opengear CONSOLE MANAGER

User Guide
Software Release 24.11.1

CONTENTS

Contents **2**

Copyright © **16**

Document Revision History **17**

Safety & FCC Statement **19**

- Safety Statement 19
- FCC Warning Statement 19

About This User Guide **21**

Installation And Connection **22**

- Power Connection 25
- Dual Power Supply 26
  - SNMP Alerts for Power-related Events 27
  - SNMP Alert Configuration 27
- Device Status LEDs 27
- Connecting to the Network 30
- Serial Connection 31
- Cellular Connectivity 32
  - Installing A New SIM Card 32
- Device Reboot 33

Initial Settings **34**

- Default Settings 35
  - Using the Web GUI 36
- Management Console Connection via CLI 37
  - Accessing the Web GUI CLI Terminal 38

Change the Root Password 38
Disable a Root User 40
Change Network Settings 41

MONITOR Menu 46

System Log 47
LLDP CDP Neighbors 47
Triggered Playbooks 48

Advanced Options 49

Opengear CLI Guide 50

Getting Started with ogcli 50
Basic Syntax 52
Common Configuration Examples 58
compare current configuration with a Proposed Configuration 59
Using the diff Tool 59

Config Shell Guide 69

Start and End a Config Shell Session 69
Navigate in the Config Shell 70
Fields, Entities and Contexts 71
Global Context Commands 73
Entity Context Commands 73
Apply or Discard Field Changes 74
Operations 76
Supported Entities 76
Example CLI Commands 78
Configuring a Port 81

Advanced Portmanager PM Shell Guide 83

Running pmshell 83

pmshell Commands 84

Custom Control Codes for Serial Ports 86

Configure Custom Control Codes 86

Configure Control Codes for a Specified Port (CLI Examples) 87

Configure a Control Code Value for All Ports 88

Docker 89

Cron 89

Options: 89

Initial Provisioning via USB Key 91

EULA and GPL 92

ACCESS Menu 93

Local Terminal 93

Access Serial Ports 94

Quick Search 95

Access Using Web Terminal or SSH 95

Serial Port Logging 96

CONFIGURE Menu 98

Configure Serial Ports 98

Edit Serial Ports 99

Configure Single Sessions for Ports 102

Single Session Enabled 102

In the Web UI 102

In Config Shell 104

Single Session Behavior 106

Configure Raw TCP Access for Serial Ports 108
Service Implementation 109
WebUI Configuration 110
Config CLI Configuration 111
ogcli Configuration 113
Autodiscovery 113
Local Management Consoles 117
Lighthouse Enrollment 118
Manual Enrollment Using UI 120
Manual Enrollment Using the CLI 121
Playbooks 121
Create Or Edit a Playbook 122
PDUs 127
Add and Configure a PDU 128
PDU Operation 131
System Alerts 132
System Alerts - General 133
Authentication 133
Configuration 133
System Alerts - Power 134
Enable Power Supply Syslog Alerts 134
System Alerts - Networking (Connection Status) 136
Configure Signal Strength Alerts 136
Network Connections 137
Network Interfaces 137
DNS Configuration 138
Configure DNS via the Web UI 139
Configure DNS via the Command Line 140
Loopback Interface 141

Loopback Characteristics 142
Dual SIM 143
Display SIM Status and Signal Strength 144
Installing A New SIM Card 146
Select The Active SIM (Manual Failover Mode) 147
Select The Primary SIM (Automatic Failover Mode) 148
Dual SIM Failover 150
Failover Modes 153
Activate or Configure Failover 154
Cellular Interface Policy Settings 156
Cellular Modem Firmware Upgrade 157
Modem Firmware Upgrade Procedures 158
Cellular Availability During Upgrade 158
cell-fw-update Help 159
Update Local File List and Download Latest Firmware Files 160
List Supported Carriers 161
Automatic Firmware Update for Current Carrier 162
Firmware Update For Specific Carrier 162
Manual Firmware Update 162
Modem Update Troubleshooting Guide 164
Determine if Modem is Ready & Available 165
Determine if the Modem is Currently Being Upgraded 165
Network Aggregates - Bonds and Bridges 166
Bridges 166
Bonds 170
Spanning Tree Protocol 175
Enable STP in a Bridge 177
Bridge With STP Enabled - UI 177
Bridge With STP Enabled - OGCLI 178
Bridge With STP Disabled - OGCLI 178

IPsec Tunnels 179
Create, Add or Edit IPsec Tunnels 179
Static Routes 183
Configure Static Routes 185
Managing Static Routes via Command Line 187
Network Resilience 188
Out Of Band Failover 189
Enable Out-of-Band Failover 189
DNS Queries on a Dormant Failover Interface 191
OOB Failover Types & Failover Behavior 192
User Management 194
Groups 194
Permission Changes in the Web UI 195
Understanding Access Rights 195
Understanding Serial Port Access 200
Create a New Group 203
Edit an Existing Group 205
Local Users 206
Create a New User With Password 207
Create a New User With No Password (Remote Authentication) 209
Modify An Existing User Account With Password 209
Manage SSH Authorized Keys for a User Account 210
Delete a User's Account 211
Remote Authentication 211
Configure RADIUS Authentication 212
Configure TACACS+ Authentication 214
Configure LDAP Authentication 215
Local Password Policy 217
Set Password Complexity Requirements 218
Set Password Expiration Interval 220

Password Policy Implementation Rules 221
Services 222

FIPS Compliance 224

Configure FIPS 224
Enable FIPS 224
Disable FIPS 225
Verify that FIPS is enabled 225
Considerations for using the FIPS Feature 227
Brute Force Protection 231
Configure Brute Force Protection 231
Viewing Current Bans 232
Managing Brute Force Protection via Command Line 233
HTTPS Certificate 235
Network Discovery Protocols 236
File Server 238
Enable TFTP Service 239
Update The TFTP Service Storage Location 240
Routing 241
Dynamic Routing 241
Static Routing (via the ogcli) 242
OSPF Configuration 243
Managed Configuration Items 244
NEW FIELDS in REST API & CONFIG SHELL 245
REST API 245
Config Shell 246
Interfaces, Neighbors and Networks. 247
Interfaces CONTEXT 247
Neighbors CONTEXT 249
Networks CONTEXT 249

Interaction With Configuration Files 250
Confirm OSPF Neighbours 251
Wireguard Configuration 252
Viewing a WireGuard Configuration 252
Configure WireGuard through Config Shell or REST API 253
Config Shell WireGuard CONFIGURATION 254
REST API WireGuard CONFIGURATION 255
Configurable WireGuard FieLds 256
WireGuard Context Sub-objects 257
Addresses 257
Peers 258
Hooks 259
Adding a WireGuard Interface to a Firewall Zone 260
SSH 260
Unauthenticated SSH to Serial Ports 261
Enable Unauthenticated SSH 262
Enable SSH 263
Enable/Disable 263
Connecting Directly to Serial Ports 264
Feature Persist 265
Properties and Settings 265
Syslog 268
Add a New Syslog Server 268
Global Serial Port Settings 269
Global Serial Port Settings Tab - Field Definitions 269
Syslog Facility Definitions 271
Syslog Severity Definitions 272
Edit or Delete an Existing Syslog Server 273
Session Settings 273
Firewall 275

Firewall Guide 276

Introduction 276
Example WebUI Configuration 277
Custom Rules (firewalld “rich-rules”) 280
Useful Templates for use in webUI or CLI 282
Sample Rich Rules Templates 283
Firewall Management 284
Firewall Zone Settings 286
Port Forwarding 287
Manage Custom Rules 287
Firewall - Source Address Filtering 288
Firewall Source Address Bulk Services 290
Firewall Egress Filtering 291
Interzone Policies 298
Create an Interzone Policy 299
Edit or Delete an Interzone Policy 301
Customized Zone Rules 301
Adding WireGuard Zones to a Firewall 302
System 302
Administration 303

Date & Time 305

Manual Date & Time Set 306
NTP Configuration & Authentication 306
CLI Commands Associated with NTP Configuration 308
Time Zone 310
Manual Settings 310
Automatic Settings 311
Factory Reset 312
Reboot 315

Perform a simple reboot from the UI 315
Export Configuration 316
Export Configuration via Web UI 317
Export Configuration via ogcli 317
Control The Export Of Sensitive Data 318
Lighthouse Node Backup 318
Restore Configuration 319
Restore Configuration Via Web UI 319
Import Configuration via ogcli 321
Automated Rollback To Working Configuration 322
System Upgrade 324
Upgrade Via Fetch From Server 326
Upgrade Via Upload 326
SNMP 326
SNMP Service 327
SNMP Alert Managers 328
Multiple SNMP Alert Managers 329
Create or Delete an SNMP Manager 329
New SNMP Alert Manager Page Definitions 330
Advanced Options 332
Config CLI GUIDE 333
Navigation in Config CLI 334
Starting a Session in Config CLI 334
Exiting a Config CLI Session 334
Navigating the Config CLI 334
Understanding Fields, Entities and Contexts 336
Global & Entity-Context Commands 339
Global Context Commands 339

Entity Context Commands 340

Config CLI Entities 341

Supported Entities 341

Config CLI Commands 349

add 350
apply 351
Apply all changes 352
Apply changes to specific sections of configuration 353
changes 354
delete 355
diff 356
discard 358
Discard groups of changes 360
Discarding specific changes 361
edit 361
exit 362
help (or ?) 363
Help command used standalone 364
Help used in conjunction with a command 364
Help used with a configuration option 365
import/export 365
show 368
up / exit / .. 373

How Changes Are Applied or Discarded 375

Applying or Discarding Changes 376

Multi-Field Updates 378
Error Messages 382
String Values In Config Commands 383
Config CLI Use Case Examples 385
Adding a User 385
Configuring a Port 387
Configure a Single Session on a Port 389
Create or Configure a Loopback Interface 391
Create Source NAT Rules 393
REST API 394
Logging and Debugging 395
Configure NET1 Static IPV4 395
Configure NET2 Static IPV4 396
Configure NET3 Static IPV4 for OM2224-24e units 396
Configure WireGuard through Config Shell 396
Root User Password - cleartext 398
Root User Password = passWOrd via SHA256 398
Define Password Complexity Rules 398
Hostname 398
Contact Info 399
Time Zone and NTP 400
Create Admin User 400
Create Breakglass User (belongs to netgrp) 401
Enable netgrp - Set to ConsoleUser 401
Change SSH Delimiiter to : default is + 402
Change Port Labels 402
Enable Tacacs - Set Mode to remotelocal 403
Enable lldp on Net1 & Net2 403
Enable tftp 403

Enable Boot Messages 404
Define Session Timeouts 404
Define MOTD 404
Enable SIMM 1 Enable and Add APN 404
Enable SIMM 1 Complete End Points 405
Enable Failover 406
Add a Syslog Server 406
Set Port Logging Remote Syslog Settings 407
Enable System Monitor SNMP Traps 408
Enable SNMP V2 Service for Polling 409
Enable 2 SNMP Traps and Trap Servers 409
Create a StaTic Route 410
Edit LAN (Net2) Firewall Zone 410
Edit WAN (Net1) Firewall Zone 411
Custom_rule Example for Port and Protocol 412
Enroll Into Lighthouse 412

Opengear CLI Guide 413

Getting Started with ogcli 413
Basic Syntax 415
Common Configuration Examples 421
compare current configuration with a Proposed Configuration 422
Using the diff Tool 422

Config Shell Guide 432

Start and End a Config Shell Session 432
Navigate in the Config Shell 433
Fields, Entities and Contexts 434
Global Context Commands 436
Entity Context Commands 436
Apply or Discard Field Changes 437

Operations 439
Supported Entities 439
Example CLI Commands 441
Configuring a Port 444

Advanced Portmanager PM Shell Guide 446

Running pmshell 446
pmshell Commands 447

Custom Control Codes for Serial Ports 449

Configure Custom Control Codes 449
Configure Control Codes for a Specified Port (CLI Examples) 450
Configure a Control Code Value for All Ports 451

Docker 452
Cron 452
Options: 452
Initial Provisioning via USB Key 454
EULA and GPL 455

UI Button Definitions 456

COPYRIGHT ©

Opengear Inc. 2024. 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

SW Release	Revision Date	Description
22.11.0	November 2022	Updates to: Group permissions - enhancements NTP Configuration added Serial Port logging data counters Serial Port autodiscovery System Alerts - UI layout changes
23.03.0	March 2023	Updates to: Added CM8100-10G SKU items to Guide 10G Updates to OOB Failover OOB Failover - additional probe address added Added Firewall - Source Address Filtering
23.10.0	November 2023	Dual DC power supply option Support for OSPF and WireGuard New user password limitations (cannot use 'default') Firewall custom rules updated Firewall source address updated Remote syslog references removed Configure Single Session on a Port
23.10.4	February 2024	FIPS Compliance information added Config CLI User Guide added Config Diff tool added in CLI Guide
24.03	March 2024	Cellular Firmware Upgrade (in-field, CLI) Loopback Interface Firewall Egress Filtering

24.07.0	July 2024	Support for Raw TCP Access for Serial Ports
24.11.1	Nov 2024	Config Diff tool updated in Ogcli Guide & Config CLI Guide Config Rollback (automated) feature added Factory Reset (Erase) procedure updated

SAFETY & FCC STATEMENT

SAFETY STATEMENT

Please take care to follow the safety precautions below when installing and operating the Console Manager:

- Do not remove the metal covers. There are no operator serviceable components inside. Opening or removing the cover may expose you to dangerous voltage which may cause fire or electric shock. Refer all service to Opengear qualified personnel.
- To avoid electric shock the power cord protective grounding conductor must be connected through to ground.
- Always pull on the plug, not the cable, when disconnecting the power cord from the socket.

Do not connect or disconnect the appliance during an electrical storm. Also use a surge suppressor or UPS to protect the equipment from transients.

FCC WARNING STATEMENT

This device complies with Part 15 of the FCC rules. Operation of this device is subject to the following conditions: (1) This device may not cause harmful interference, and (2) this device must accept any interference that may cause undesired operation.

Proper back-up systems and necessary safety devices should be utilized to protect against injury, death or property damage due to system failure. Such protection is the responsibility of the user.

This device is not approved for use as a life-support or medical system.

Any changes or modifications made to this device without the explicit approval or consent of Opengear will void Opengear of any liability or responsibility of injury or loss caused by any malfunction.

This equipment is for indoor use and all the communication wiring are limited to inside of the building.

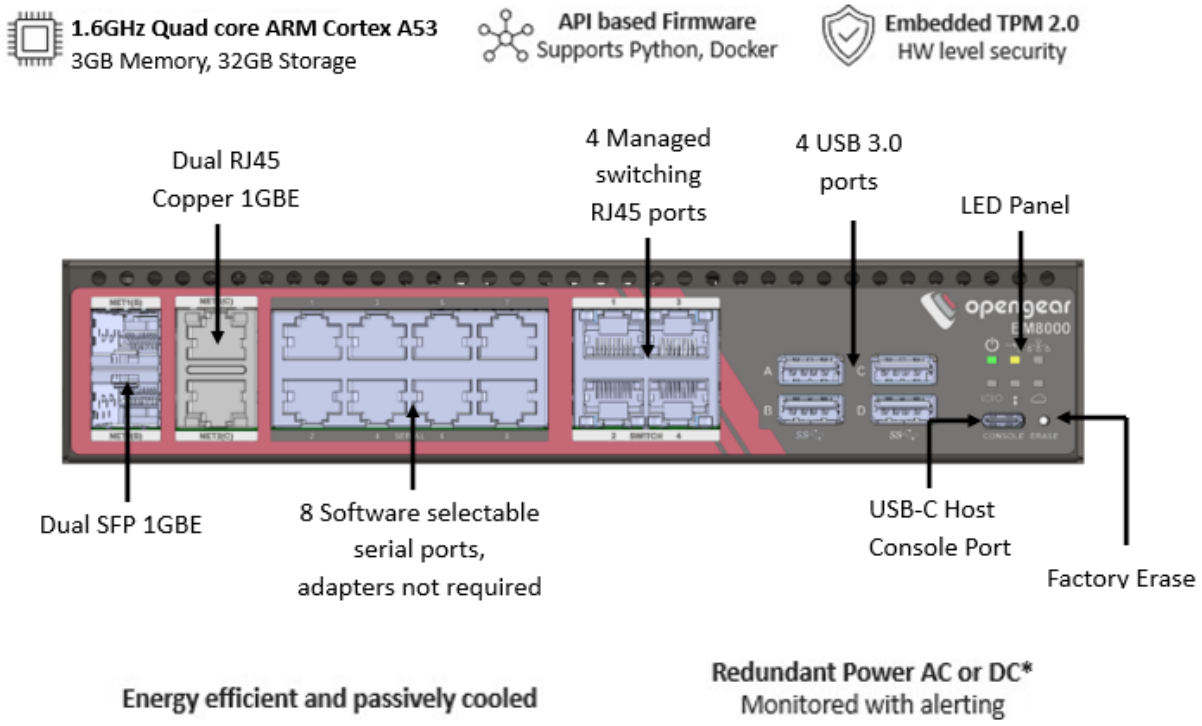
ABOUT THIS USER GUIDE

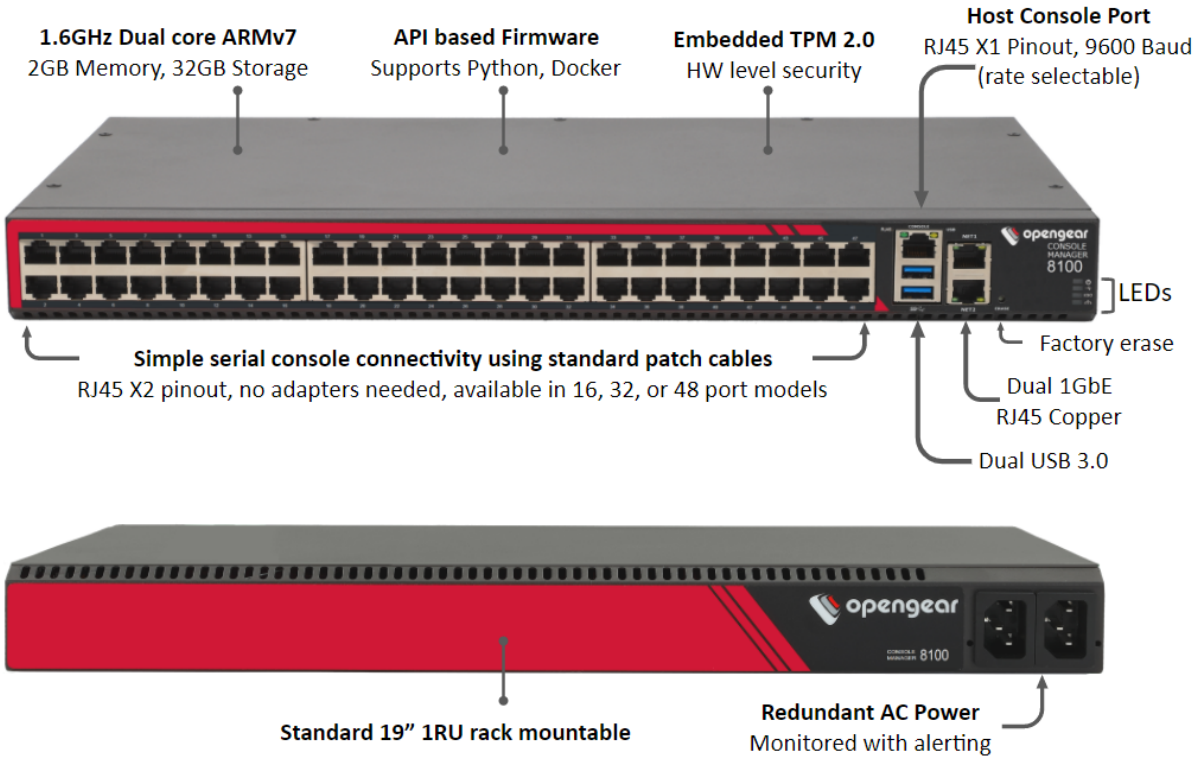
This user guide is up to date for the 24.11.1 firmware release. When using a minor release there may or may not be a specific version of the user guide for that release.

INSTALLATION AND CONNECTION

This section describes how to install the appliance hardware and connect it to controlled devices.

Console Manager Features:





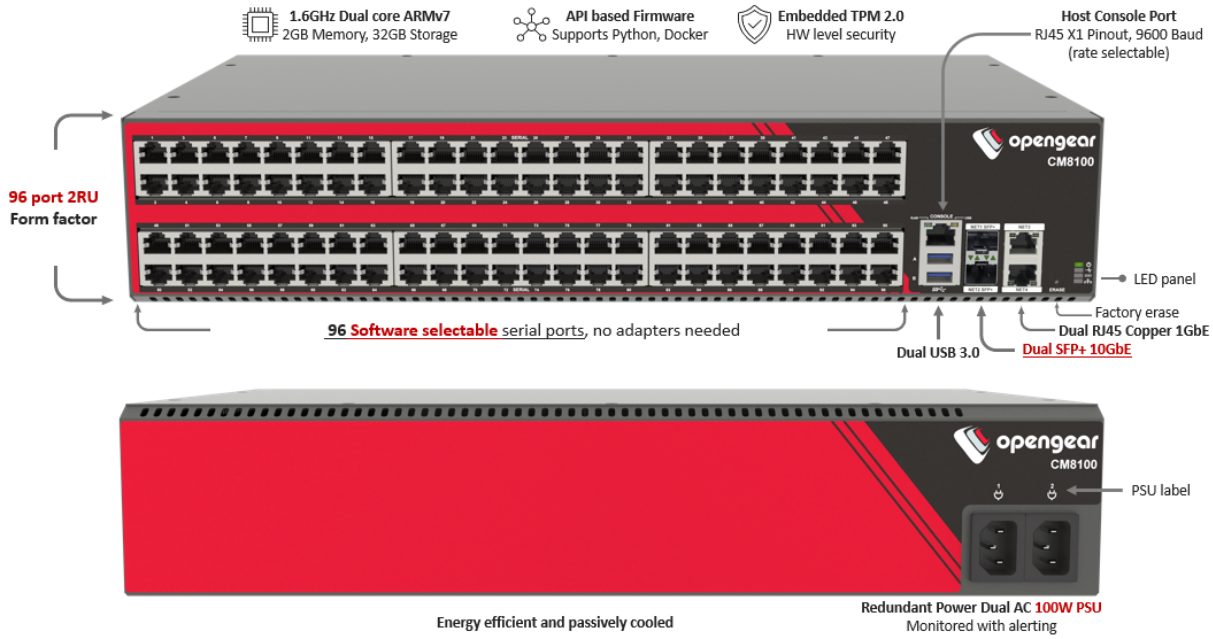
CM8100-10G Features:

The following features apply to the CM8100-10G model:

- Static IP on Net 3.
- Pin out switching by software selectable pinout.
- Two additional 10G SFP+ fiber interfaces.

CM8148-10G

CM8196-10G


1.6GHz Dual core ARMv7
2GB Memory, 32GB Storage
API based Firmware
Supports Python, Docker
Embedded TPM 2.0
HW level security
Host Console Port
RJ45 X1 Pinout, 9600 Baud
(rate selectable)
96 port 2RU
Form factor
opengear
CM8100
LED panel
Factory erase
Dual RJ45 Copper 1GbE
96 Software selectable serial ports, no adapters needed
Dual USB 3.0
Dual SFP+ 10GbE
opengear
CM8100
PSU label
Energy efficient and passively cooled
Redundant Power Dual AC 100W PSU
Monitored with alerting

POWER CONNECTION

The CM8100 models have dual power inlets with auto failover built in. These power supplies accept AC input voltage between 100 and 240 VAC with a frequency of 50 or 60 Hz. See the following tables for typical power draw.

Two IEC AC power sockets, which use conventional IEC AC power cord, are located on the power side of the metal case.

Note: Dual DC Power Supply. DDC models have a dual DC power supply with screw-in DC terminals (supplied).

Note: Country specific IEC power cords are included with the CM8100.

See also "Dual Power Supply" on the next page and "System Alerts - Power" on page 134.

Console Manager Platform (CM8100) Environmental And Power	
Power Supply	Dual AC
Power Draw CM8100	Typical ly <15W
Power Draw CM8100-10G	CM8148-10G <25W Typical CM8196-10G <30W Typical
Operating con-ditions	Temperature 5~50C, Rel Humidity 5~90%

Cooling	Passive
Power Draw Sensors	Active multi-zone power draw monitoring of 12V power. No monitoring on 120V AC.

