcfg> exit
```

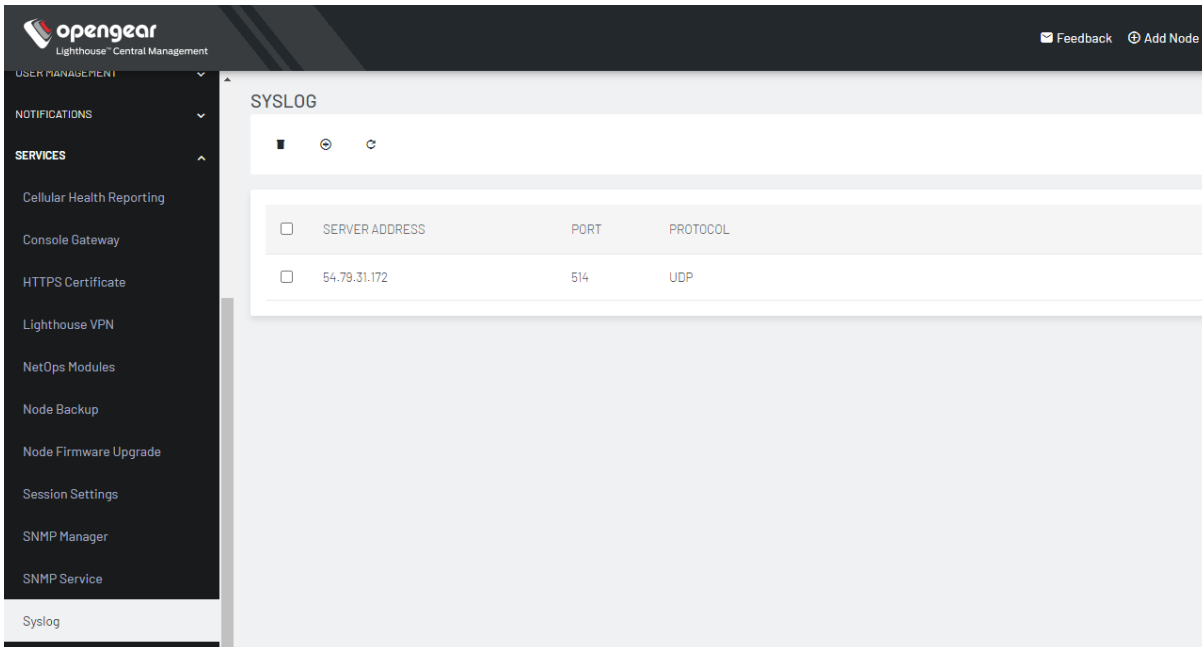
Replace <value> with the desired setting:

- system.logging\_cli\_enabled
- system.logging\_rest\_enabled
- system.logging\_rest\_request\_enabled
- system.logging\_rest\_response\_enabled

## Syslog Export

Administrative users can specify multiple external servers to export the syslog to via TCP or UDP.

Select **SETTINGS > SERVICES > Syslog**.



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



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

ADD EXTERNAL SYSLOG SERVER

Server Address ?

Protocol ?

Port ?

Cancel

Apply

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

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

## Command line tools

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

1. Select **MANAGE > LIGHTHOUSE > Local Terminal**. A login prompt displays



2. Enter an administrator's username and press Return.
3. A password: prompt displays. Enter the administrator's password and press Return.
4. A bash shell prompt displays.

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

Lighthouse-specific shell-based tools are listed below.

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

## Example `node-command` Output

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

```
14:08:33 [SUCCESS] BNE-R02-IM7216 192.168.128.3:22  
OpenGear/IM72xx Lighthouse 3b90d826 -- Tue Jul 5 13:42:16 EST 20167
```

## node-info

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

## Example node-info output

```
$ node-info -A
BNE-R01-ACM7004-5
  address: 192.168.128.2
  id: nodes-1
  ssh port: 22
  description: Brisbane Rack 1
  Enrollment status: Enrolled
  connection status: Connected
BNE-R02-IM7216
  address: 192.168.128.3
  id: nodes-2
  ssh port: 22
  description: Brisbane Rack 2
  Enrollment status: Enrolled
  connection status: Connected
```

## node-upgrade

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

`node-upgrade` accepts the following arguments:

Short Argument	Long Argument	Description
-h	--help	Display usage information and exit
-q	--quiet	Suppress log messages
-V	--verbose	Display logs generated while upgrading
-l	--list-nodes	Display nodes and their upgradeable paths without executing upgrade
-D	--debug	Display detailed log messages, implies --verbose
-I	--ignore-version	Ignore firmware version warnings for upgrade
-i	--node-id=<id>	Select node by config ID
-n	--node-name-e=<name>	Select node by name
-a	--node-address-s=<address>	Select node by VPN address
-A	--all	Select all available nodes
-p	--product=<family>	Select node by product family

Short Argument	Long Argument	Description
-g	--smart-group=<name>	Select nodes by smartgroup filter
-f	--firmware-dir-r=<directory>	The directory of the firmware files(s)
-F	--firmware-file-e=<path>	The firmware image to use for upgrade
-v	--version=<version>	The firmware version to upgrade to

## An example node-upgrade run

The following is an example node-upgrade command. It sets `/mnt/data/nvram/latest-firmware` as the directory node-upgrade looks to for the firmware image used as the source for all the firmware upgrade attempts. Every console server being managed from the active Lighthouse instance is targeted for an upgrade and the target console servers are set to upgrade to firmware 4.11.0.

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