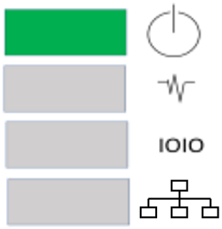
DUAL POWER SUPPLY

Dual Power Supply, including Dual DC (DDC) can provide power redundancy for devices, especially those that may operate in harsher environments. A secondary power supply provides redundancy for the device if one PSU is unplugged or in the event of a failure.

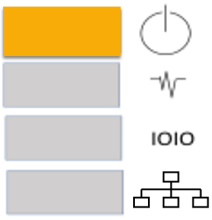
LED POWER STATUS INDICATOR

The power LED indicator requires no configuration and will display the dual power status on the Console Manager device.

On a **dual** PSU device that has power connected to *two* PSUs, the LED power status indicator should be green at all times.



If a **dual** PSU device has power connected to *one* PSU (power supply unit), the LED power status indicator is colored amber indicating that the unit has no redundancy in the event of a power failure.



SNMP ALERTS FOR POWER-RELATED EVENTS

The System Voltage Range SNMP alert is triggered when there is a change in power status such as a system reboot or when the voltage on either power supply leaves or enters the configured range of the System Voltage alert.

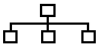
SNMP ALERT CONFIGURATION

The System Voltage Range SNMP alert is configured in the Configure > SNMP Alerts page, see "System Alerts - Power" on page 134.

DEVICE STATUS LEDS

The LED states shown below are determined through infod status and config-server data. The config server holds a configurable threshold value for the Cell LED Amber / Green light, and modem enabled / disabled information.

Status LEDs
LED Condition

	LED Off	Amber Flashing	Amber Solid	Green Flashing	Green Solid
Power	Device is off.		On a dual power supply system: Only one PSU is connected.		On a single power supply system: power is connected. On a dual power supply system: Redundant power is connected.
Heartbeat	Device has halted.	Device is booting.		Normal operation.	Device is halted.
Network 	No active network connection	Device is failover starting.	Device is in failover.	Normal network connection is stopping or normal network is up and failover is stopping.	Network is connected.

Status LEDs (continued).

	LED Condition				
	LED Off	**Amber Flashing**	**Amber Solid**	**Green Flashing**	**Green Solid**

CM8100 ONLY					
NET1	No active network connection	Network activity	Network link (any speed)	N/A	Network link 1G.
NET2	No active network connection	Network activity	Network link (any speed)	N/A	Network link 1G.
CM8100-10G ONLY					
NET1	No active network connection	Network activity	Network link	Network activity 10G	Network link 10G.
NET2	No active network connection	Network activity	Network link	Network activity 10G	Network link 10G.
NET3	No active network connection	Network activity	Network link (any speed)	N/A	Network link 1G.
NET4	No active network connection	Network activity	Network link (any speed)	N/A	Network link 1G.
IOIO				Any serial activity is received, on either console/usb console or device serial ports.	

Note: The amber LED signal threshold config is set to 50%.of normal signal strength.

For information on the setting of network and power alert thresholds, see:

"System Alerts - Networking (Connection Status)" on page 136

"System Alerts - Power" on page 134

CONNECTING TO THE NETWORK

Generally, Console Manager products have two network connections labeled NET1 and NET2. In the CM8100 there are options for copper wiring (on a standard RJ-45 connector). The CM8100-10G also has a static IP port on NET3.

Note: Installing an SFP module will trigger a communication between the module and the device. Usually, this will automatically limit the Ethernet interface to 1G. However, if a 10G interface is indicated, support the interface will be configured for 10G speeds.

The network connections on the CM8100 are located on the serial port side of the unit. Connect the provided shielded CAT5 cable to the NET1 to a computer or into your network for initial configuration. By default NET1 and NET2 are enabled.

The CM has two network connections labeled NET1 and NET2. There are options for copper wiring (on a standard RJ-45 connector) and fiber (through a standard SFP module).

The network connections on the CM are located on the serial port side of the unit. Connect the provided shielded CAT5 cable to the NET1 to a computer or into your network for initial configuration. By default NET1 and NET2 are enabled.

You can use either 10/100/1000BaseT over Cat5 or fiber-optical transceiver (1Gbps) in the SFP slot for NET1 or NET2.

SERIAL CONNECTION

Note:X1 and X2 are Opengear specific labels, where X2 = Cisco straight and X1 = Cisco reversed.

Local Console Port:
Serial Port 1 is the default local console port.

CM8100

Serial Ports:
The serial connections feature RS-232 with Cisco Straight X2 pinout, 50 to 230,400bps. Connect serial devices with the appropriate STP cables.
Note: The CM8100-10G also offers a software-selectable pin out (Port PinOut) on all serial ports.

Console Port:
1 x RJ45 RS-232 Console Port - Cisco rolled X1 pinout, baud rate 9600

Console Manager

Serial Ports:
RJ45 RS-232 Console Port (50 to 230,400 bps, software selectable Cisco-straight or Cisco-rolled pinout)

Local Console Ports:
1 x USB-C 2.0 Console Port and 1 x RJ45 Serial (Straight Pinout)

CELLULAR CONNECTIVITY

The cellular interface is certified for global deployments with most carriers and provides a CAT12 LTE interface supporting most frequencies in use. To activate the cellular interface, you should contact your local cellular carrier and activate a data plan associated to the SIM installed.

INSTALLING A NEW SIM CARD

When installing a new SIM card into its slot while the appliance is active (hot swapping), it is possible that it will take a minute or two for the system to react and stabilize after the SIM card change.

Two SIM card slots are located on the rear face of the device between the antennas, insert each SIM card(s) into its respective slot (marked 1 and 2) until you feel the card click into place.

Connect the RJ11 cable at the RJ11 port at the rear, you will hear or feel a slight click when it is correctly inserted. The modem is configured at the WebUI.

DEVICE REBOOT

CONFIGURE > System > Reboot

When the Console Manager reboots, the cellular IP address will not be preserved.

To reboot the unit*:*

Select **CONFIGURE > System > Reboot**.

To conduct a full erase and factory reset see "Factory Reset" on page 312

> **Note:** Factory reset will reset the appliance to its factory default settings. Any modified configuration information is erased.

INITIAL SETTINGS

This section provides step-by-step instructions for the initial settings on your Console Manager.

By default, all interfaces are enabled. The unit can be managed via Web GUI or by command line interface (CLI).

- "Default Settings" on the next page
- "Management Console Connection via CLI" on page 37
- "Change the Root Password" on page 38
- "Disable a Root User" on page 40
- "Change Network Settings" on page 41
- For Configure Serial Ports (see "Configure Serial Ports" on page 98)

Tip: There is also a Quick Start Guide to assist with easy setup of the Console Manager. The QSG is available at: https://opengear.com/support/documentation/

DEFAULT SETTINGS

Tip: See also the Quick Start Guide available at the Opengear documentation web page: https://opengear.com/support/documentation/

The CM8100 is configured with a default static IP Address for NET1 of 192.168.0.1 Subnet Mask 255.255.255.0.

The CM8100-10G devices are configured with a default static IP Address for NET3 of 192.168.0.1 Subnet Mask 255.255.255.0.

The Console Manager comes configured with a default static IP Address for NET1 of 192.168.0.1 Subnet Mask 255.255.255.0.

SERIAL PORT SETTINGS

The default settings for the serial ports (1 up to 48) on a new device are:

The default settings for the serial ports (4 up to 8) on a new device are:

“Console server” mode, 9600, 8N1, X2 (Cisco straight) pinout; the escape character is “~” .

BROWSER WEB GUI

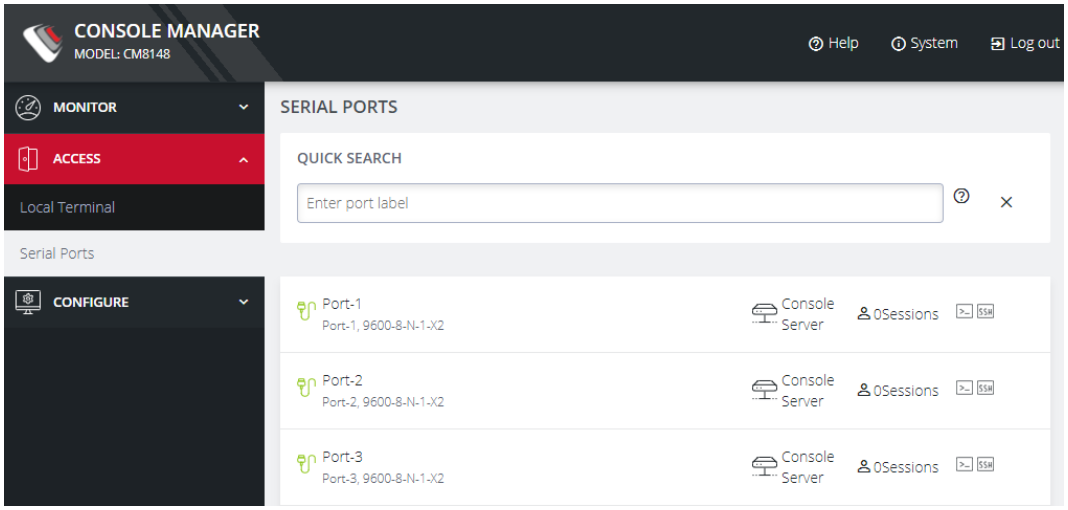
The Console Manager offers a Web GUI via web browser that supports HTML5.

1. Type https://192.168.0.1 in the address bar. HTTPS is enabled by default.
2. Enter the default username and password

   Username: root

   Password: default

3. After the first successful log-in you are required to change the root password.
4. After log-in the Web GUI is available. Check system details in the top right-hand side of the Web GUI.
5. In the Navigation Bar on the left side, navigate to the **ACCESS > Serial Ports** page. The Serial Ports page displays a list of all the serial devices, including the links to a Web Terminal or SSH connection for each.



USING THE WEB GUI

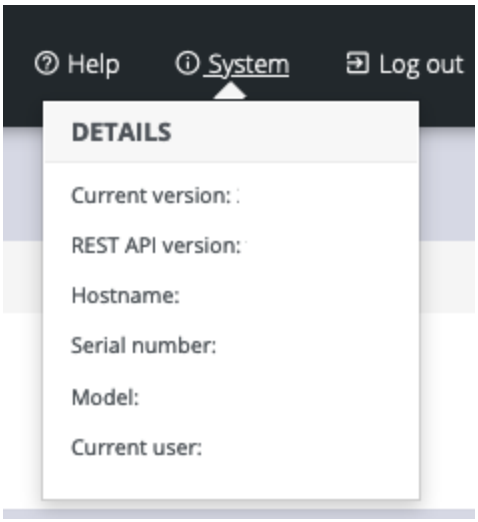
The Web GUI can switched between **Light** or **Dark** mode by adjusting the toggle on the bottom left.

Light mode changes the user interface to display mostly light colors. This is the default UI setting. Dark mode changes the user interface to display mostly dark colors, reducing the light emitted by device screens.

The Web GUI has three menu options on the upper right: **Help**, **System**, and **Log out**.

The **Help** menu contains a link to generate a **Technical Support Report** that can be used by Opengear Support for troubleshooting. It also contains a link to the latest User Guide.

The System menu presents the **Current version**, **REST API version**, **Hostname, Serial Number, Model**, and **Current user**.



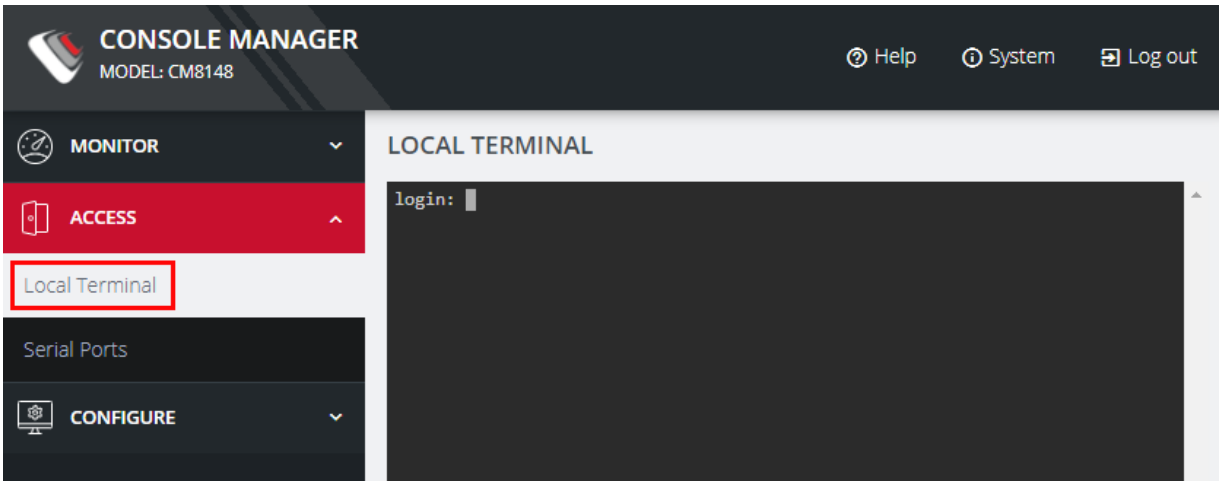
MANAGEMENT CONSOLE CONNECTION VIA CLI

The Command Line Interface (CLI) is accessible using your preferred application to establish an SSH session. Open a CLI terminal on your desktop, then:

1. Input the default IP Address of 192.168.0.1. SSH port 22 is enabled by default.
2. When prompted, enter the log in and password in the CLI.
3. After a successful log in, you'll see a command prompt.

ACCESSING THE WEB GUI CLI TERMINAL

An alternative CLI terminal is provided within the Web GUI. To access this terminal, in the left-hand side **Navigation Bar**, navigate to the **ACCESS > Local Terminal** page. You will be required to submit your log-in credentials.



CHANGE THE ROOT PASSWORD

CONFIGURE > User Management > Local Users > Edit User

For security reasons, only the root user can initially log in to the appliance. Upon initial login the default password must be changed.

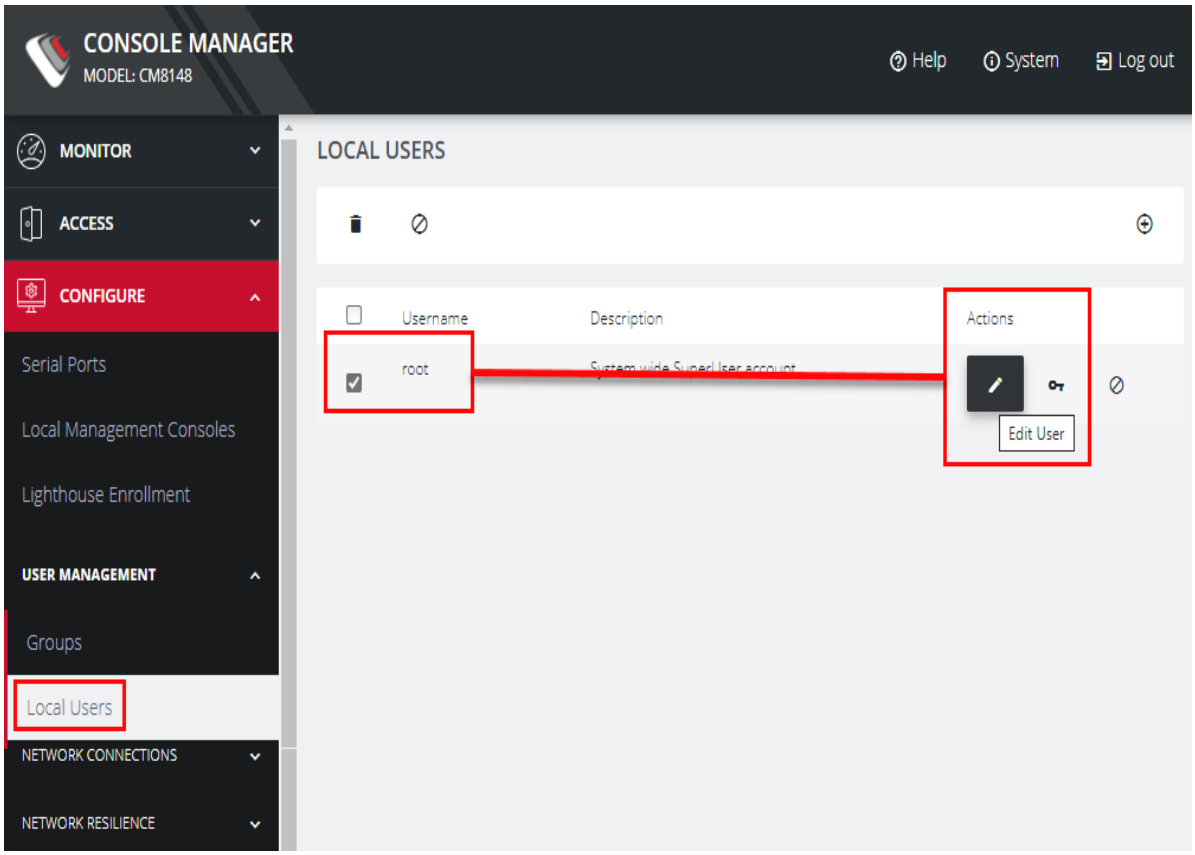
Note: Users are prevented from reusing the word “default” as their password. The factory default password automatically expires after a factory reset and users must choose a new password. This policy is applied through the WebUI, Config Shell and CLI.

Tip: Other Users' passwords may be changed using the same procedure by selecting the User's account name under the **Username** heading.

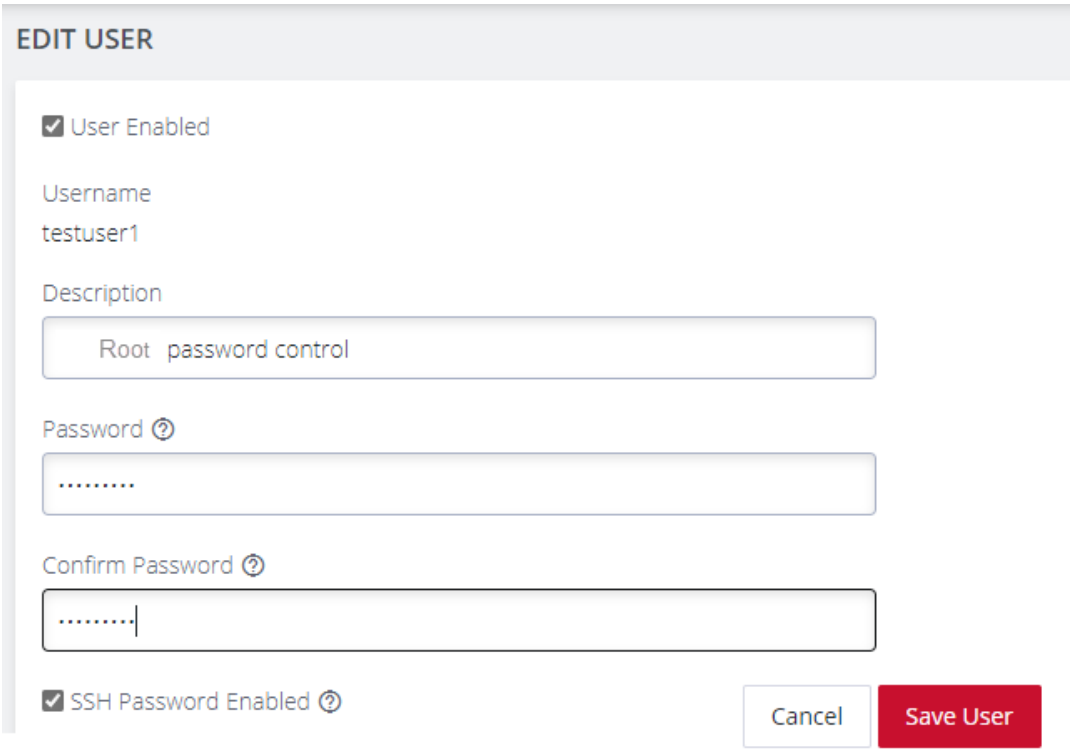
Note: The password must comply with your company's password complexity policy. See "Local Password Policy" on page 217

To change the password at any time:

1. Navigate to **CONFIGURE > User Management > Local Users**
2. Click the Root user's **Edit User** icon below the **Actions** heading.



3. In the **Edit User** page, if required, enter an optional description in the **Description** field. Enter a new password in the **Password** field and re-enter the password in the **Confirm Password** field.



4. Click **Save User**. A green banner confirms the password change has been saved.

DISABLE A ROOT USER

CONFIGURE > User management > Local Users

To disable a root user:

Note: Before proceeding, make sure that another user exists that has the Administrator role or is in a group with the Administrator role. For information on creating, editing, and deleting users, see "Local Users" on page 206

1. Navigate to **CONFIGURE > User management > Local Users**
2. Click the **Disable User** button in the **Actions** section next to the root user.
3. Click **Yes** in the **Confirmation** dialog.

To enable root user, log in with another user that has the Administrator role and click the **Enable User** button in the **Actions** section next to the root user.

.

CHANGE NETWORK SETTINGS

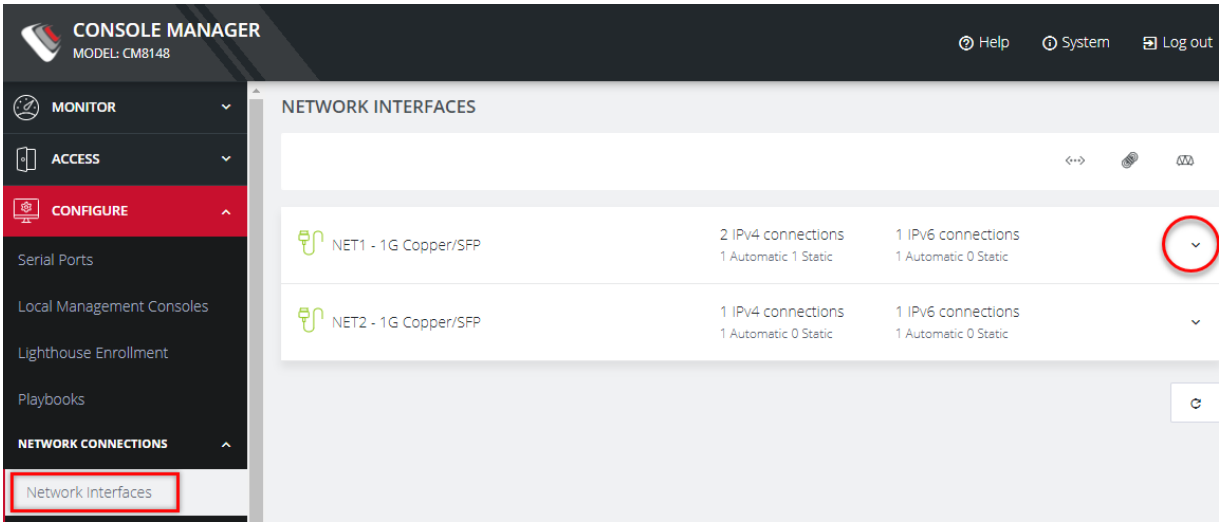
CONFIGURE > Network Connections > Network Interfaces

The interface supports both IPv4 and IPv6 networks. The IP address of the unit can be setup for Static or DHCP. The following settings can be configured for network ports:

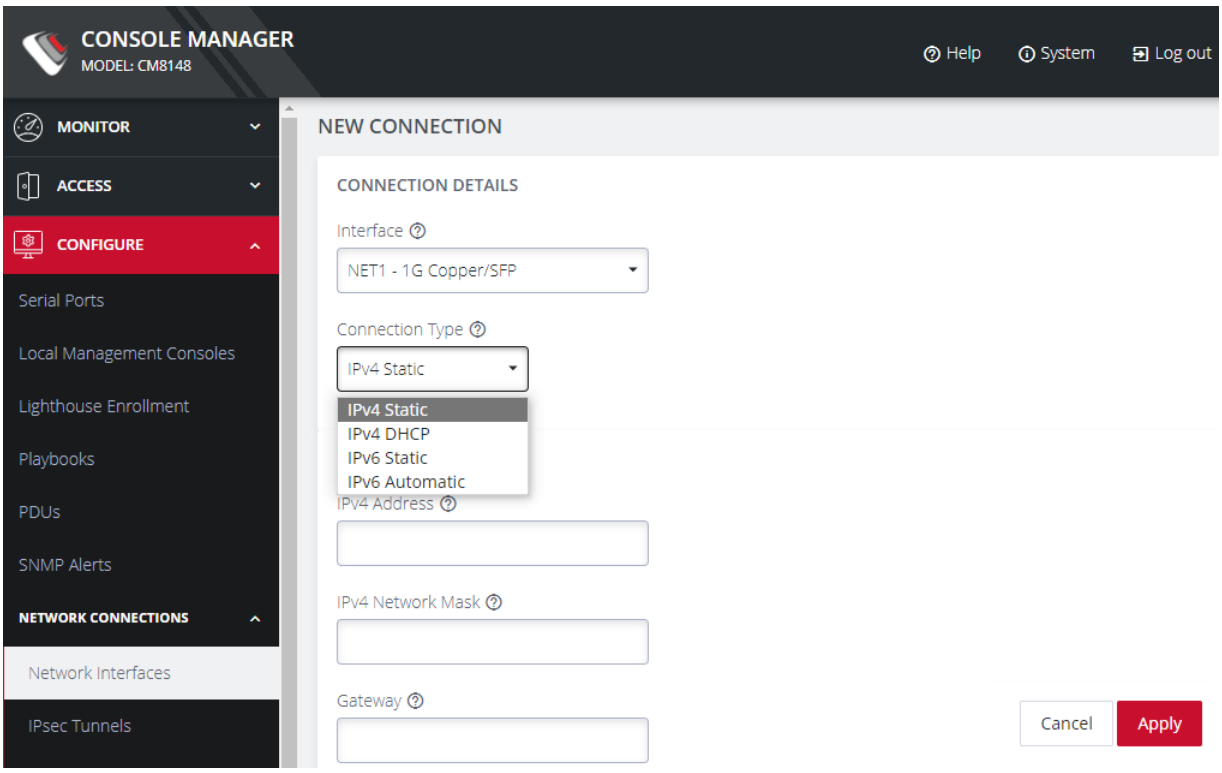
- IPv4, IPv6
- Static and/or DHCP
- Enabling or disabling network interfaces
- Ethernet Media types

To add a new connection:

1. Click **CONFIGURE > Network Connections > Network Interfaces**



2. Click the **expand arrow** to the right of the desired interface to view its details.
3. Click the **plus icon** to open the **New Connection** page.



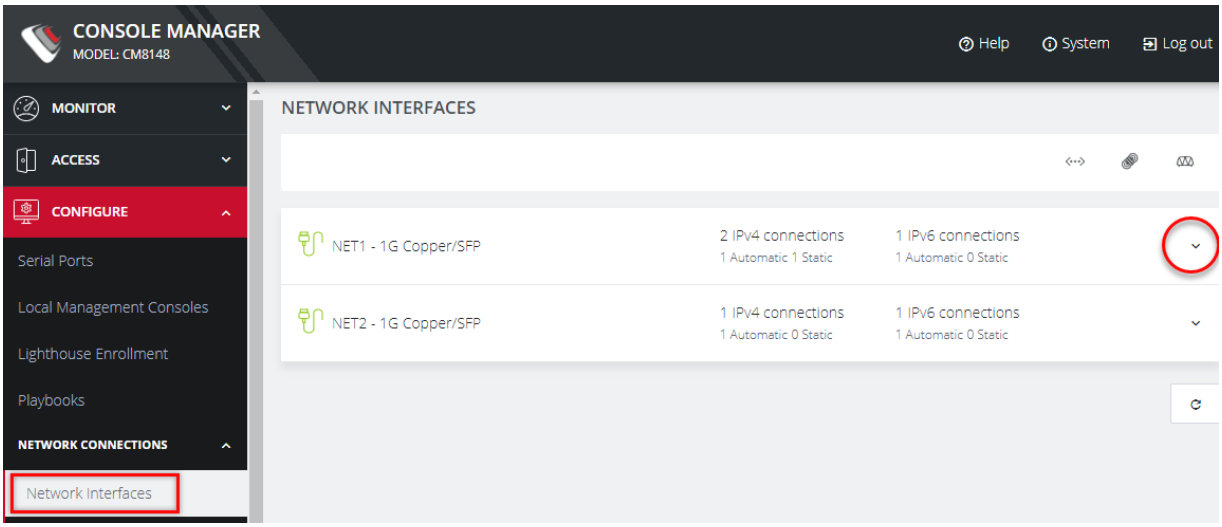
4. Select the **Interface** and **Connection Type** for your new connection.
5. The form on the bottom part of the page will change based on the **Connection Type** you choose. Enter the necessary information and click **Apply**.

   To disable or delete interfaces, use the controls on the expanded section on the **CONFIGURE > Network Connections > Network Interfaces** page.

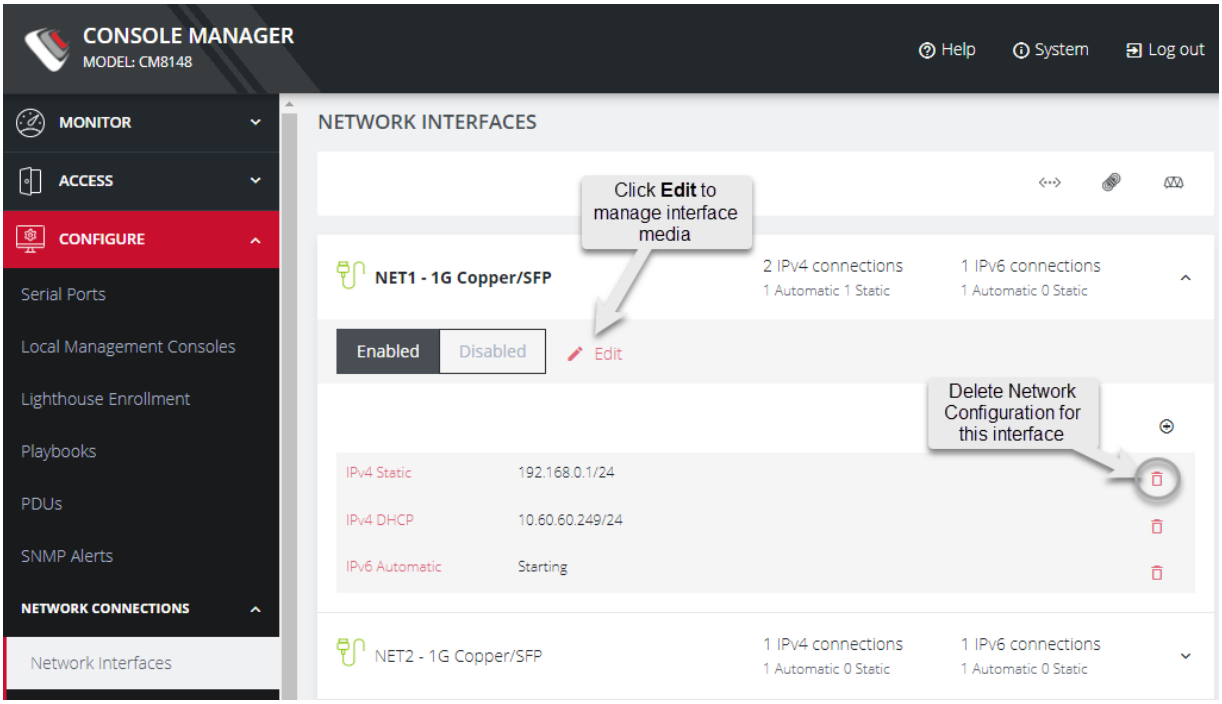
> **Note:**If you experience packet loss or poor network performance with the default auto-negotiation setting, try changing the Ethernet Media settings on the Console Manager and the device it is connected to. In most cases, select 100 megabits, full duplex. Make sure both sides are set identically.

To change the Ethernet Media Type:

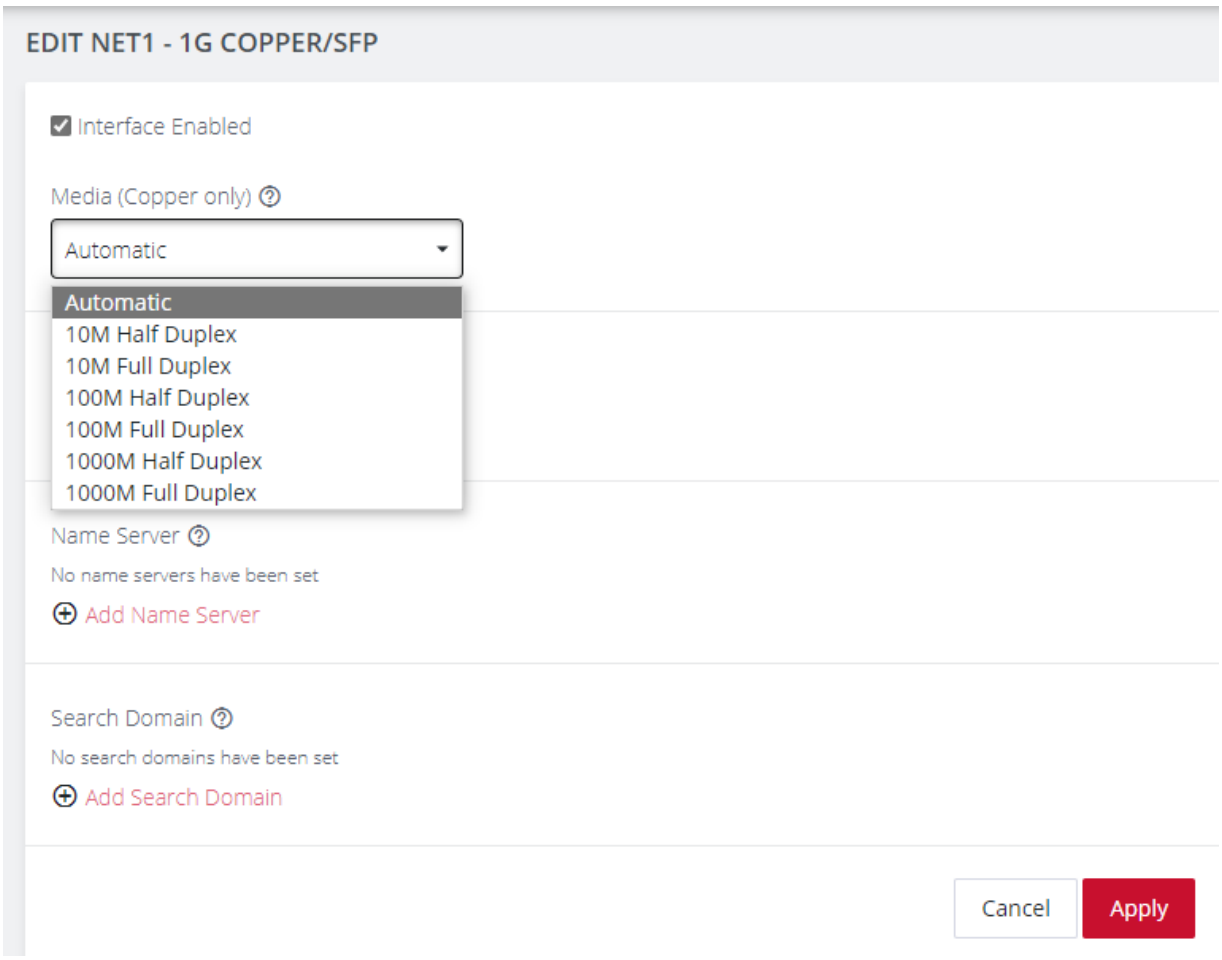
1. Click **CONFIGURE > Network Connections > Network Interfaces**



2. Click the expand arrow to the right of the interface you wish to modify.



3. Click **Enabled** .
4. To change the interface media setting, click the **Edit** button and edit the media settings as needed and click **Apply**.


EDIT NET1 - 1G COPPER/SFP
Interface Enabled
Media (Copper only)
Automatic
Automatic
10M Half Duplex
10M Full Duplex
100M Half Duplex
100M Full Duplex
1000M Half Duplex
1000M Full Duplex
Name Server
No name servers have been set
Add Name Server
Search Domain
No search domains have been set
Add Search Domain
Cancel
Apply

MONITOR MENU

The MONITOR Menu is a relatively short section comprising only three topics.

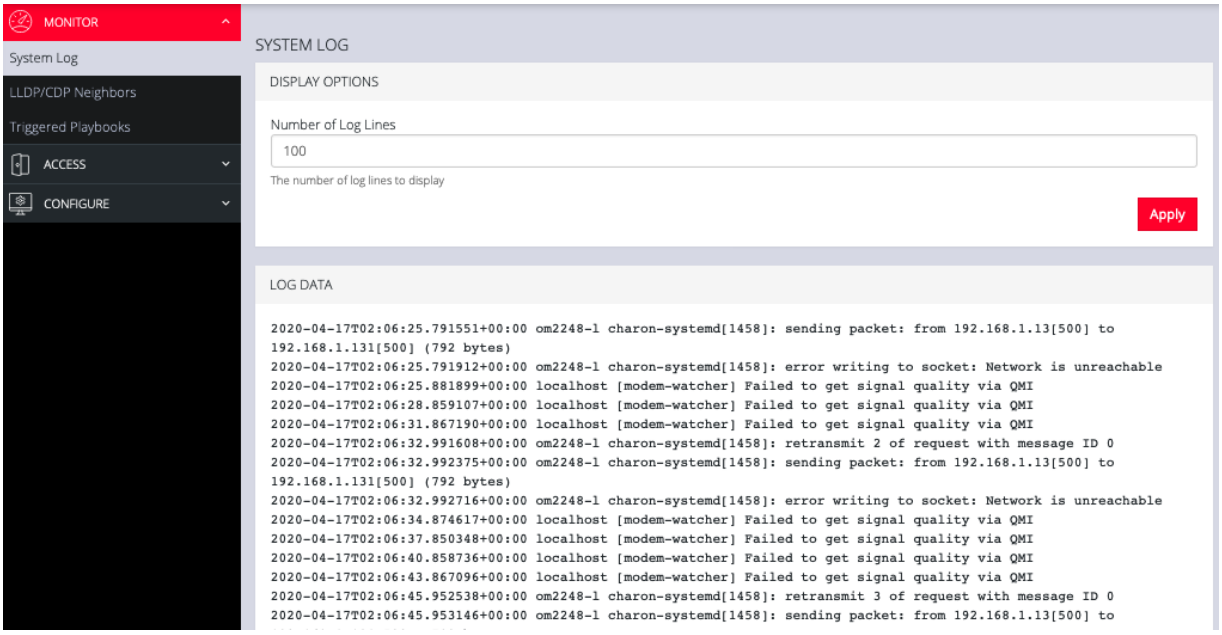
- System Log
  - Details of the system activity log, access and communications events with the server and with attached serial, network and power devices.
- LLDP/CDP Neighbors
  - Details of the LLDP/CDP Neighbors that are displayed when enabled for a connection.
- Triggered Playbooks
  - Monitoring current **Playbooks**, and applying filters to view any Playbooks that have been triggered.

SYSTEM LOG

MONITOR > System Log

The Console Manager maintains a log of system activity, access and communications events with the server and with attached serial, network and power devices.

To view the System Log, click **MONITOR > System Log**.



The System Log page lets you change the Number of Log Lines displayed on the screen. The newest items appear on the bottom of the list. Click the **Refresh** button on the bottom right to see the latest entries.

LLDP CDP NEIGHBORS

MONITOR > LLDP/CDP Neighbors

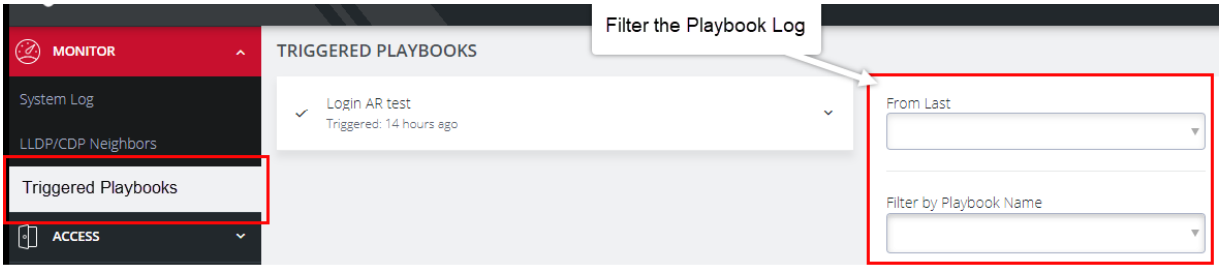
The Console Manager displays LLDP/CDP Neighbors when enabled for a connection. See **CONFIGURE > SERVICES > Network Discovery Protocols** to enable/disable.

TRIGGERED PLAYBOOKS

MONITOR > Triggered Playbooks

For information on creating **Playbooks**, see the Playbooks topic in this User Guide.

To monitor current **Playbooks**, click on **Monitor > Triggered Playbooks**. Choose the time period if desired, and filter by **Name** of **Playlist** to view any that have been triggered.



ADVANCED OPTIONS

The Console Manager supports a number of command line interface (CLI) options and REST API.

address : Primary Lighthouse address to enroll with

api_port : Optional port to use for the primary address when requesting enrollment

password : LH global or bundle enrollment password

bundle : Name of LH enrollment bundle

OPENGEAR CLI GUIDE

The **ogcli** command line tool is used for getting and setting configuration, and for retrieving device state and information. The purpose of ogcli is perform a single operation and exit. Operations are performed on a single entity, a list of entities, or all entities. Entities in ogcli are collections of related information items that represent device state, information or configuration.

For a list of operations supported by ogcli, see the "ogcli Operations" section.

> **Note:**ogcli is not an interactive shell, it runs a single command and exits.

GETTING STARTED WITH OGCLI

The best way to get started with ogcli is to use the help command. Refer to the table below to access help topics within ogcli.

For detailed information about ogcli and how it works, view the ogcli help topic by running this command:

```
ogcli help ogcli
```

ACCESS OGCLI HELP AND USAGE INFORMATION

Help Command	Displays...
ogcli help	Basic ogcli help and usage information.

Help Command	Displays...
ogcli help help	Detailed information about the help command.
ogcli help operations	The full list of operations and a brief description of each.
ogcli help entities	The full list of entities and a brief description of each.
ogcli help syntax	How to get information into and out of ogcli.
ogcli help ogcli	More detailed information about the ogcli tool.
ogcli help usage	Common ogcli usage examples.
ogcli help secrets	Detailed information about controlling the display of secrets in ogcli.
ogcli help <operation>	A description and example usage of a specific ogcli operation.
ogcli help <entity>	A description of a specific entity and the operations it supports.
ogcli help <entity> <operation>	An example of how to perform a specific operation on a specific entity.

BASIC SYNTAX

The ogcli tool is always called with an operation, with most operations also taking one or more arguments specifying an entity for the operation to act on.

```
ogcli <operation> [argument] [argument]
```

OGCLI OPERATIONS

Operation	Description
create	Create an item.
export	Export the system configuration.
diff	Show additions, removals, changes and functional differences between the input and running configurations. See also `config diff`.
get	Retrieve a list or single item.
help	Display ogcli help.
import	Import system configuration, merging with current system configuration.

Operation	Description
merge	Merge a provided list with existing config.
replace	Replace a list or single item.
restore	Import system configuration, replacing the current system configuration.
update	Update an item, supports partial edits.

SUPPLYING DATA TO OGCLI

For operations that modify an entity (e.g. 'update') the new information can be passed as inline positional arguments, but this quickly becomes cumbersome when setting a large number of fields. Information can instead be supplied through stdin by piping the contents of a file, or with Here Document (heredoc) style. The heredoc style is the most flexible format and is used extensively in ogcli examples.

HERE DOCUMENT

A here document (heredoc) is a form of input redirection that allows entering multiple lines of input to a command. The syntax of writing heredoc takes the following form:

```
ogcli [command] << 'DELIMITER'
 HEREDOC
DELIMITER
```

- The first line starts with the ogcli command, followed by the special redirection operator << and a delimiting identifier. Any word can be used as the delimiter, commonly 'EOF' or 'END'.
- The `HEREDOC` block can contain multiple lines of strings, variables, commands or any other type of input. Each line can specify one field to update.
- The last line ends with the delimiting identifier used above, indicating the end of input.

```
ogcli update user <username> << 'END'
 description="operator"
 enabled=false
END
```

INLINE ARGUMENTS

Field data can be entered inline with the ogcli command as arguments, with each field separated by a space.

```
ogcli update user <username> enabled=false description=\"operator\"
```

PIPES AND STANDARD INPUT

The data can also be entered via `stdin` by piping the data to the ogcli command.

```
echo 'enabled=true description="operator"' | ogcli update user
<username>
```

Alternatively, you can provide a file via input redirection with <.

```
echo 'enabled=true description="operator"' > partial_record
```

```
ogcli update user <username> < partial_record
```

QUOTING STRING VALUES

All string fields require the argument to be specified with double quotes ". The shell can consume double quotes, so care must be taken when specifying strings to ensure the quotes are passed to ogcli as input.

1. Double quotes in heredoc do not need to be escaped.

```
ogcli update physif <device-identifier> << 'END'
 description="test network"
END
```

2. Double quotes within single quotes do not need to be escaped.

```
ogcli update physif user <username> 'description="test user"'
```

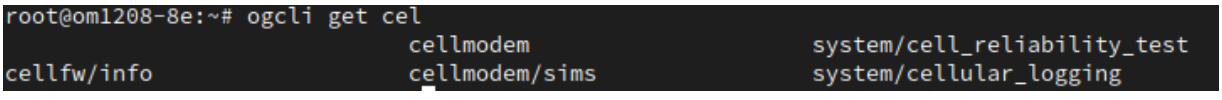
3. Double quotes not within single quotes need to be escaped.

```
ogcli update physif user <username> description=\"test user\"
```

TAB COMPLETION

ogcli includes tab completion to assist with typing commands. When entering the start of a command, press the **<tab>** key to complete the phrase to the nearest match.

If there are multiple matches, all options will be displayed for your reference.



DISPLAYING SECRETS IN OGCLI

Fields containing sensitive information are called **secrets**, which are handled specially by **ogcli** to obfuscate their values when they are displayed or exported.

Passwords and private keys are examples of secret fields.

The obfuscation process provides protection against "casual observation" only and offers no cryptographic security. The **obfusc** tool can be used to obtain the clear text version of any obfuscated secret generated by any Console Manager.

For more information, view the secrets help topic by running:

```
ogcli help secrets
```

The default behavior is for secrets to be passed to ogcli in clear text, and exported or displayed in obfuscated form.

For example, setting the password:

```
ogcli update services/snmpd auth_password=\"my secret\"
```

Retrieving the password (note, the output is abridged):

```
# ogcli get services/snmpd
auth_
password="TkcxJAAAABBSB3xoFWhPA6B7sDrzq3HwaTOAO/jsURqFa0qa7hc3TA=="
```

This behavior can be overridden to display sensitive fields in clear text, obfuscated form, or masked form using the **--secrets** option. The clear text and obfuscated forms are also accepted when supplying a sensitive field.

```
# ogcli --secrets=cleartext get snmpd
auth_password="my_secret"
```

```
# ogcli --secrets=obfuscate get snmpd
auth_password="my secret"
```

```
# ogcli --secrets=mask get snmpd
auth_password="********"
```

If an export is performed with the **--secrets=mask** option it is impossible to subsequently import the configuration, because the secrets have been removed.

COMMON CONFIGURATION EXAMPLES

These examples contain a variety of notations and usage patterns to help illustrate the flexibility of ogcli. The examples can be copied and pasted into the CLI.

REPLACE MESSAGE OF THE DAY (MOTD) DISPLAYED AT LOGIN

```
ogcli replace banner banner=\"updated message\"
```

RETRIEVE USER RECORD

```
ogcli get user <username>
```

UPDATE ITEM WITH FIELD WHERE VALUE IS A STRING

```
ogcli update user <username> description=\"operator\"
```

UPDATE ITEM WITH FIELD WHERE VALUE IS NOT A STRING

For example, a numeric or boolean value

```
ogcli update user <username> enabled=true
```

EXPORT SYSTEM CONFIGURATION

```
ogcli export <file_path>
```

IMPORT SYSTEM CONFIGURATION

```
ogcli import <file_path>
```

RESTORE SYSTEM CONFIGURATION

```
ogcli restore <file_path>
```

COMPARE CURRENT CONFIGURATION WITH A PROPOSED CONFIGURATION

The updated `ogcli diff` tool enables Opengear users to compare a proposed configuration with an existing configuration so that they may understand any prospective changes to the config.

The diff function performs a comparison of active configuration and an input configuration file, which must be in the format an export file produced by either a `config export <template-file>` or an `ogcli export <template-file>` operation. Any manual changes to this export file must include config or ogcli commands in a multi-line format using the 'END' heredoc marker as produced by an export. One line config or ogcli commands will not be accepted.