```
NODE (UUID) MODEL FAMILY ADDRESS VERSION RESULT
```

```
-----
-----
```

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

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



## Results and Error Messages in node-upgrade

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

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

In addition, the following exit statuses may be returned.

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

## cron

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

### Usage:

```
crontab [options] file
```

```
crontab [options]
```

```
crontab -n [hostname]
```

### Options:

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

To perform start/stop/restart on `crond` service:

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

To verify the current `crond` status:

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

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

```
crontab -l
```

To edit or create a custom crontab file:

```
crontab -e
```

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

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

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

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

When finished, save and close the `crontab` file.

## sysflash

`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.

Basic syntax is as follows:

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

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

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

`-b, --board-name <name>` Override board name (currently `lighthouse-vm`)

`-B, --board-revision <version>` Override board revision (currently `1.0`)

`-V, --vendor <vendor>` Override vendor (currently `opengear`)

`-l, --no-version-check` Do not check software version for upgradability

`-m, --no-migration` Do not migrate current config. Start fresh.

`-v, --verbose` Increase verbosity (may repeat)

`-o, --no-boot-once` Do not modify bootloader (implies `--no-reboot`)

`-r, --no-reboot` Do not reboot after upgrading

`-h, --help` Print this help

## Glossary

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

Term	Definition
AUTHDOWNLOCAL (RADIUS/LDAP/AAA)	When this authentication option is selected, if remote authentication fails because the user does not exist on the remote AAA server, the user is denied access.
AUTHLOCAL (RADIUS/LDAP/AAA)	When this authentication option is selected, if remote authentication fails because the user does not exist on the remote AAA server, Lighthouse tries to authenticate the user using a local account.
CELLULAR HEALTH	Status of the cellular connection of a node.
DARK MODE	Changes the user interface to display mostly dark colors, reducing the light emitted by device screens.
DOCKER	<p>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.</p> <p>Docker powers the NetOps platform within the Lighthouse product.</p>
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.
LIGHT MODE	Changes the user interface to display mostly light colors. This is the default UI setting.
LIGHTHOUSE	System for accessing, managing and monitoring Opendgear console servers.
LIGHTHOUSE ENTERPRISE	Offers an elevated centralized management solution with additional functionality. It supports growing trends such as edge computing and SD-WAN with High Availability and Remote IP Access.
LIGHTHOUSE VPN	The OpenVPN based connections that the Lighthouse instance has with the nodes it is managing
LOCALAUTH (RADIUS/LDAP/AAA)	When this authentication option is selected, if local authentication fails, Lighthouse tries to authenticate the user using a remote AAA server.
MANAGED DEVICE	A device that is managed via a node through a

	serial, USB, or network connection.
MULTIPLE INSTANCE	Access nodes through multiple Lighthouse instances at the same time.
NODE	A device that can be enrolled with Lighthouse, allowing it to be accessed, managed, and monitored. Currently, Opengear console servers are supported on a standard license, with support for other vendors Console Servers available as an add-on.
PASSWORD POLICY	Administrative users can define rules for Lighthouse user passwords including length, types of characters, reuse, and expiration period.
PENDING NODE	A node that has been connected to Lighthouse and has been configured with a VPN Tunnel, but which has not yet been approved for access, monitoring, or management. The approval operation can be automated by configuring Lighthouse to auto-approve nodes.
PRIMARY INSTANCE	The main instance of Lighthouse used for updating configuration and node enrollment.
REMOTE LOGGING/REMOTE SYSLOG	The ability to send logs to a remote server, for the offsite storage and review of logs.
REPLICATION	Automatic copying of the primary Lighthouse database to any connected dependent instances. Replication ensures that these instances mirror the same information and maintains connections to the same nodes.
ROLE	A set of access rights for a particular group. Three roles are defined within Lighthouse: Lighthouse Administrator, Node Administrator, and Node User.

SECONDARY/DEPENDENT INSTANCES	Redundant instances of Lighthouse that are used to access Lighthouse information and connected nodes.
SMART GROUP	Dynamic filter used to search for particular nodes, or for defining the access rights of a group of users. Smart Groups use node properties, as well as tags defined by users.
TAG	User-defined attribute and value that is assigned to one or more nodes. Tags are used when creating Smart Groups for filtering views or access to nodes.