USING THE DIFF TOOL

The diff tool can be used by any user with administrator permission via the command line.

```
ogcli diff <input file>
```

or using config:

```
config diff <input file>
```

Note: `config diff` and `ogcli diff`, and can be used interchangeably using export files in either format.

If there are no differences between the active configuration and the input configuration file, the diff tool will not print any output, and the operation will have an exit code of 0.

```
root@om2248:~# ogcli export config_file
root@om2248:~# ogcli diff config_file
root@om2248:~# echo $?
0
```

The diff function will show any additions, removals and changes clearly in a streamlined format with only functional differences between the input and running configurations. Any additions that will be made to the active configuration are marked with a (+). For example, the `new_user` user does not exist in the active configuration, but is present in the input file supplied. If the input file was imported, this user would be added.

```
ogcli --secrets=obfuscate merge users <<'END'
+ users[1].enabled=true
+ users[1].groups[0]="admin"
+ users[1].no_password=false
+ users[1].ssh_password_enabled=true
+ users[1].username="new_user"
END
```

If the `new_user` user exists in the active configuration, but does not exist in the input file, this user will be removed if the input file was imported or restored. Removals will be marked with a (-) symbol.

```
ogcli --secrets=obfuscate merge users <<'END'
- users[1].enabled=true
- users[1].groups[0]="admin"
```

```
- users[1].no_password=false
- users[1].ssh_password_enabled=true
- users[1].username="new_user"
END
```

Changes in configuration between an item which exists in both the active configuration and the input file will also be displayed. The existing configuration will be marked with a (-) and the incoming change as a (+). In the example below, the `new_user` user belongs to the netgrp group on the device. However, if the input file is imported, it will belong only to the admin group.

```
ogcli --secrets=obfuscate merge users <<'END'
- users[1].groups[0]="netgrp"
- users[1].groups[1]="admin"
+ users[1].groups[0]="admin"
END
```

If any differences are found, the operation will have an exit code of 1. If there are any errors, the diff tool will have an exit code of 2.

Comparison to Default Values

If the input configuration file is missing properties or sections of configuration, the diff function will instead consider the differences between active configuration and the default values for those properties. Missing sections or properties from the input file will only be displayed in the diff tool output if the active configuration is different from the default system values. If any property or configuration section is missing from input configuration, and the running configuration is identical to the system defaults, it is omitted from the diff output.

How Secrets are Handled

The `--secrets` flag can be used to control how sensitive fields are displayed in the diff output. By default, sensitive fields are obfuscated. If the proposed config file was exported with `--secrets=cleartext` or `--secrets=mask` then the same value must be used when running `ogcli diff`.

```
root@om2248:~# ogcli --secrets=cleartext export config_file
root@om2248:~# ogcli --secrets=cleartext diff config_file
```

If the input file contains a different `--secrets` parameter than is passed to `ogcli diff` or `config diff`, an error will be returned:

```
root@om2248:~# config --secrets=cleartext export config_file
oot@om2248:~# config --secrets=cleartext diff config_file
root@om2248:~# config --secrets=mask diff config_file
The secrets flag provided doesn't match the flag in the proposed
config for physifs.
This error can be ignored with the --ignore-secrets-mismatch flag.
Type ogcli diff --help for more information.
```

This behaviour is the same for config:

```
root@om2248:~# config --secrets=cleartext export config_file
oot@om2248:~# config --secrets=cleartext diff config_file
root@om2248:~# config --secrets=mask diff config_file
The secrets flag provided doesn't match the flag in the proposed
config for physifs.
This error can be ignored with the --ignore-secrets-mismatch flag.
Type ogcli diff --help for more information.
```

The `--ignore-secrets-mismatch` flag can be used to ignore a difference in the `--secrets` parameter:

```
root@om2248:~# ogcli diff --ignore-secrets-mismatch config_file
```

Diff Tool Help

Basic help for `ogcli diff` can be accessed with `ogcli diff -h`. Similarly, help for `config diff` can be accessed with `config diff -h`. Detailed help for both diff tools can be accessed by `ogcli help diff`.

Limitations
JSON template files are no longer supported with config diff.
`ogcli diff` only supports input configurations generated by the same product SKU and software version.
Comments can be included between ogcli or config commands in the export file, but not within the commands or an error will be thrown. Comments must start with #. These will be ignored by the diff tool.

See also "diff" on page 356

ENABLE LOCAL CONSOLE BOOT MESSAGES

```
ogcli get managementports
```

```
ogcli update managementport mgmtPorts-1 kerneldebug=true
```

CREATE NEW USER

```
ogcli create user << 'END'
 description="superuser"
 enabled=true
 groups[0]="admin"
 password=”test123"
 username="superuser123"
END
```

CHANGE ROOT PASSWORD

```
ogcli update user root password=\"oursecret\"
```

CREATE NEW ADMINISTRATIVE USER

```
ogcli create user << 'END'
 username="adal"
 description="Ada Lovelace"
 enabled=true
 no_password=false
 groups[0]="groups-1"
 password="oursecret"
END
```

MANUALLY SET DATE AND TIME

```
ogcli update system/timezone timezone=\"America/New_York\"
```

```
ogcli update system/time time=\"15:30 Mar 27, 2020\"
```

ENABLE NTP SERVICE

```
ogcli update services/ntp << 'END'
 enabled=true
 servers[0].value="0.au.pool.ntp.org"
END
```

UPDATE SYSTEM HOSTNAME

```
ogcli update hostname hostname=\"system-hostname\"
```

ADJUST SESSION TIMEOUTS

```
ogcli update system/cli_session_timeout timeout=180
```

```
ogcli update system/webui_session_timeout timeout=180
```

SETUP REMOTE AUTHENTICATION WITH TACACS+

```
ogcli update auth << 'END'
 mode="tacacs"
 tacacsAuthenticationServers[0].hostname="192.168.250.21"
 tacacsMethod="pap"
 tacacsPassword="tackey"
END
```

SETUP REMOTE AUTHENTICATION WITH RADIUS

```
ogcli update auth << 'END'
 mode="radius"
 radiusAuthenticationServers[0].hostname="192.168.250.21"
 radiusAccountingServers[0].hostname="192.168.250.21"
 radiusPassword="radkey"
END
```

CREATE USER GROUP WITH LIMITED ACCESS TO SERIALPORTS

```
ogcli create group << 'END'
 description="Console Operators"
 groupname="operators"
 role="ConsoleUser"
 mode="scoped"
 ports[0]="ports-10"
 ports[1]="ports-11"
 ports[2]="ports-12"
END
```

VIEW AND CONFIGURE NETWORK CONNECTIONS

```
ogcli get conns
```

```
ogcli get conn system_net_conns-1
```

```
ogcli update conn system_net_conns-1 ipv4_static_
settings.address=\"192.168.0.3\"
```

```
ogcli create conn << 'END'
 description="2nd IPv4 Static Address Example"
 mode="static"
 ipv4_static_settings.address="192.168.33.33"
 ipv4_static_settings.netmask="255.255.255.0"
 ipv4_static_settings.gateway="192.168.33.254"
 physif="net1"
END
```

CONFIGURE SERIAL PORTS

```
ogcli get ports
```

```
ogcli get ports | grep label
```

```
ogcli get port ports-1
```

```
ogcli update port "port05" << 'END'
 mode="consoleServer"
 label="Router"
 pinout="X2"
 baudrate="9600"
 databits="8"
 parity="none"
 stopbits="1"
 escape_char="~"
 ip_alias[0].ipaddress="192.168.33.35/24"
```

```
 ip_alias[0].interface="net1"
 logging_level="eventsOnly"
END
```

CONFIG SHELL GUIDE

The Config Shell feature provides an interactive and familiar environment similar to older Opengear appliances. The result is an user-experience that feels like an Interactive CLI.

Advantages of the Config Shell are:

- Items can be created or updated without being applied immediately
- Items that are not applied are indicated by an asterisk (*) beside them when viewing information.
- Tab complete is supported for many commands.
- Built-in help (see "Global Context Commands" on page 73).
- Has a structured, tabular view when displaying lists of data.

START AND END A CONFIG SHELL SESSION

Start the config shell by typing `config` at a bash prompt. The bash prompt is presented to root and admin users when they log in via SSH or on the maintenance console.

You can exit the Config Shell by any of the following:

- Type `exit` to end the session.
- Send an EOF (**Control+D**).
- Send an INT (**Control+C**).

Note: The session is prevented from exiting if there are un-committed changes, this condition is indicated by a message. However, you can force an exit by immediately executing an exit command again, any un-committed changes will be discarded.

NAVIGATE IN THE CONFIG SHELL

The Config Shell operates in a hierarchy of entities. Due to the variety of entities, there are several ways for you to get to a place where you can make changes.

Starting at the root, enter the entity names to descend down through lower entities. Every entity name is an operation that descends into that entity. Similarly, type the names of entities higher in the hierarchy to ascend towards the root.

IDENTIFIERS:

Singleton entity	Require only the entity name to be uniquely identified.
List/item entity	The first level is the entity name, the second level is the item identifier (the identifier is the same identifier used by ogcli).
Multiple identifiers	A **single entity** (ssh/authorized_keys) requires an extra identifier. In this case, the hierarchy is: **ssh/authorized_keys > userid > [key_id]**.
Nested fields	The Config Shell treats nested fields as additional hierarchy levels. This applies both to arrays and maps. For arrays of complex values, each value shall also be a hierarchy level.

FIELDS, ENTITIES AND CONTEXTS

The config shell allows you to configure a number of fields which define settings.

The fields are grouped in entities that describe a small set of functionality. For example, there is a ‘user’ entity which is used to access user settings. Entities can contain sub-entities as well as simple fields.

CONTEXT WITHIN CONFIG SHELL

Once in the shell, a number of commands are available depending on the current context. The context is the current entity that is the focus of the config shell. When the shell is first started, the context is a special parent context from which sub-entities can be seen.

Once a context is selected by typing the name of the entity, it is shown in the prompt between brackets. For example, in the following snippet, the ‘user’ context is accessed and then the ‘john’ sub-entity is accessed causing the context to become ‘user john’. The ‘show’ command is used to list the entities and fields that descend from the current context.

```
config: user
config(user): show
Item names for entity user
 john matt myuser netgrp root
config(user): john
config(user john):
Entity user item john
 description
 enabled true
 no_password false
```

```
  password
  ssh_password_enabled true
  groups (array)
config(user john):
```

GLOBAL CONTEXT COMMANDS

The following commands are available on any context:

help (or '?')	Show help which is context sensitive. It will list some special details about the current context, the list of sub entities (or fields) and a list of available commands.
help <entity>	Show the help for the specific entity.
help <field>	Show the help for the specific field.
show	List the available entities and fields.
<entity>	Typing the name of an entity changes the context to focus on the named entity.
exit	Exit the command shell.

ENTITY CONTEXT COMMANDS

The following commands available on any entity context:

<field>	Show the value of a field.
<field> <value>	Change the field to the specific value.

delete	Delete the current entity. This is available when the context entity is an item in a list.
add	Append a sub entity or field to the current entity. This is only available when the context entity is a list.

APPLY OR DISCARD FIELD CHANGES

When fields and entities are changed, they are not yet applied to the system configuration but are kept staged. Items that are staged are indicated with an '*' when the `show` command is used. In addition, the `changes` command can be used to show what fields have been changed.

In the following example, the user 'john' has been changed to alter the description. The `show` command indicates the changed field with an '*'. The `changes` command lists the changed field.

```
config(user john): description "Admin"
config(user john): show
Entity user item john
 description Admin *
 enabled true
 no_password false
 password
 ssh_password_enabled true
 groups (array)
config(user john): changes
Entity user item john (edit)
 description Admin
config(user john):
```

OPERATIONS

Once a change has been made, the following commands are available:

changes	show staged changes on all entities
apply	apply changes only on the current entity
discard	discard changes only on the current entity
apply all	apply changes on all entities
discard all	discard changes on all entities

SUPPORTED ENTITIES

The following entities are supported in phase 1 of this feature and are available in release 22.06.0:

auth	Configure remote authentication, authorization, accounting (AAA) servers.
group	Retrieve or update user group information.
ip_passthrough	Passthrough entities are for retrieving / changing IP Passthrough settings.
ip_passthrough/status	The IP Passthrough status entity provides information about what part of the IP

	Passthrough connection process the device is currently at and information about the connected downstream device.
local_password_policy	Configure the password policy for local users. This includes expiry and complexity settings.
logs/portlog_settings	Check and update port log settings.
managementport	Used for working with local management console information.
port	Configure and view ports information.
ports/auto_discover/schedule	Manage Port Auto-Discovery scheduling.
system/admin_info	Retrieve or change the Console Manager appliance system's information (hostname, contact and location).
system/banner	Retrieve or change the Console Manager appliance system's banner text.
system/cloud_connect	Retrieve or change the Console Manager appliance system's cloud connect configuration.
system/model_name	Retrieve the Console Manager appliance's Model Name.
system/serial_number	Retrieve the Console Manager appliance's Serial Number.

system/session_timeout	Retrieve or change the Console Manager appliance session timeouts.
system/ssh_port	The SSH port used in Direct SSH links.
system/time	Retrieve and update the Console Manager's time.
system/timezone	Retrieve and update the system's timezone.
system/version	Retrieve the Console Manager's most recent firmware and REST API version.
user	Retrieve and update user information.

EXAMPLE CLI COMMANDS

ADDING A USER

In this example below, some commentary is added. Commentary added later is denoted with a '//' prefix.

```
# config
Welcome to the Opengear interactive config shell. Type ? or help for
help.
// Move to the user entity
```

```
config: user
config(user): help add
Add a new item for entity user.
```

```
The add command requires a unique value to identify the record.
This will be used for the username field.

Description for the item:
 Retrieve and update information for a specific user.

// Create the new user

config(user): add matt
config(user matt): show
Entity user item matt
 description
 enabled true
 no_password false
 password (required)
 ssh_password_enabled true
 username matt
 groups (array)
```

```
// Fill out some fields

config(user matt): password secretpassword
config(user matt): description Admin
config(user matt): show
Entity user item matt
 description Admin *
 enabled true
 password secretpassword *
```

```
 ssh_password_enabled true
 username matt
 groups (array)
```

```
// Edit the groups
config(user matt): groups
config(user matt groups): show
Entity user item matt field groups
config(user matt groups): add // Tab completion to show available
values
admin myuser netgrp
config(user matt groups): add admin
config(user matt groups): up // Exit the groups list
```

```
// Show and apply
config(user matt): show
Entity user item matt
 description Admin *
 enabled true
 password secretpassword *
 ssh_password_enabled true
 username matt
 groups (array)
 0 admin *
config(user matt): apply
Creating entity user item matt.
config(user matt):
```

CONFIGURING A PORT

```
config: port
config(port): help
You are here: entity port
Description for the entity:
  Configuring and viewing ports information

Names (type <name> or help <name>)
==================================
USB-A  USB-E  USB-front-lower  port03  port07  port11  port15  port19  port23
USB-B  USB-F  USB-front-upper  port04  port08  port12  port16  port20  port24
USB-C  USB-G  port01           port05  port09  port13  port17  port21
USB-D  USB-H  port02           port06  port10  port14  port18  port22

Commands (type help <command>)
==============================
exit  help  show  up
config(port): port01
config(port port01): baudrate // tab completion
110     1200    150     19200   230400  300     4800    57600   75
115200  134     1800    200     2400    38400   50      600     9600
config(port port01): baudrate 57600
config(port port01): label Router
config(port port01): control_code
config(port port01 control_code): break a
config(port port01 control_code): up
config(port port01): show
Entity port item port01
  baudrate      57600         *
  databits      8
  escape_char   ~
  label         Router        *
  logging_level disabled
  mode          consoleServer
  parity        none
  pinout        X2
  stopbits      1
  control_code (object)
    break   a *
    chooser
    pmhelp
    portlog
    power
    quit
  ip_alias (array)
config(port port01): apply
Updating entity port item port01.
config(port port01):
```

ADVANCED PORTMANAGER PM SHELL GUIDE

The Portmanager program allows you to access any serial port on the console server using `pmshell` commands.

- Routes network connection to serial ports.
- Checks permissions.
- Monitors and logs all the data flowing to/from the ports.
- Allows you to run power commands if the serial port is associated with a PDU outlet.

RUNNING `PMSHELL`

`pmshell` provides an environment that allows you to access and interact with serial ports via a number of command sequences. It lets you navigate between ports using the chooser command (`~m`). For example, you can use pmshell to connect to port 8 via the portmanager via the following command line sequence.

```
# pmshell -l port08
```

PMSHELL COMMANDS

When running `pmshell` there are a number of command sequences that you can use that begin with the ~ key.

Note: If you are connected to `pmshell` via SSH, you must add an additional ~ escape sequence.

Options	Name	Result
~c		The Single Session feature can be enabled or disabled by editing the single_session field in a given port. When a user port level administration access is logged in via pmshell, the port configuration menu can be accessed via any port by pressing the escape character (~ by default) followed by c (~c).
~b	break	Generates a BREAK on the serial port (if you're doing this over ssh, you'll need to type "~~b").
~h	portlog	Generates a history on the serial port. Displays the traffic logs for the port - must have port logging enabled.
~.	quit	Quits pmshell.
~p	power	Opens the power menu for the port. The port

Options	Name	Result
		must be configured for a PDU.
~u		Opens the list of user sessions, select by number to disconnect.
~m	chooser	Connects to the port menu - go back to the serial port selection menu.
~?	pmhelp	Displays help message.

CUSTOM CONTROL CODES FOR SERIAL PORTS

Custom control codes can be defined for ease of use per port or can be applied to all ports. For example, users could define a different Power Menu control code for every port, while having a single control code for View History that applies to all ports.

Custom control codes can be used by any user with access to the serial port. In order to run the shortcuts, the user presses the CTRL key + the keycode.

> **Note:**Only Admin users can specify short-cut control codes.

CONFIGURE CUSTOM CONTROL CODES

Admin users can configure control codes for any of the `pmshell` commands through the REST API, ogcli and the new interactive config shell.

Control code limitations are as follows:

- Cannot set multiple control codes for a port to use the same keycode
- The available key codes are a-z, excluding ‘i’ and ‘m’ as these can be triggered by commonly used keys TAB and BACKSPACE.

To disable a certain control code for an individual port, set the port’s control code to an empty string.

CONFIGURE CONTROL CODES FOR A SPECIFIED PORT (CLI EXAMPLES)

Control Codes Action	CLI Examples
Set control codes for a given port. In this example, the user sets multiple control codes for port 2	`ogcli update port port02 << 'END' control_code.break="b" control_code.chooser="c" control_code.pmhelp="h" control_code.portlog="l" control_code.power="p" control_code.quit="q" END`
Clear all control codes for a given port, in this example, port 2	`ogcli update port port02 << 'END' control_code.break="" control_code.chooser="" control_code.pmhelp="" control_code.portlog="" control_code.power="" control_code.quit="" END`

CONFIGURE A CONTROL CODE VALUE FOR ALL PORTS

To set a particular control code to one value across all serial ports, admin users can use the script `set-serial-control-codes` from the CLI as follows:

```
set-serial-control-codes CONTROL_CODE KEY
```

where:

- CONTROL_CODE - Must be one of the following values: `break, chooser, pmhelp, portlog, power` or `quit`.
- KEY - Must be a single lower case letter a-z excluding ‘i' and ‘m’ or an empty string designated by '' which is used to clear the control code.

CONTROL CODES FOR ALL PORTS VIA CLI (EXAMPLES)

Control Codes Action	CLI Examples
Set chooser control code to CTRL-a on all ports	`set-serial-control-codes chooser a`
Clear chooser control code on all ports	`set-serial-control-codes chooser ''`

DOCKER

Docker is a tool designed to make it easier to create, deploy, and run applications by distributing them in containers. Developers can use containers to package up an application with all of the parts it needs, like libraries and dependencies, and then ship it out as one package. Docker is running by default on the Console Manager. You can access commands by typing docker in the Local Terminal or SSH.

For more information on Docker, enter `docker --help`.

CRON

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

Usage:

crontab [options] file

crontab [options]

crontab -n [hostname]

OPTIONS:

-u <user> define user

-e edit user's crontab

-l list user's crontab

-r delete user's crontab

-i prompt before deleting

-n <host> set host in cluster to run users' crontabs

-c get host in cluster to run users' crontabs

-x <mask> enable debugging

To perform start/stop/restart on crond service:

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

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

To verify the current crond status:

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

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

```
crontab -l
```

To edit or create a custom crontab file:

```
crontab -e
```

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

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

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

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

Save and close the file.

INITIAL PROVISIONING VIA USB KEY

Also known as “ZTP over USB”, this feature allows provisioning an unconfigured (factory erased) unit from a USB storage device like a thumb drive.

The USB device must contain a filesystem recognized by the CM (currently FAT32 or ext4) with a file named `manifest.og` in the root directory. This file specifies which provisioning steps will be done. An article with a partial description of the file format is available here:Automated enrollment using USB.

The USB device can be inserted any time (before or after power is applied to the unit) and as long as the unit is unconfigured, the ZTP over USB process will be triggered. Here “unconfigured” has the same meaning as for ZTP: no changes made to the ogconfig data store.

> **Note:**Setting the root password on first log in counts as a config change.

The following manifest.og keys are implemented. This provides image installation, Lighthouse enrollment, and arbitrary script execution:

manifest.og contains <key>=<value> pairs. Recognized keys are:

image : Firmware image file name on the USB device's filesystem that will be flashed after boot once the image is validated

script : Configuration script to run

address : Primary Lighthouse address to enroll with

api_port : Optional port to use for the primary address when requesting enrollment

password : LH global or bundle enrollment password

bundle : Name of LH enrollment bundle

EULA AND GPL

The current Opengear End-User License Agreement and the GPL can be found at http://opengear.com/eula.

ACCESS MENU

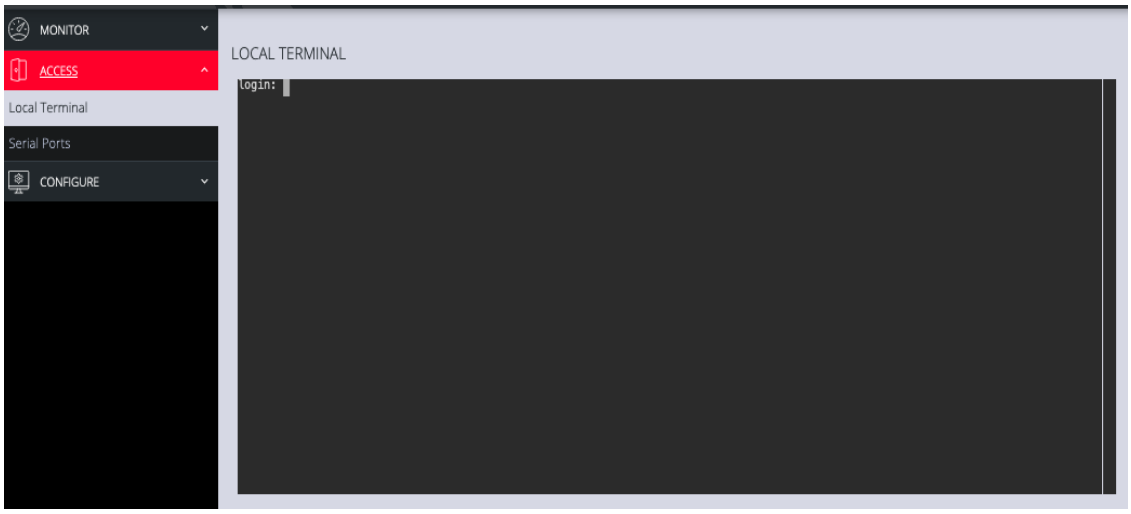
The ACCESS menu provides access to Local Terminal of the Console Manager. It also provides SSH and Web Terminal access to specific ports.

LOCAL TERMINAL

ACCESS > Local Terminal

The Console Manager includes a web-based terminal. To access this bash shell instance:

1. Select **ACCESS > Local Terminal**



2. At the login prompt, enter a username and password.
3. A bash shell prompt appears.

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

To close a terminal session, close the tab, or type exit in the Web Terminal window. The session will timeout after 60 seconds.

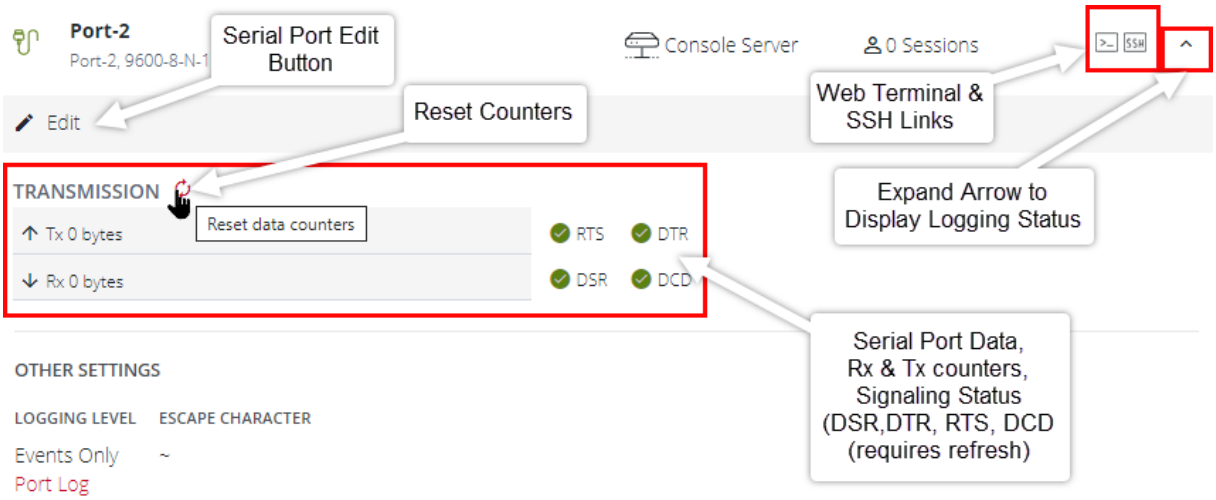
Tip: The default for the CLI session timeout is “never” (value of 0), however, the Web session timeout defaults to 20min. The web session time-out will kill the CLI session even though the CLI session itself is set to “never”.

ACCESS SERIAL PORTS

ACCESS > Serial Ports

Tip: Ensure you are on the **ACCESS > Serial Ports** page and not the similar **CONFIGURE > Serial Ports** page.

The **ACCESS > Serial Ports** page allows you to quickly locate and access specific ports via Web Terminal or SSH link shown in the image below.



Click the **Expand arrow** to the right of the port to see the Port Logging status or access the port **Edit** button, which is a link to the **CONFIGURE > Serial Ports** page. (ogcli: `ogcli get ports/ports_status`).

The following information is displayed under **Access > Serial Ports** when the individual serial ports are expanded:

- Rx byte counter (counter reset requires 'Admin' or 'port config' rights)
- Tx byte counter (counter reset requires 'Admin' or 'port config' rights)
- Signaling information (DSR, DTR, CTS (see tip), RTS and DCD)

  > **Tip:** CTS information is not displayed in the UI but is available via the ogcli query `ogcli get ports/ports_status`.

- Logging information.

QUICK SEARCH

To find a specific port by its port label, use the **Quick Search** form at the top-right of the **ACCESS > Serial Ports page**.

Ports have default numbered labels. You can edit the port label for a given serial port under **CONFIGURE > Serial Ports**. Click the **Edit** button to open the **EDIT SERIAL PORT** page.

ACCESS USING WEB TERMINAL OR SSH

To access the console port via the Web Terminal or SSH:

1. Locate the particular port on the **ACCESS > Serial Ports** page and click the expand arrow.
2. Click the **Web Terminal** or SSH link for the particular port.

- Choosing **Web Terminal** opens a new browser tab with the terminal.
- Choosing **SSH** opens an application you have previously associated with SSH connections from your browser.

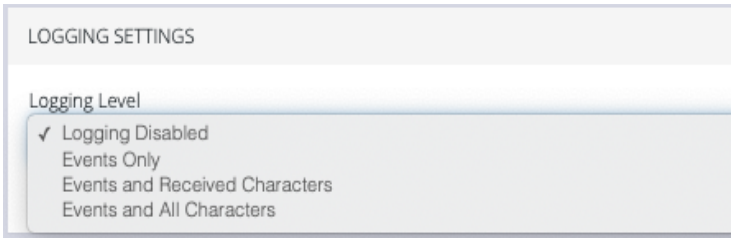
Note:MS WIndows does not connect to puTTY by default. You may need to install the WinSCP program to launch puTTY from the Opengear WebGUI SSH Serial Port button.

SERIAL PORT LOGGING

The port logging facility and severity associated with the serial port logs is controlled and set at the **Configure > Services > Syslog > Global Serial Port Settings** page.

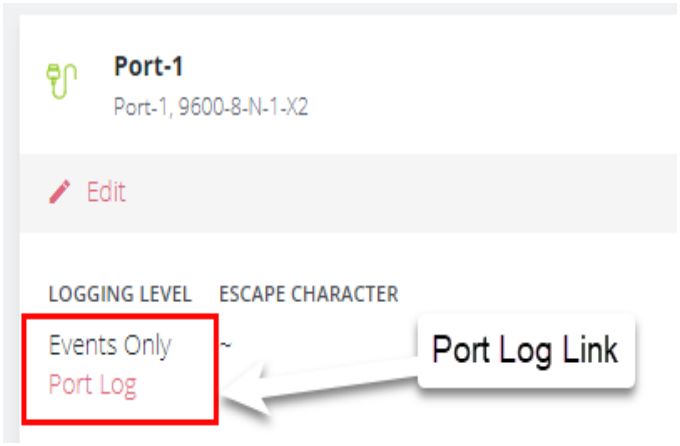
There is a separate setting to enable sending of serial port logs to remote side.

Note: Serial port logging is disabled by default. The logging level for each serial port is set at Logging Settings in **Configure > Serial Ports > Edit** .



DISPLAY PORT LOGS

Tip: The log is accessed by clicking the **Port Log** link on the **ACCESS > Serial Ports** page. The link is only available when port logging is enabled.


Port-1
Port-1, 9600-8-N-1-X2
Edit
LOGGING LEVEL
ESCAPE CHARACTER
Events Only
Port Log
~
Port Log Link

CONFIGURE MENU

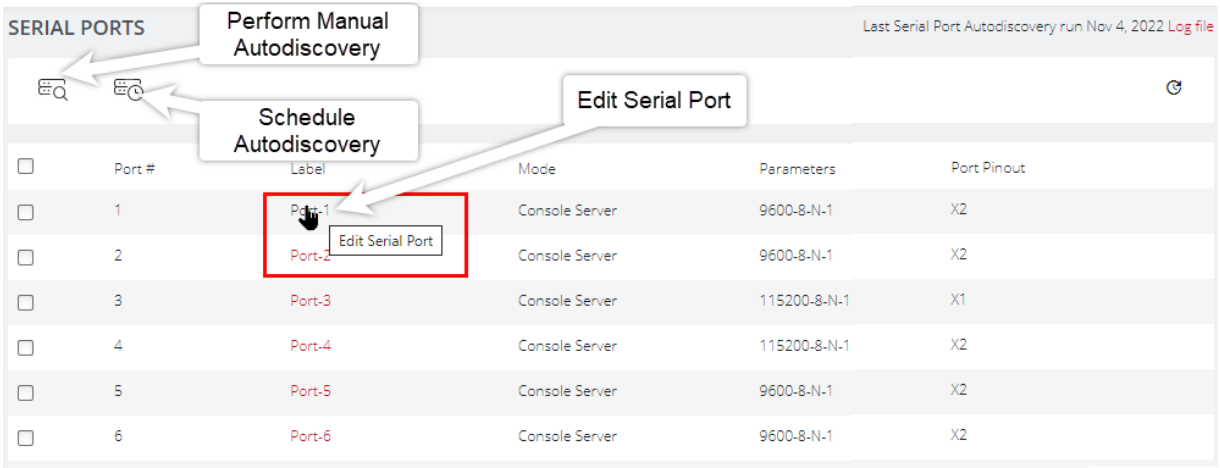
This section provides step-by-step instructions for the menu items under the CONFIGURE menu.

CONFIGURE SERIAL PORTS

CONFIGURE > Serial Ports

> **Tip:** Ensure you are on the **CONFIGURE > Serial Ports** page and not the similar **ACCESS > Serial Ports** page.

Navigate to **CONFIGURE > Serial Ports**; a list of serial ports is displayed. On this page you can configure and edit specific ports. Click the **Edit** button (pencil icon) to the right of the port to display the port editing page.



EDIT SERIAL PORTS

From the **Configure > Serial Ports** page, click the **Port label** text in the Label column. The **Edit Serial Port** page is displayed.

Edit Serial Port Properties		
Field	**Options**	**Definition**
Label	Default or Custom	The serial port unique identifier. This can be used to locate this port using the **Quick Search** form on the **ACCESS > Serial Ports** page.
Mode	Disabled Console Server Local Console	**Console Server** mode allows access to a downstream device via its serial port. **Local Console** mode allows access to the OM device's console through a serial port.
Port Pinout	Fixed - X2 Cisco Straight	The pin-out type is fixed on the CM8100.
Port Pinout CM8100-10G	Selectable - X2 Cisco Straight	The pin-out type is software selectable on the CM8100-10G
Baud Rate	Baud rate	Select the Baud rate expected for this port. From 50 to 230,400 bps.

Data Bits	Integer	The data bit length for character.
Parity	None, Odd, Even, Mark, Space.	The parity type for character.
Stop Bits	1, 1.5, 2	The Stop bit length used in character.
Escape Character	~	The character used for sending OOB Shell commands.
LOGGING SETTINGS		
Logging Level	Disabled Events Only Events & Received Characters Events & All Characters	Specify the level of detail you require in the logs. Logs may also be sent to a Syslog server. Other settings to consider are: "GLOBAL SERIAL PORT SETTINGS" under Services > Syslog. "Send Serial Port Logs" under Services > Syslog > Add Syslog Server
PORT IP ALIASES		
IP Address	Alias IP Address and interface type.	Allocate an IP address for dedicated access to a specific serial port.

ASSIGNING UNIQUE IP ADDRESSES FOR EACH CONSOLE PORT

Note: For further information about assigning unique IP addresses for each console port see the Knowledge Base article Configure IP alias for serial ports.

CONFIGURE SINGLE SESSIONS FOR PORTS

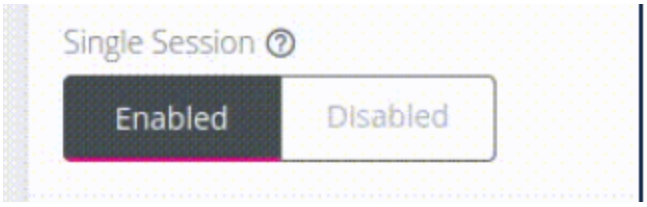
Single Session Port Config, or *Single Session* is a feature that can be enabled on a given port to prevent multiple users from connecting to that port or limit the port to a single concurrent connection. This feature is port-specific and is disabled by default. This feature needs to be enabled on a port-by-port basis. It can be enabled on all types of serial ports (including USB).

Similarly to config shell, a single session must be enabled/disabled on a port by port basis, currently it cannot be enabled on all ports.

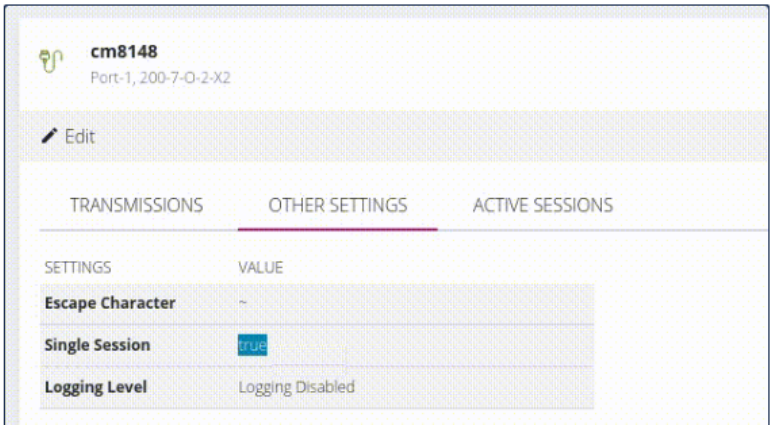
SINGLE SESSION ENABLED

IN THE WEB UI

Single Session can be viewed and configured in the Web UI. It is enabled (or disabled) in the configure page for a given serial port. The buttons to connect to a serial port are automatically disabled when the feature is enabled and the session is in use.



You can also confirm the session in the access/serial_ports page, and also see if the feature is enabled.



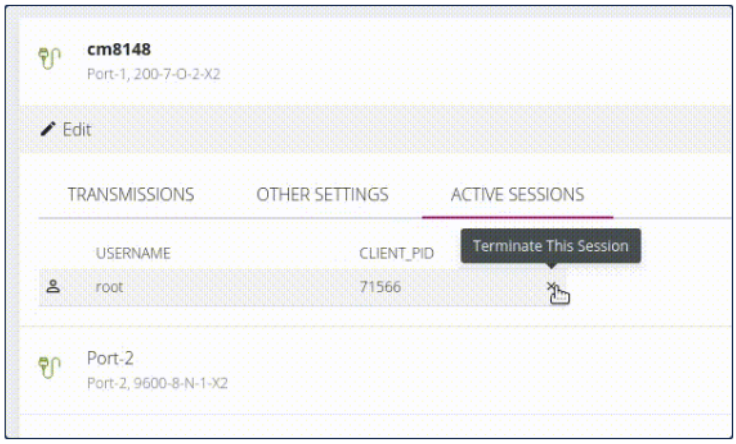
When the Single Session feature is enabled and the port is in use, if a subsequent user attempts to connect to the port, the connection is declined and the second user will receive the message:

```
Unable to connect. Another session is currently active.
Please disconnect from the current session before attempting to
connect again.
```

The pmshell will exit, and the user who tried to connect will not have seen the first user's session. Everything they have done will remain confidential.

The single session is indicated next to the user

If necessary, a user's single session can be terminated with the **Terminate all sessions** button which is shown beside individual users. This will re-enable the **Single Session** button and allow you to connect.



IN CONFIG SHELL

The Single Session feature can be enabled or disabled by editing the single_session field in a given port. When a user port level administration access is logged in via pmshell, the port configuration menu can be accessed via any port by pressing the escape character (~ by default) followed by c (~c).

You can access a port with the following commands, the following example will access “Port 1”:

```
config: port
config(port): port01
```

The port configuration might look like this below. You can see for this port, single_ session is set to false, so the feature is disabled:

```
config(port port01): show
Entity port item port01
   baudrate 9600
   databits 8
   escape_char ~
   label                    Port-1
   logging_level    disabled
```

```
mode                    consoleServer
parity none
pinout                  X2
portnum 1
single_session   false
stopbits 1
control_code (object)
   break       ""
   chooser    ""
   pmhelp ""
   portlog      ""
   power       ""
   quit ""
ip_alias (array)
```

The feature is enabled by typing `single_session true`, then apply the change.

```
config(port port01):      single_session true
config(port port01):      apply
Updating entity port      item port01.
config(port port01):      show
Entity port item              port01
    baudrate       9600
...
single_session                true
...
ip_alias (array)
```

SINGLE SESSION BEHAVIOR

The following table describes single session feature behavior in various circumstances.

Q.	What occurs if users are connected to the port with the feature disabled, then the feature is enabled while users are still connected?
A.	Users who are already connected will continue to be able to use the port. If they leave, they will not be able to rejoin (unless there are no active sessions). Their current session will continue as normal, however, their session can be manually terminated from config shell (config(port_session):) or from the Web UI from the Access/Serial Ports page.
Q.	What if a user needs to be removed from a port?
A.	Administrators can remove the right for a given user to access a port. They can also manually remove them from the port in the config shell (config(port_session):) or the Web UI from the Access/Serial Ports page.
Q.	What if someone tries to join a port that is already in use?
A.	The user who tries to join will be prevented from doing so and receive a notification. The person currently using the port will be unaffected and not be aware of the attempt.
Q.	Is there a way to enable the feature for every port?
A.	Currently, the feature must be enabled/disabled on a port-by-port basis.
Q.	What if I enable this port on localConsole mode?
A.	The feature is ignored on local console mode and is only active for Console Server mode. It also remains ignored if the port mode is set

	to disabled.

CONFIGURE RAW TCP ACCESS FOR SERIAL PORTS

The Raw TCP Access feature is an option under the ports endpoint and provides a means of accessing serial ports directly through netcat (nc), or Telnet.

Note: Raw TCP can only be enabled when the port is in **consoleServer** mode.

Raw TCP is enabled or disabled through the WebUI, Config Shell, or through ogcli. When enabled, Raw TCP will open a TCP socket on a TCP port in the range of 40XX, where XX corresponds to the serial port number on which Raw TCP access is enabled. Any TCP messages sent to this port will be relayed to the corresponding serial port.

The Console Manager serial ports can be configured to operate in Raw TCP mode on a port-by-port basis.

Pre-defined firewall services allow Raw TCP connections through the firewall. These services correspond to each serial port on the device.

Caution: Raw TCP access bypasses any authentication methods. When Raw TCP access is enabled on a serial port, anyone with network access will be able to access that serial port, and any devices connected serially to it. **This feature should only be used on a secure network.**

Note: Raw TCP access is disabled by default on Opengear devices. Users must enable Raw TCP access on a serial port through the WebUI, Config CLI or ogcli.

SERVICE IMPLEMENTATION

Raw TCP access allows you to access serial ports on a device directly by connecting to a TCP port in the range 40XX.

In order to achieve Raw TCP access, you first need to allow TCP packets through port 4002 in the firewall:

1. Navigate to the **Firewall Management** page in the WebUI.
2. Add a `raw_tcp_serial02` service, which corresponds to serial port 2.
3. Add the service for the firewall zone the service will be connecting over; in the example below, it is the LAN zone. Check the service has been correctly added (as shown below).

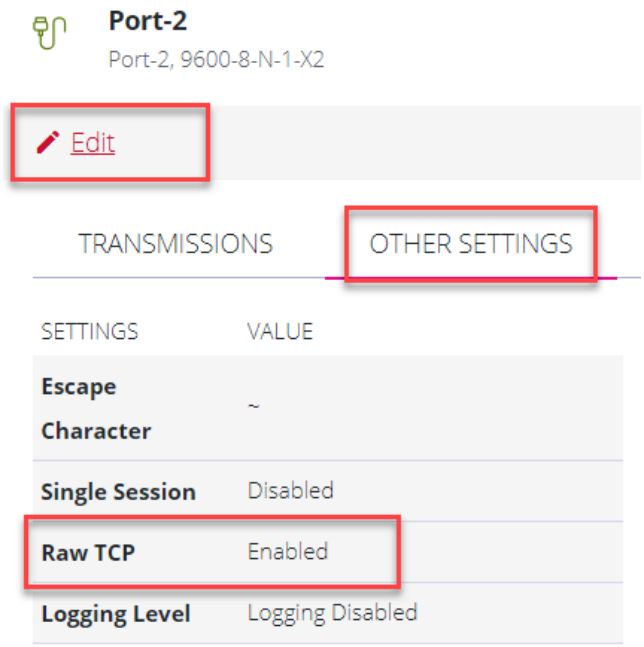
Once this service has been added to the correct firewall zone, you can create a Raw TCP connection to the target port.

> **Note:** `tcp_serial` service needs to be manually added to relevant firewall zones after raw tcp has been enabled on a port.

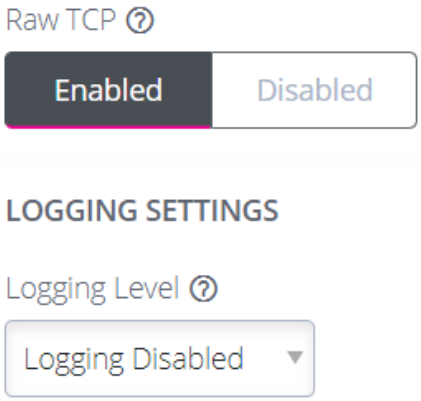
WEBUI CONFIGURATION

Raw TCP access can be enabled or disabled on a selected serial port through the WebUI. When looking at the serial port access page, the enabled/disabled status of Raw TCP access is visible under the **Other Settings** tab for each serial port.

1. In the WebUI, navigate to **Access > Serial Ports** and click the drop-down arrow to the right side of the target port. This will display the port settings, including the Raw TCP status.
2. Click the **Edit** text to open the **Edit Serial Port** page:



3. At the **Edit Serial Port** page, scroll down the page to see the Raw TCP settings:



4. To Enable Raw TCP, click the **Enabled** button then click **Apply** at the bottom of the page. A confirmation message is flagged when Raw TCP is successfully enabled.

CONFIG CLI CONFIGURATION

Raw TCP can be configured through the Config Shell. Navigate to the port endpoint and enter the context of the target serial port (for example, serial port 2 is used in the following procedure):

```
config: port port02
config(port port02): show
Entity port item   port02
  baudrate            9600
  databits             8
  escape_char     ~
  label                  Port-2
  logging_level   disabled
  mode                consoleServer
```

```
  parity none
  pinout X2
  portnum           2
  raw_tcp false
  single_session false
  stopbits 1
  control_code (object)
    break ""
    chooser    ""
    pmhelp    ""
    portlog     ""
    power       ""
    quit          ""
  ip_alias      (array)
```

To enable Raw TCP access:

```
config(port port02): raw_tcp true
config(port port02): apply
Updating entity port item port02.
```

To disable Raw TCP access:

```
config(port port02): raw_tcp false
config(port port02): apply
Updating entity port item port02.
```

OGCLI CONFIGURATION

To enable Raw TCP access on a port through ogcli, users can use ogcli update to set raw_tcp to **true** on the target port (the device information in the ogcli command below is shown as an example):

```
root@om2216-l-tp1-p3:~# ogcli update port port02 raw_tcp=true
```

To disable Raw TCP, set raw_tcp to **false** on the target port:

```
root@om2216-l-tp1-p3:~# ogcli update port port02 raw_tcp=false
```

You can check that the socket is active by running:

```
systemctl status raw-tcp-serial-port02.socket
```

AUTODISCOVERY

The Autodiscovery feature attempts to discover the host name of connected devices, this uses the hostname of the device used to set the port label, so as to set the hostname as the port label of each serial port. This can save the need to manually provide hostnames during setup.

Autodiscovery will attempt to discover port settings even if the hostname discovery fails. The first discovery run uses currently configured port settings such as the current baud rate, etc. Thereafter, it will fetch or use a single set of pre-configured credentials to login and discover the hostname from e.g. the OS prompt, for devices that do not display hostname pre-authentication.

Syslogging enhancement assists in the diagnosis of common issues (for example, no comms or, hostname failed validation). Autodiscovery does not collect a hostname when there is a communication issue between the console server and the target device. The logs are saved for the last-run instance of autodiscovery.

The UI displays error messages and logs with the reason for auto-discovery failure, for example:

- Authentication failed.
- Communication issue with the target device.
- Password to renew before being able to authenticate to the target device.
- Abnormal characters or strings detected.

Autodiscovery has been enhanced to discover baud rate and pinout. The UI now indicates if ports are scheduled for discovery.

The **Serial Ports** page also allows you perform an Autodiscovery on selected ports. Autodiscovery of console ports attempts to set the port label by setting the baud rate to various rates (in the following order): 9600, 115200, 38400, 19200, and 57600.

> **Tip:** Autodiscovery on other Baud rates may be done by manually running the port_discovery script from the Web Terminal.

Autodiscovery may be done manually by clicking **Perform Autodiscovery**.

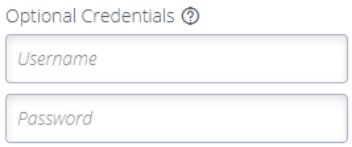
AUTODISCOVERY ENHANCEMENTS

From the 22.11 release, the following parameter enhancements have been added to the port_discovery script which can be configured via the REST API or CLI:

- `--username` and `--password`
- `--apply-config` and `--no-apply-config`

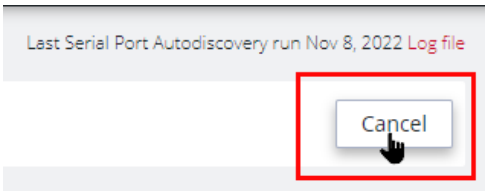
- `--auth-timeout`
- `--hostname-pattern`

The `--username` and `--password` options can also be configured via the UI under *Optional Credentials*.
If the values are provided (optional), they will be used to attempt login to obtain the hostname to a downstream serial device. You can only specify a single username and/or password to try on all devices.



CANCEL AUTODISCOVERY

Port Autodiscovery may be canceled *while running* by clicking on the **Cancel** button at the top-right of the Serial Ports window of the UI.

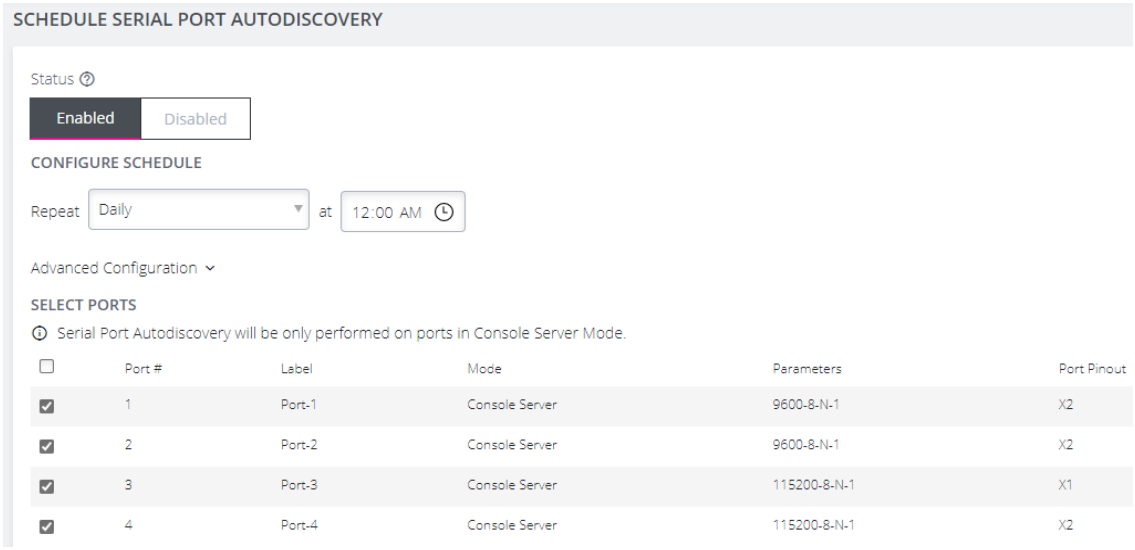


SCHEDULE AUTODISCOVERY

Autodiscovery can be scheduled periodically as required by clicking the **Schedule Autodiscovery** button in the **Serial Ports** window.

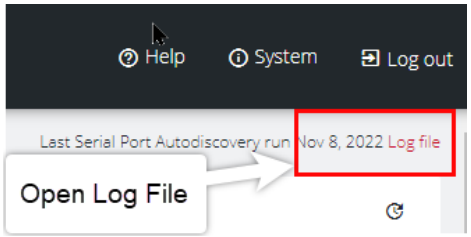
The **Schedule Autodiscovery** window allows you to select the ports and specify a time and period for port detection to run. Activate the schedule by clicking on the **Enabled** button.

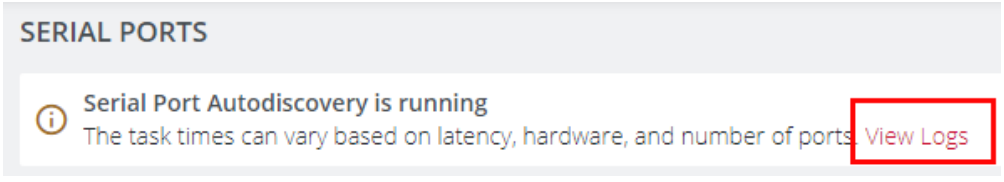
The Serial Port Autodiscovery Page:



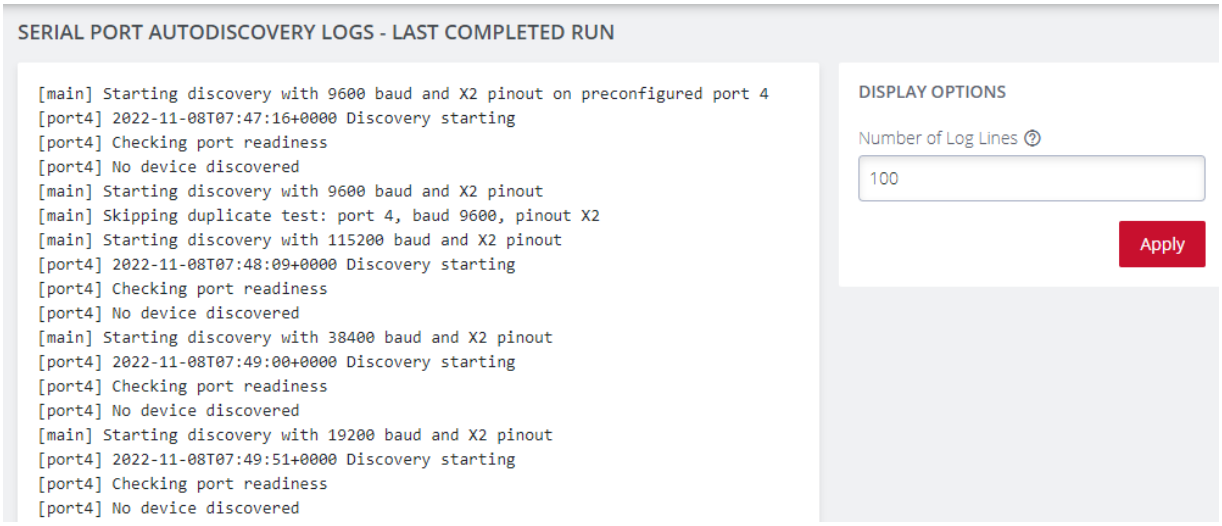
RETRIEVE PORT DISCOVERY LOGS

At the top-right of the UI window, click on the **Log File** red text to retrieve the port discovery logs or by clicking on the **View Logs** red text in the **autodiscovery running** banner.





Port Discovery Log File Example:



LOCAL MANAGEMENT CONSOLES

CONFIGURE > Local Management Consoles

This feature allows administrators to log in and configure the OM via the RJ-45 or USB ports on the device. You can edit settings or disable the local RJ45 serial console (Cisco straight -X2 pinout) and the USB serial console (needs user supplied micro-USB to USB-A cable).

This feature allows administrators to log in and configure the CM via the RJ-45 ports on the device. Not accessible by USB.

To edit the settings of a local management console:

1. Navigate to **CONFIGURE > Local Management Consoles**. Here you'll see a list of consoles.
2. Locate the console you want to manage, then, click on the **Edit Management Console Port** button (pencil icon) under **Actions**.

3. On the **Edit Local Management Console** page you can set the parameters for:
    - **Baud Rate**
    - **Data Bits**
    - **Parity**
    - **Stop Bits**
    - **Terminal Emulation**
    - Enable or disable **Kernel Debug Messages**
    - Enable or disable the selected **Management Console**

> **Note:**Enabling **Kernel Debug Messages** can only be applied to a single serial management console.

To disable a local management console:

1. Click **CONFIGURE > Local Management Consoles**.
2. Click on the **Disable Management Console Port** button under **Actions** next to the console you wish to disable.

LIGHTHOUSE ENROLLMENT

CONFIGURE > Lighthouse Enrollment

Opengear appliances can be enrolled into a Lighthouse instance, providing centralized access to console ports, automation, and central configuration of Opengear devices.

Lighthouse central management uses a persistent, public key authenticated SSH tunnels to maintain connectivity to managed console servers.

All network communications between Lighthouse and each console server (e.g. access to the web UI), and the console server's managed devices (e.g. the serial consoles of network equipment), is tunneled through this SSH management tunnel.

The below articles and Lighthouse user guide contain further information about Lighthouse Enrollment:

Manual enrollment using UI or CLI

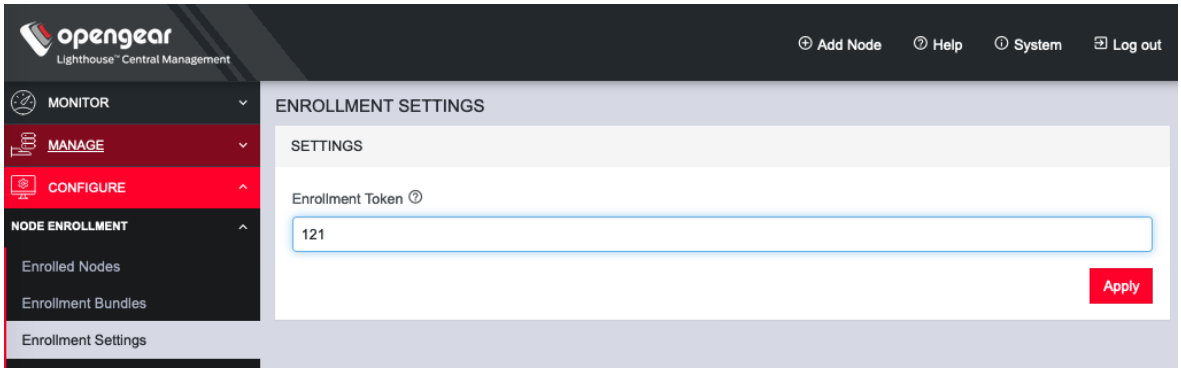
How do I add Nodes to Lighthouse

Lighthouse User Guide

MANUAL ENROLLMENT USING UI

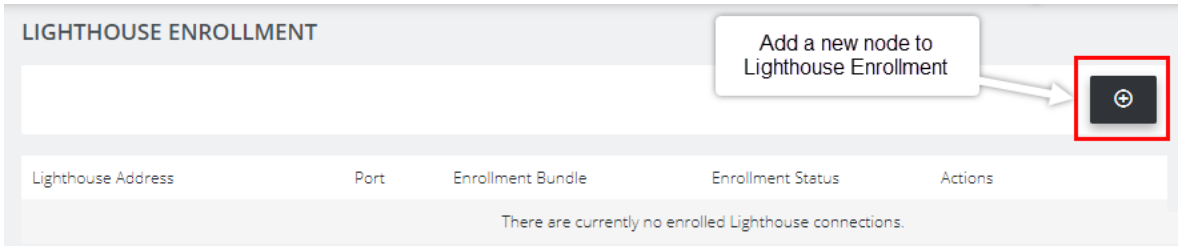
Note: To enroll your Console Manager to a Lighthouse instance, you must have Lighthouse installed and have an enrollment token set in Lighthouse.

1. In Lighthouse. Set a CM enrollment token, click on **CONFIGURE > NODE ENROLLMENT > Enrollment Settings** page, and enter an **Enrollment Token**.

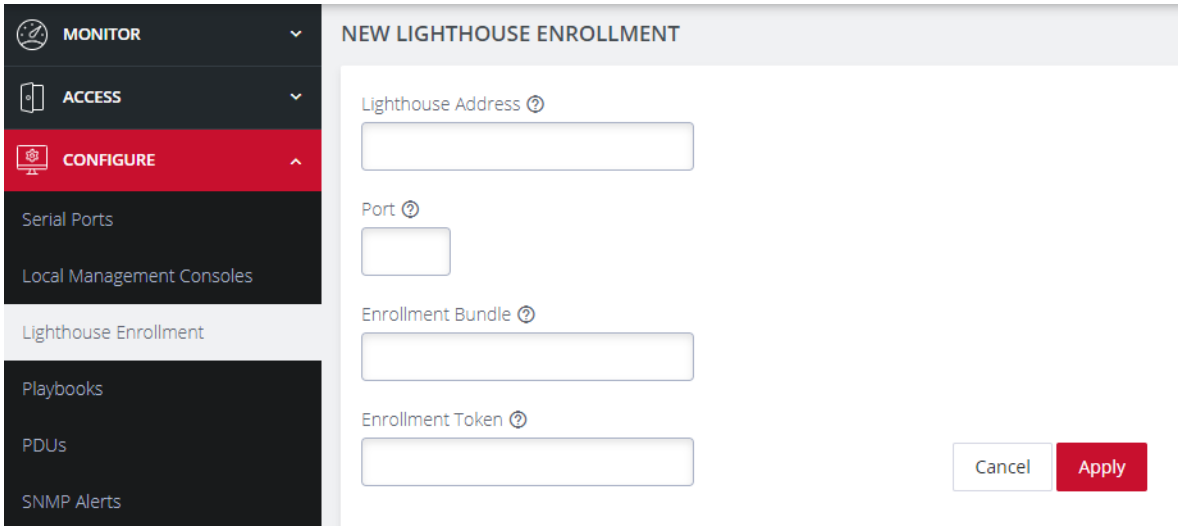


Tip: The same token will be entered in the NEW LIGHTHOUSE ENROLLMENT page of the Console Manager.

2. Enroll your Console Manager in this Lighthouse instance:
   Click **CONFIGURE > Lighthouse Enrollment**
3. Click on the **Add Lighthouse Enrollment** button on the top-right of the page.
   The **New Lighthouse Enrollment** page opens.



4. Enter the IP address or fully qualified domain name of the Lighthouse instance and the **Enrollment Token** you created in Lighthouse. Optionally enter a **Port** and an **Enrollment Bundle** (see the Lighthouse User Guide for more information about Bundling).



5. Click the **Apply** button. A flag will confirm the enrollment.

> **Note:**Enrollment can also be done directly via Lighthouse using the Add Node function. See the Lighthouse User Guide for more instructions on enrolling Opengear devices into Lighthouse.

MANUAL ENROLLMENT USING THE CLI

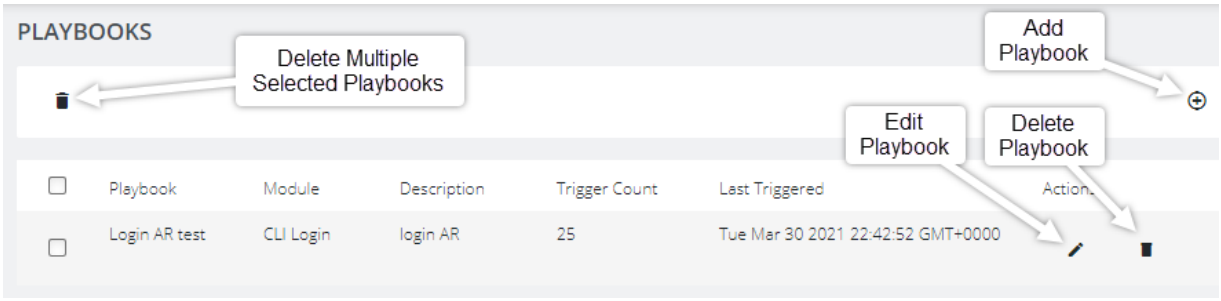
For complete instructions on Lighthouse Enrollment via the CLI please refer to this link: Manual enrollment using UI or CLI .

PLAYBOOKS

CONFIGURE > Playbooks

Playbooks are configurable systems that periodically check if a user-defined **Trigger** condition has been met. Playbooks can be configured to perform one or more specified **Reactions** when a specific trigger event occurs.

The Playbook Landing Page:



CREATE OR EDIT A PLAYBOOK

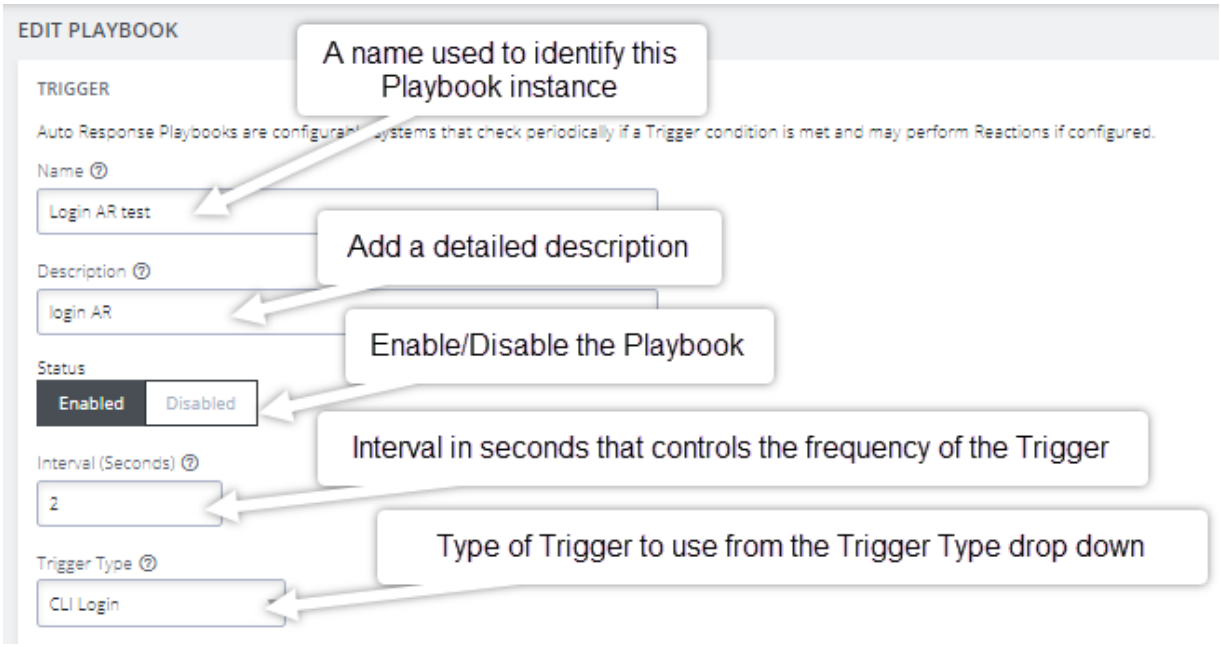
CONFIGURE > Playbooks > Add Playbook

To create a new Playbook:

Navigate to the **Configure > Playbooks** page.

Click the **Add Playbook** button (top-right) to create a new **Playbook**. The **Edit Playbook** page is displayed. Complete the required Playbook setup information as detailed in the following procedures.

TRIGGER SECTION:



1. Enter a unique **Name** for the **Playbook** that reflects its purpose.
2. Add a detailed **Description** that will help others to understand what it does.
3. Select **Enabled** to activate the **Playbook** after you have created it.
4. Enter an **Interval** in seconds to control the frequency that the **Trigger** will be checked.
5. Choose the type of **Trigger** to use from the **Trigger Type** drop down.

Tip: See the Trigger Type table on the following page for additional trigger type information.

TRIGGER TYPES:

Trigger	Reaction Description
CLI Login	Triggers upon Login or Logout events. Select either or both.
CLI Login Failure	Monitor the terminal and trigger on failed user login attempts.
Cell Connection	Triggered whenever the cellular connection state changes. This Trigger type is only compatible with cellular units.
Cell Message	Triggered when an SMS message that matches the user-defined message pattern. Cellular units only.
Cell Signal Strength	Triggered if the cellular signal strength moves below a user-defined percentage.
Curl	Periodically attempts to perform a HTTP request using curl and triggers the Playbook reaction based on the results.
Custom Command	Periodically runs a custom Shell command and triggers the Playbook reaction upon failure.
Load	Monitors the system load average and triggers the Playbook if it breaches the user-defined acceptable range.
Memory Usage	Triggered if the system memory usage exceeds the user-defined percentage threshold.

Network Settings	Monitors network interfaces for specific attributes and triggers a user-defined response when they change.
Ping	Periodically pings an address and triggers a user-defined response upon failure.

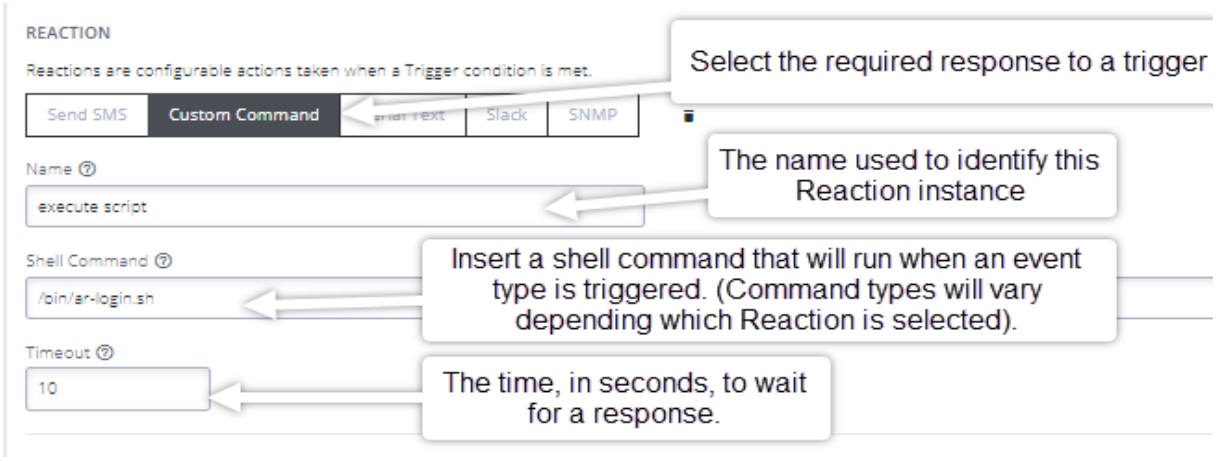
Continued...

Trigger	Description
Serial Login	Monitors selected serial ports and triggers a user-defined reaction upon user login and logout events.
Serial Pattern	Monitors serial ports and triggers a reaction when data matching a pattern is received on specific ports.
Serial Signal	Monitors selected serial ports and triggers when signals are changed.

REACTION SECTION:

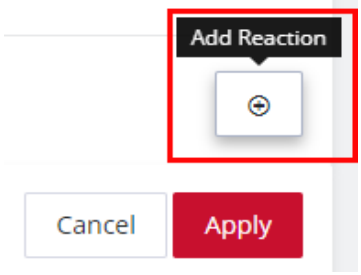
In this section you customize the response to the Trigger that you created.

1. Clicking on each **Reaction** opens a custom screen to provide necessary information.

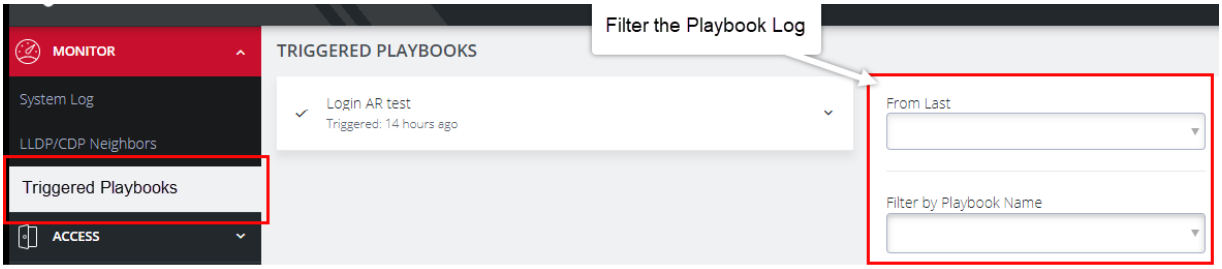

REACTION
Reactions are configurable actions taken when a Trigger condition is met.
Send SMS
Custom Command
Slack
SNMP
Select the required response to a trigger
Name
execute script
The name used to identify this Reaction instance
Shell Command
/bin/ar-login.sh
Insert a shell command that will run when an event type is triggered. (Command types will vary depending which Reaction is selected).
Timeout
10
The time, in seconds, to wait for a response.

Continued...

2. To create additional Reactions, click the **Add Reaction** button.



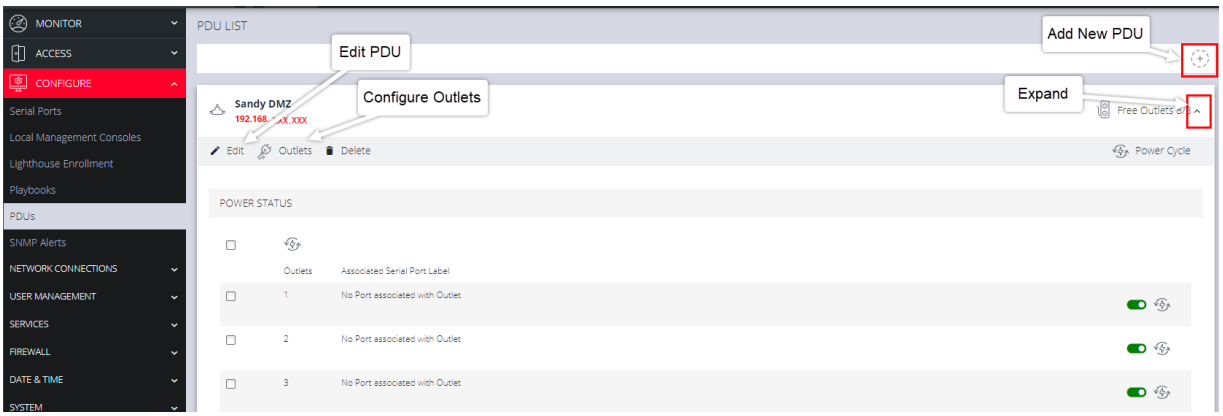
3. When you are finished, click **Apply**. A banner confirms that the Playbook settings are saved, if the Playbook is **Enabled** it is activated.
4. To monitor current **Playbooks**, click on the **Monitor > Triggered Playbooks** menu (shown below). Select the time period if desired and filter by **Name** of **Playlist** to view any that have been triggered.



PDUS

CONFIGURE > PDUs

One or more Power Distribution Units **(PDUs)**, both **Local** and **Remote** can be monitored. To add information for a **PDU**, select **Configure > PDUs**.



ADD AND CONFIGURE A PDU

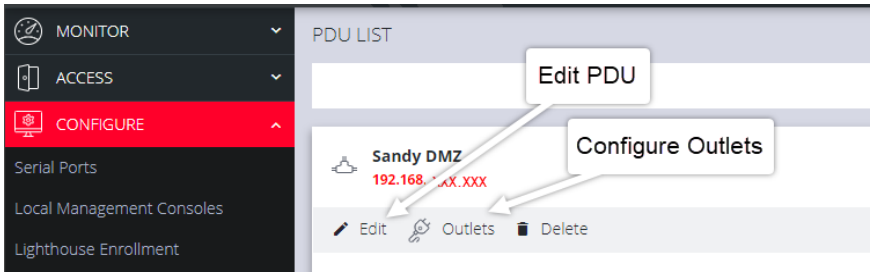
PDU configuration definitions are provided in the on the "PDU Settings Table" on the next page.

1. In the PDU List page, click the **Add New PDU** button. The **Edit** page opens.
2. Enter a meaningful **Label** that will easily identify this **PDU**.
3. Select the **Monitor** checkbox.
4. Select **Local** or **Remote**.

> **Note:**Note that **Local** or **Remote** have different settings forms.

5. Complete the **Local** or **Remote** settings in accordance with the "PDU Settings Table" on the next page.

6. Click on the **Configure Outlets** link, assign a port for each of the PDUs' ports and enter a meaningful name for each outlet.



7. When you are finished, click **Apply**. A green banner confirms your settings.

PDU SETTINGS TABLE

PDU Settings	
Label	Enter a meaningful label that will easily identify the individual **PDU**.
Monitor	Click to check this box to monitor the outlet's status.
Mode	Note that (**Local** or **Remote** have different settings forms).
Driver	Select the appropriate driver compatible with this PDU.
Local Mode Only	
Port	The serial port to which the PDU is connected.
Username	Enter the Username to use when connecting.

Password	User password to use when connecting to the device.

The table is continued on the following page...

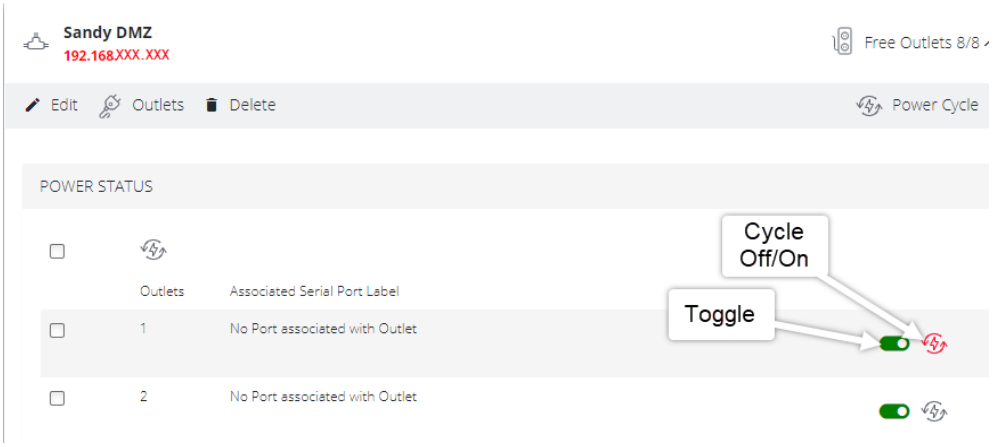
Continued...

Remote Mode Only	
Address	The remote address of the PDU.
SNMP Protocol	Click the drop-down arrow and select the correct transport protocol used to communicate with the PDU. The default value is UDP.
Version	The version of SNMP to use, V1, V2c and V3 are supported. The default value is V1.
Community	Enter a group name authorized to communicate with the device for SNMP versions 1 and 2c.

After you have created **PDUs**, you can **Edit** or **Delete** them from the **Configure > PDUs** page.

PDU OPERATION

After the PDU has been created and configured, PDU operation is simple. For any PDU that has Monitoring set to **Enabled**, the **Toggle** on/off switch will power-on or power-off the PDU, and the **Cycle** button cycles the PDU through a power-down and power-up cycle.



SYSTEM ALERTS

CONFIGURE > System Alerts > General/Power/Temperature/Networking

> **Tip:** For more detailed information about configuring SNMP Alerts see the individual topic pages that follow.

System Alert Managers can be added or deleted under Configure > Services > "SNMP Alert Managers" on page 328, for the following:

- **General**: Covers notification for the following causes.
  - **Authentication**: Notifies when a user attempts to log in via SSH, REST API, Web GUI, or the device's serial ports. An alert is sent regardless of whether the log in has succeeded or failed.
  - **Configuration Change**: For changes that occur to the system configuration.
- **Power**: When voltage SNMP alerts are enabled, network operators are immediately notified should the PSU begin operating outside design tolerances.

- **Temperature**: When system temperature alerts are enabled, network operators are immediately notified should the system begin operating outside user-defined tolerances.
- **Networking (Cell Signal Strength)**: Be notified when cell signal strength leaves or re-enters the selected range, or when the network link state changes. A slider adjusts the upper and lower signal strength.

Tip: Manage the system settings on the **CONFIGURE > System Alerts > System Alerts** pages.

SYSTEM ALERTS - GENERAL

AUTHENTICATION

CONFIGURE > System Alerts > General > Authentication

Notifies when a user attempts to log in via SSH, REST API, or the device's serial ports. An alert is sent regardless of whether the log in has succeeded or failed.

1. Navigate to **Configure > System Alerts > General > Authentication**.
2. Click on the **Enabled** button to activate the function.
3. Click **Apply**. The **Details Saved** banner confirms your settings.

CONFIGURATION

CONFIGURE > SNMP Alerts > System > Configuration Change

Notifies of changes that occur to the system configuration.

1. Navigate to Configure > SNMP Alerts > System > Configuration.
2. Click on the **Alerting** button to activate the function.
3. Click **Apply**. The **Details Saved** banner confirms your settings.

SYSTEM ALERTS - POWER

Configure > System Alerts > Power > Voltage

The PSU is one of the most critical part of the Console Manager so it is essential to ensure that the PSU is operating within its design tolerances.

When voltage SNMP alerts are enabled, network operators are immediately notified of PSU failures (subject to network connectivity and latency). Should the PSU begin operating outside design tolerances, PSU-related SNMP Alerts will trigger an alert for the following conditions:

- Output DC voltage of both PSUs
  If the voltage drops too low, it risks the Console Manager going into brown-out state. If it gets too high, it can damage components.

System generated SNMP Alerts send SNMP traps to a remote SNMP manager which alerts the user of system events. The Console Manager can send network, power and system events to the remote SNMP manager.

> **Tip:** The Console Manager can send network, power and system events to the remote SNMP manager.

ENABLE POWER SUPPLY SYSLOG ALERTS

Configure > System Alerts > Power > Voltage

The System Voltage Range alert sends an alert when the system reboots or the voltage on either power supply leaves or re-enters the fixed voltage range between 11.4V to 12.6V (SNMP) (or 11V to 13V Syslog).

1. Navigate to **Configure > System Alerts > Power > Voltage**.
2. Click on the **Enabled** button to activate the function.

> **Note:**The **Disabled** button de-activates the power syslog function and power alerts will be stopped until activated again

Syslog Alert Severity

Configure > Syslog > Add Syslog Server

3. For **Power Lost** alert, click the drop-down list and select the severity level required (default level is **3 - ERROR**) when power level is outside the pre-set range.
4. For **Power Restored** alert, click the drop-down list and select the severity level required (default is **6 - INFO**) after an error condition has been fixed.
5. Click **Apply**. The **Details Saved** banner confirms your settings.

When an event occurs that causes the voltage range on any power supply to leave or re-enter the configured voltage range, it will cause an SNMP alert to be triggered. The alert will report the event type and identity and status of the PSU, as in the example below.

```
Nov 03 06:09:35 om2232 system-alerts[850]: Redundant Supply Active (PSU0 online, PSU1
online)
Nov 03 07:05:02 om2232 system-alerts[850]: Redundant Supply Inactive (PSU0 offline, PSU1
online)
Nov 03 07:05:05 om2232 system-alerts[850]: Redundant Supply Active (PSU0 online, PSU1
online)
```

To view log severity messages locally, use the journal tool command `journalctl -f -u alert-logger -o verbose` where: f = follow. Check the alert-logger using the `systemctl status alert-logger` command.

SYSTEM ALERTS - NETWORKING (CONNECTION STATUS)

Configure > SNMP Alerts > Networking > Network Connection Status

The alert related to this functionality is the Network Connection Status which sends an alert when cell signal strength leaves or re-enters a user-defined range, or, when the network link state changes. A slider adjusts the upper and lower signal strength limits.

CONFIGURE SIGNAL STRENGTH ALERTS

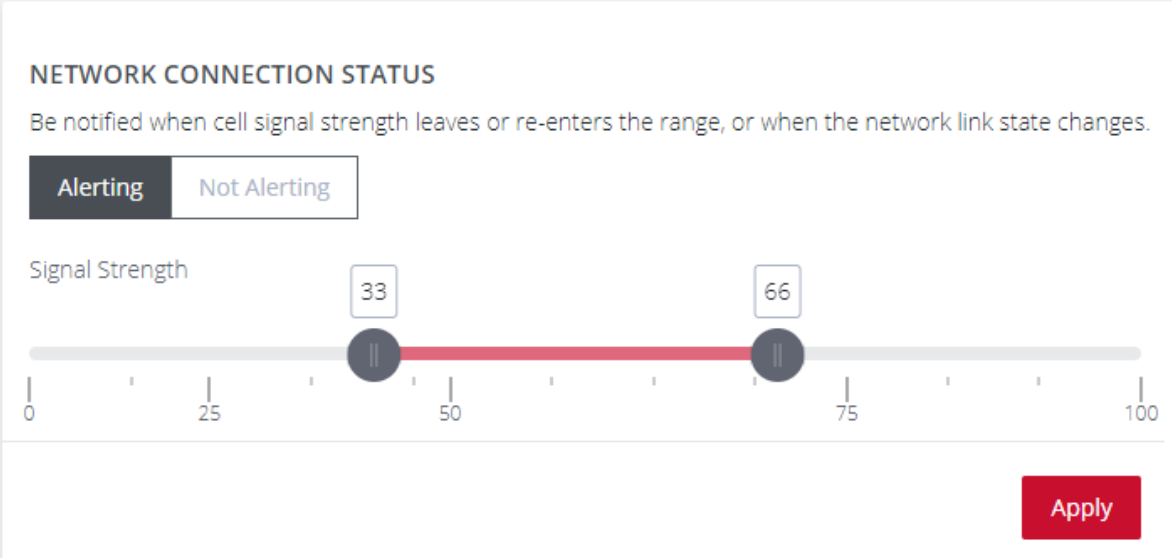
Configure > SNMP Alerts > Networking > Network Connection Status

To set the Network Connection Status signal strength boundaries:

1. Navigate to Configure > SNMP Alerts > Network Connection Status > Signal Strength page.
2. Click on the **Alerting** button to activate the function, this also activates the user-defined range sliders.
3. Click+Drag the signal strength range limiters to the required upper and lower limits.

> **Note:**The **Not Alerting** button de-activates the function and signal strength alerts will be stopped until activated again.

4. Click **Apply**. The **Details Saved** banner confirms your settings.



When an event occurs that causes the signal strength to re-enter the user-defined range, an SNMP alert will be triggered.

In the above image, if any anomaly occurs that causes the signal strength to drop below 33 or above 66, an SNMP alert will be triggered.

NETWORK CONNECTIONS

CONFIGURE > NETWORK CONNECTIONS

The **Network Connections** menu contains the **Network Interfaces**, **IPsec Tunnels** and **Static Routes** settings.

NETWORK INTERFACES

CONFIGURE > NETWORK CONNECTIONS > Network Interfaces

The interface supports both IPv4 and IPv6 networks. The IP address of the unit can be setup for Static or DHCP. The following settings can be configured for network ports:

- IPv4, IPv6
- Static and/or DHCP
- Enabling or disabling network interfaces
- Ethernet Media types

For detailed information about Network Interface configuration and adding a new connection, see "Change Network Settings" on page 41.

For information about VLAN interfaces, bridges and bonds, see "Network Aggregates - Bonds and Bridges" on page 166

DNS CONFIGURATION

DNS settings such as Name Servers and Search Domains can be configured for each network interface, which will override the DHCP provided settings.

Name servers allow the system to resolve hostnames to IP addresses to communicate with remote systems. Search domains allow the system to resolve partially qualified domain names (PQDN) by appending entries from the listed search domains to form a fully qualified domain name (FQDN).

When adding an interface to a Bond or Bridge, it will use the DNS configuration of the aggregate interface.

> **Note:** Interfaces must have at least one network connection to be able to perform DNS resolution.

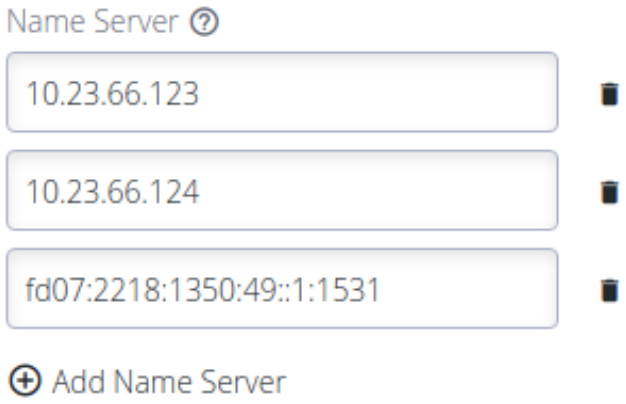
CONFIGURE DNS VIA THE WEB UI

CONFIGURE > NETWORK CONNECTIONS > Network Interfaces

On the Network Interfaces page, select the desired interface and click the Edit link.

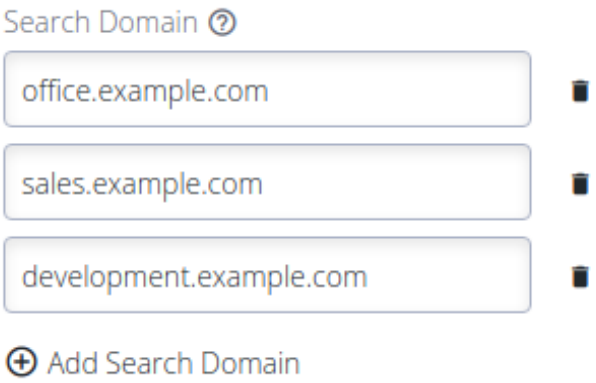
NAME SERVERS

1. Add one or more name servers to the list by clicking the **Add Name Server** button.
2. Name servers can be IPv4 or IPv6 addresses.
3. Name servers can be removed from the list by clicking the **Delete** button next to each row.
4. Click **Apply** to save the changes.



DNS SEARCH DOMAINS

1. Add one or more DNS search domains to the list by clicking the **Add Search Domain** button.
2. Search domains should be fully qualified domain names.
3. Search domains can be removed from the list by clicking the **Delete**



button next to each row.

4. Click **Apply** to save the changes.

CONFIGURE DNS VIA THE COMMAND LINE

Description	Command
Display configured DNS settings for an interface	`ogcli get physif "net1"`

Update DNS settings for an interface	`ogcli update physif "net1" << END`
	` dns.nameservers[0]="1.1.1.1"`
	` dns.nameservers[1]="1.0.0.1"`
	` dns.search_domains[0]="example.net"`
	` dns.search_domains[1]="example.com"`
	`END`

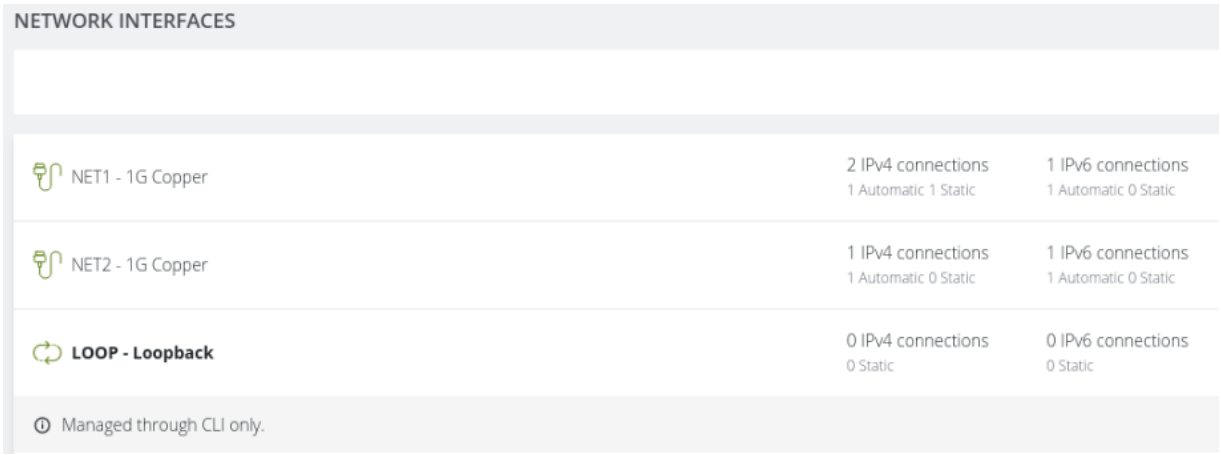
Check unbound service status	`systemctl status unbound.service`
List forward-zones in use	`unbound-control list_forwards`

LOOPBACK INTERFACE

Network Administrators deploying devices on a large scale network, including connections to multiple WAN routers, are able to utilize loopback interfaces (and other virtual interfaces not tied to physical connections) as source addresses for device management.

Opengear's Loopback Config CLI interface enables Administrators to create, advertise and reach loopback addresses through which to manage devices using a universal networking segment, ensuring consistent source addresses, and facilitating management services like SNMP, RADIUS, TACACS, Syslog, and SSH. The Opengear loopback interfaces with existing routing protocols (for example, OSPF, BGP) and provides specific configuration control via the Config CLI.

Note: Loopback interfaces are configured in the Config CLI and cannot be configured via the WebUI in the same way as other physical interfaces or aggregates. However, the status of existing loopback interfaces are shown on the Network Interfaces page of the WebUI. Static connections added to the loopback will also be displayed in the WebUI under the loopback interface.



Loopbacks are created using config shell and ogcli through the `physifs` endpoint. Created loopbacks can be viewed through the web UI under the Network Interfaces section. You can also use the cli command ip a to see a created loopack interface, if it has been enabled.

Provided that the connecting device has a route to the loopback, it will work with any management service like remote auth or ssh. For example, if you have a loopback address at 1.1.1.1, you can ssh into your device using the command: ssh root@1.1.1.1. A static route to the loopback must first be configured in order for this to work.

Up to 5 loopback interfaces can be created through the `physif` endpoint, with 5 connections attached to each interface. Service translations can be created through the `firewall/service_translation` endpoint to change the source address of outbound packets to the loopback address.

LOOPBACK CHARACTERISTICS

- Loopback interfaces support both /32 IP addressing for IPv4, and /128 IP addressing for IPv6.
- Multiple loopback interfaces can be created or deleted, along with their associated addresses. Addresses are individually editable or deletable.
- Services (e.g., SNMP, RADIUS, TACACS, Syslog, SSH) are reachable via the loopback address. Services must be configurable to use the IP loopback interface/address as the source address. Only tcp or udp packets leaving the device are service translated.
- Loopback interfaces can be discovered, you may need to configure dynamic route sharing settings to share directly connected routes. Opengear supports ospf routing protocols through config shell or ogcli. For this, you must set the

`redistribute_connected` option to **true**. For other dynamic routing protocols users must provide the configuration file.

- Loopback interfaces may be integrated into existing firewall configuration functionality.
- Loopback interface addresses may be pinged, provided the routes and firewall are configured correctly.
- Duplicate loopback IP configuration and duplicate IP with other network interfaces are disallowed.

See Create or Configure a Loopback Interface in the Config CLI Guide section of this document for information about creating, configuring or debugging a loopback interface.

DUAL SIM

CONFIGURE > NETWORK CONNECTIONS> Network Interfaces > WWAN0 - Cellular Interface

Console Manager has been available for some time with support for two SIM cards/slots, whereby, it is possible designate which SIM slot is the Active SIM that is normally used by the CM for OOB communications (in Automatic failover mode this SIM is termed the Primary SIM). The secondary SIM is used as a failover SIM. This feature increases the reliability of the OOB solution by providing redundant Out-Of-Band access over a cellular connection.

> **Note:**The terminology changes when SIM Failover policy is switched from **Manual** to **Automatic**. In Manual failover mode the active SIM is designated ACTIVE, whereas in Automatic failover mode the active SIM is designated PRIMARY.

With the Dual SIM feature activated, in the event of a failure of OOB communications through the Active SIM, it is possible to manually de-select the failed SIM and activate the secondary SIM by making *it* the Active SIM. This changeover allows OOB communications to resume through the newly designated Active SIM.

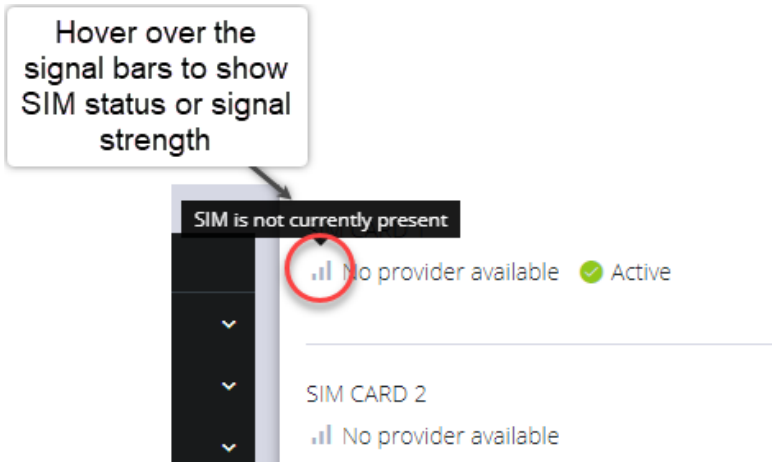
DISPLAY SIM STATUS AND SIGNAL STRENGTH

> **Note:**For information about configuring the **Signal Strength Thresholds** see: "System Alerts" on page 132

1. Navigate to Configure > Network Connections > Network Interfaces.
2. Click on the **WWAN0 - Cellular Interface** row.
3. .
   The information bar expands, and the page shows the current status of the active and inactive SIM cards.

   > **Note:**If the unit does not have a cell modem - then the cellular interface will not be visible.

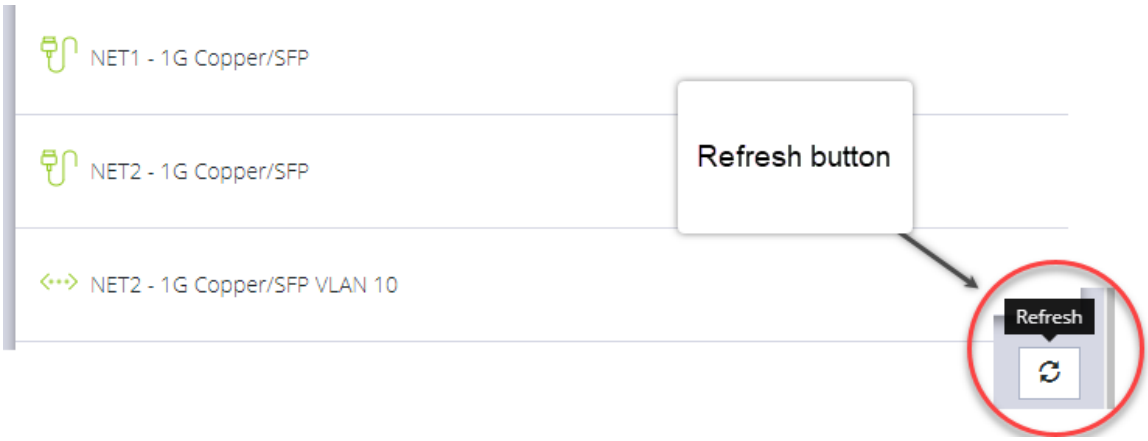
4. The active SIM indicates the color of the signal strength based upon the selected thresholds in **Configure → SNMP Alerts** under the **Networking Signal Strength Alert**.



The signal bar color (not the number of bars) indicates signal strength:

- **Green** if signal is above the higher threshold.
- **Amber** if signal is between lower and higher threshold.
- **Red** if signal is below the lower threshold,
- **Grey** for 0 or not active,

5. Click the **Refresh** button to display the current signal strength of the active SIM.



Note: When the **Refresh** button is clicked the signal strength is only updated for the active SIM. If you would like to know what the other SIM Signal Strength is, you need to activate it, let the modem come back online, which may take 3 minutes or more.

INSTALLING A NEW SIM CARD

When installing a new SIM card into its slot while the appliance is active (hot swapping), it is possible that it will take a minute or two for the system to react and stabilize after the SIM card change.

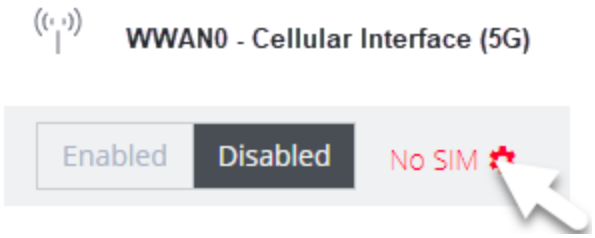
Two SIM card slots are located on the rear face of the device between the antennas, insert each SIM card(s) into its respective slot (marked 1 and 2) until you feel the card click into place.

Connect the RJ11 cable at the RJ11 port at the rear, you will hear or feel a slight click when it is correctly inserted. The modem is configured at the WebUI.

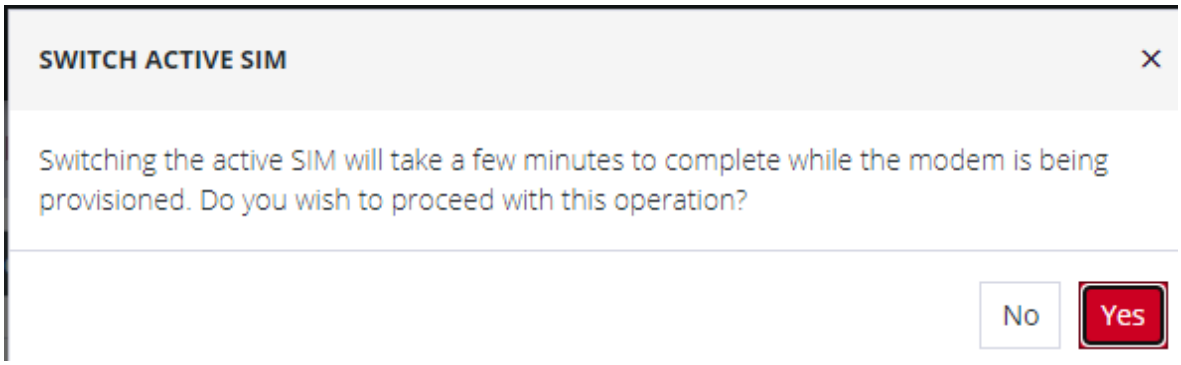
SELECT THE ACTIVE SIM (MANUAL FAILOVER MODE)

Switching the active SIM must be done manually. To switch the Active SIM:

1. Navigate to **CONFIGURE > NETWORK CONNECTIONS > Network Interfaces > WWAN0 - Cellular Interface** .
2. Click the **Settings cog** , this will display the **MANAGE WWAN0 - Cellular Interface** page and the current status of both SIM slots, including the current carrier name.



3. On the right, select the **Make Active** button of the new, active SIM and apply the change by selecting **Confirm**.
4. A pop-up alert states that this operation will take a few minutes to complete. Click **Yes** to confirm the change.



Note: During the change-over the current IP address is hidden and then returned when the modem re-connects.

5. If you require, you can monitor the interface during the changeover via the CLI with the command:.

   ```
   watch ip address show dev wwan0
   ```

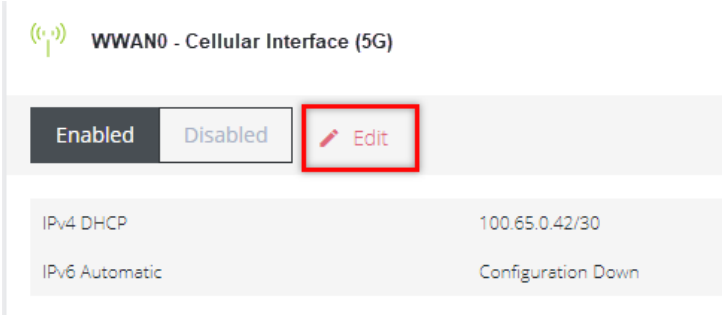
You can also set the SIM settings by expanding the menu for each SIM to set the APN.

If no SIM is inserted you can still select a SIM slot. If you insert a SIM it will not force it to become the active SIM.

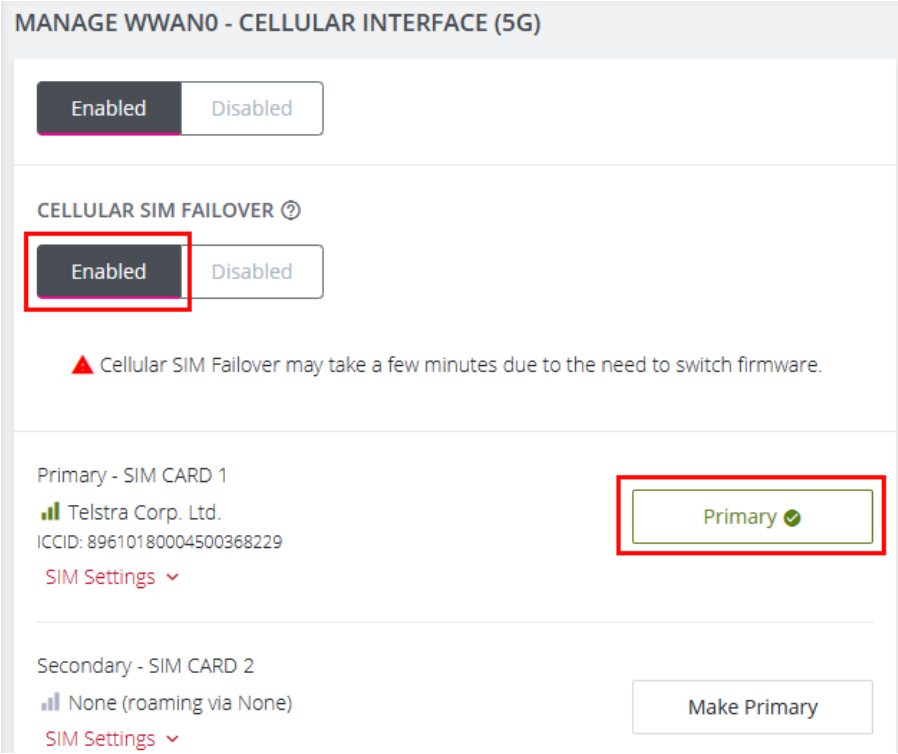
SELECT THE PRIMARY SIM (AUTOMATIC FAILOVER MODE)

Switching the primary SIM must be done manually. To switch the Primary SIM:

1. Navigate to **CONFIGURE > NETWORK CONNECTIONS > Network Interfaces > WWAN0 - Cellular Interface**.
2. Click the **Edit** icon, this will display the **MANAGE WWAN0 - Cellular Interface** page and the current status of both SIM slots.

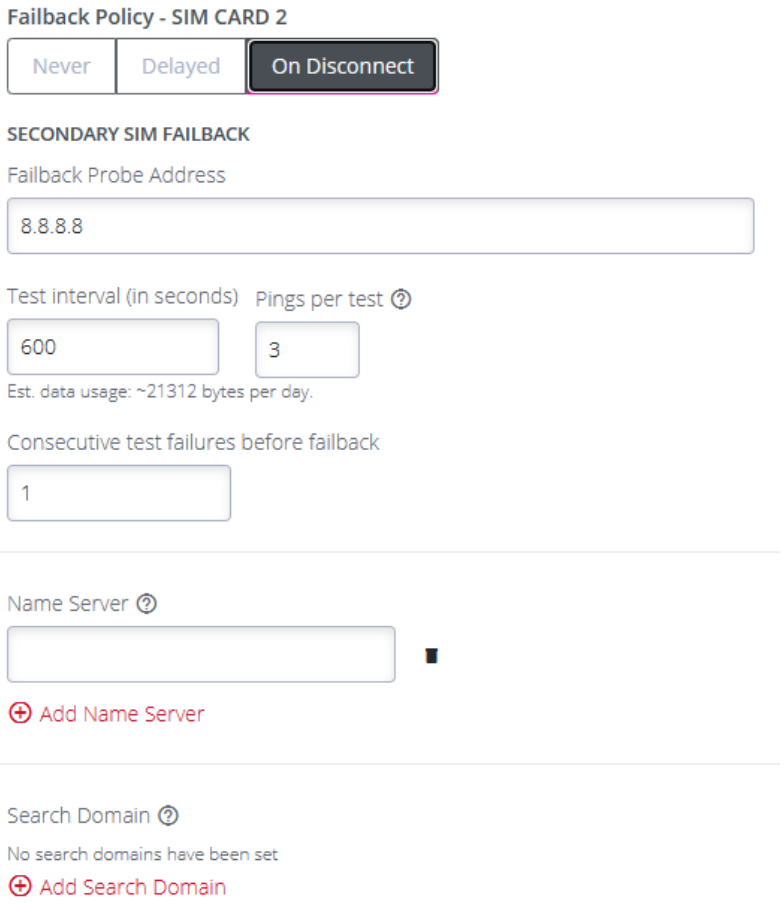


3. Ensure the cellular interface is enabled by clicking the **Enabled** button.



1. Click the **Primary** button of the SIM selected to be the primary SIM.

2. Select the required **Failback Policy** for the failback SIM and complete the failback policy details:



3. Click the **Confirm** button at the bottom of the page. A green banner will appear to confirm that the new settings have been saved.

DUAL SIM FAILOVER

CONFIGURE > NETWORK CONNECTIONS> Network Interfaces > WWAN0 - Cellular Interface > Edit

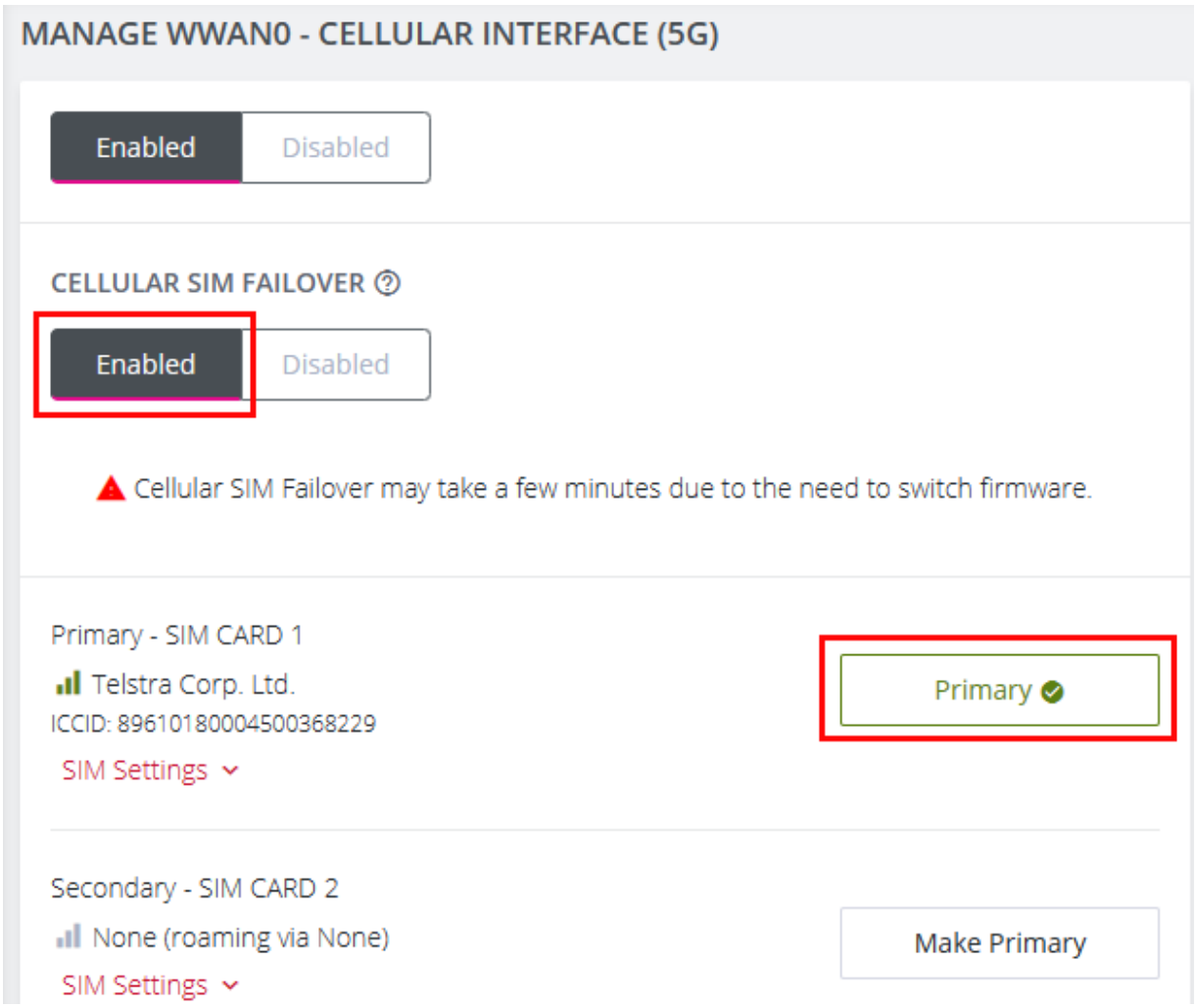
Console Managers that carry two SIM cards can be configured so that either SIM card slot may be activated. In failover mode, either of the two SIM cards may be designated as the Primary SIM. (see "Dual SIM" on page 143).

Dual SIM Failover works seamlessly with the existing failover solution to provide another layer of redundancy. This feature allows the software to detect a failure in OOB communications via the Primary SIM and will failover to the Secondary SIM without the need for manual operator intervention.

Options within the configuration also allow you to configure the failback settings from Secondary SIM, back to the previous Primary SIM when OOB communications have been restored. See "Cellular Interface Policy Settings" on page 156.

See the image on the following page for a depiction of Primary and Secondary SIM card slots.

Either of the SIM card slots can be designated as the Primary SIM. In the following image, SIM card 1 has been designated as the Primary SIM and is currently the active SIM, while SIM card 2 is designated as the Secondary SIM which, (in the scenario below), is activated in the event of a failover such as occurs during an OOB communications failure on the Primary SIM.



FAILOVER MODES

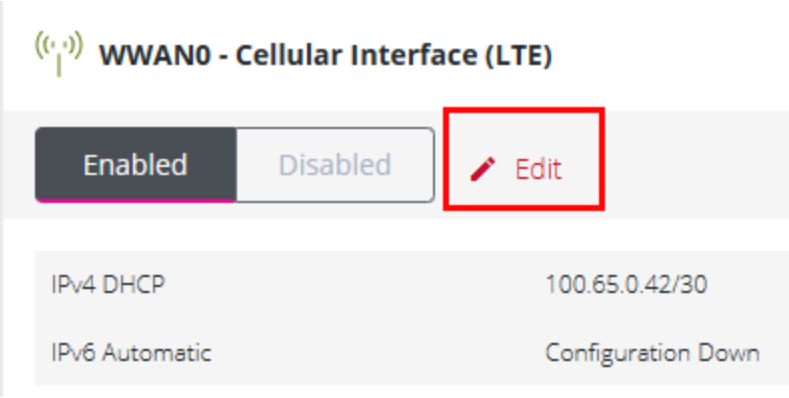
Features of Failover include:

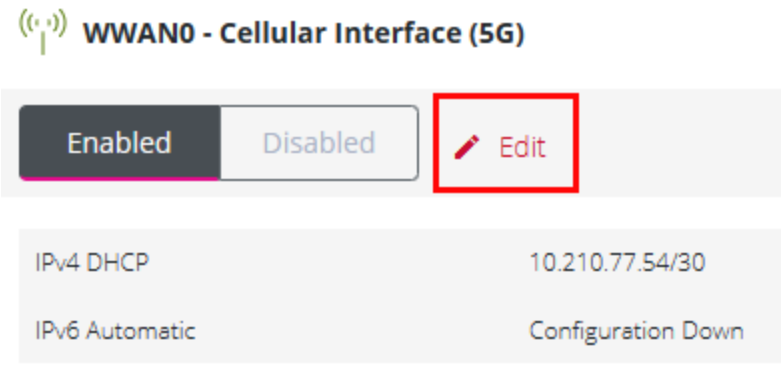
- Select **Enabled** SIM failover.
- Specify SIM failback policy (applicable when the Ethernet connection and primary SIM are both down):
  - **Never** - The node never switches back to the Primary.
  - **Delayed** (specified in minutes) - The node switches back to primary after a pre-defined time has elapsed.
  - **On disconnect** - See the table "Cellular Interface Policy Settings" on page 156 for an explanation of the policy.
- SIM failover settings allow you to configure the parameters that affect cellular data usage, for example, quicker failover (consumes more data) vs less frequent tests (consumes less data). The configuration preferences include
  - Ping test for failover from Primary to Secondary and failback from Secondary to Primary.
  - Failover settings are per SIM slot and consist of a failover and failback ping test.

ACTIVATE OR CONFIGURE FAILOVER

CONFIGURE > NETWORK CONNECTIONS> Network Interfaces > WWAN0 - Cellular Interface > Edit

1. Navigate to the Cellular Interface page at: CONFIGURE > NETWORK CONNECTIONS> Network Interfaces > Cellular Interface (LTE).
2. Click the **Edit** link next to the Cellular Interface Enabled/Disabled switch.





3. Select the **Enabled**failover option.
4. Ensure the correct SIM card is selected as the Primary SIM (see 'Set Primary SIM' in "Dual SIM" on page 143).
5. Complete the Cellular Interface options in accordance with the table below.

6. Click **Confirm** to activate the failover policy settings, a green banner will confirm the settings are enabled.

CELLULAR INTERFACE POLICY SETTINGS

MANAGE WWAN0 CELLULAR INTERFACE Properties	
Field	**Definition**
CELLULAR SIM FAILOVER - **Enabled.**	Switch between the Primary SIM Card and the secondary SIM Card on disconnection.
Primary SIM Failover	
Failover Probe Address.	Network address to probe in order to determine if connection is active. **Note:** The probe address accepts IPv4, IPv6 addresses and hostnames.
Test interval (seconds).	The number of seconds between connectivity probe tests.
Pings per test.	The maximum number of times a single ping packet is sent per probe before considering the probe failed.
Consecutive test failures before failover.	The number of times a probe must fail before the connection is considered failed.
Failback Policy	
Never / Delayed / On Disconnect.	Select the policy to be used to determine Failback recovery from the Secondary SIM Card back to the Primary SIM Card.

Never	No Failback recovery is attempted.
Delayed	Attempted failback after *n* minutes. The number of minutes after failover to the secondary SIM Card that the connection should failback to the Primary SIM Card.
On Disconnect	Secondary SIM Failback.
	Failback Probe Address ie. The Network address to probe in order to determine if the connection is active.
	Test Interval The number of seconds between connectivity probe tests (this not the same thing as Attempted Failback).
	Pings per Test The maximum number of times a single ping packet is sent per probe before considering the probe failed.
	Consecutive Test Failures (before failover) The number of times a probe must fail before the connection is considered failed.

CELLULAR MODEM FIRMWARE UPGRADE

This Cellular Modem Firmware Upgrade procedure provides an automatic download and upgrade process for carriers, and, a secondary manual upgrade process for users who must use a firmware set that has not been tested by Opengear or use a carrier that is not supported by the standard cellular modem firmware.

Opengear devices use a standard modem, however, due to the variety of carriers that exist, there is a wide variety of firmware packages which are offered by Sierra Wireless (Opengear's modem provider) in order to accommodate these different carriers. When Opengear devices are supplied, they are provided with the most common set of modem firmware pre-installed; this minimizes difficulty when setting up cellular services on devices. The manual cellular upgrade procedure supports users deploying cellular capable devices to regions that use a carrier that is not supported by the standard cellular modem firmware.

Note: The Cellular Firmware Upgrade procedure is only available through terminal or shell access. The use of automated tools such as cron jobs is not supported and is therefore discouraged.

MODEM FIRMWARE UPGRADE PROCEDURES

CELLULAR AVAILABILITY DURING UPGRADE

The `cell-fw-update` command will disable the cellular modem during the upgrade process. This will cause a loss of availability of the Out-of-Band (OOB) link which can only be restored once the cellular modem has returned to a working state. The 'defer if failed over' feature provides some protection.

CELL-FW-UPDATE HELP

```
root@om2248-l-tp1-p14:~# cell-fw-update --help
Usage: /usr/bin/cell-fw-update [options] <actions>

Actions:
    -m <file> [-m <file>].. Flash modem with firmware <file>(s)
    -c <carrier>        Flash modem with firmware suitable for <carrier>
    -l              List carrier IDs suitable for use with -c
    -f              Write current fingerprint and timestamp to stdout
    -u              Update file lists from remote server
    -d              Download/synchronize fw files from remote server
    -h              Show this usage
Options:
    -a              Report automated upgrade messages
    -b <url>            Specify base URL to remote
    -v              Verbose messages
    -C              Continue/resume partial downloads
    --unsafe            Ignore all checksums/signatures and allow downgrades.
                    This enables existing firmware to be re-flashed when using the qmi-firmware-
update back-end
    --defer             deprecated! Do not permit firmware upgrade if system is currently failed-over.
                    This is now default behaviour. Use the flag --ignore-defer to bypass this.
    --libqmi            Force use of libqmi tool qmi-firmware-update. Cannot use with --mbpl
    --mbpl              Force use of Sierra Wireless MBPL fwdwl-lite. Cannot use with --libqmi
    --ignore-defer      Bypass the 'failover defer' check to force a modem firmware upgrade
```

UPDATE LOCAL FILE LIST AND DOWNLOAD LATEST FIRMWARE FILES

This procedure will update the local file list and download the latest firmware files.

Note: `cell-fw-update` can be run directly from a CLI shell as root and requires no configuration. You can combine this update action with the following download operation by providing both -u and -d simultaneously.

```
root@om8148-10g-tp2-p35:~# cell-fw-update -ud
Waiting for clients to stop using the modem...
The modem is now locked

=== INFO ===
The modem is locked by client cellfw
No clients want to use the modem
UIM failover status is disabled
Active UIM slot is 1 (ICCID: 89610180003137049629)
Operator is telstra corp. ltd.
0157863e6fe95988415b264e35ac0b4f687ffbf9 2024-01-18
download e4c83bb1ae1e5be73c3a254fca7e13e38b33e39a SWIX65C_
02.13.08.00.cwe
download 31dca80c90d37100b17ac8e49998ce35724c6b90 SWIX65C_
02.13.08.00_GENERIC_030.047_001.nvu
download 5ed78eb2d69d651d73e177c855eaecb02c6df0b0 SWIX65C_
02.13.08.00_PTCRB_030.045_001.nvu
download 91b8c518ddfad508ffe22c0f099465abb8b31d88 carrier-canon.txt
download b8d3a9cb4faabcf6f5e1fa5acb0f4e41ed72f506 carriers.txt
copy a6ddf97fb6b6f8dd0d011d54dcdfe34db64b25ee cell-firmware.txt
```

```
copy - localfiles.txt
copy - localdb.txt
copy - SHA1SUMS
```

Note: The `cell-fw-update -u` and `cell-fw-update -d` commands may be run separately.

LIST SUPPORTED CARRIERS

The resulting carriers shown below are for example only (local results may vary).

```
root@om2216-l:~# /etc/scripts/cell-fw-update -l
att AT&T
docomo DoCoMo
generic Generic
kddi KDDI
kt Korea Telecom
rogers Rogers
softbank SoftBank
sprint Sprint
telstra Telstra
telus Telus
tmo T-Mobile
uscellular U.S. Cellular
verizon Verizon Wireless
```

AUTOMATIC FIRMWARE UPDATE FOR CURRENT CARRIER

This procedure detects the currently connected carrier and updates the firmware set for that specific carrier. A firmware set consists of the modem's firmware image (.cwe) and a carrier specific PRI firmware image (.nvu). This set is required for modem operation.

```
cell-fw-update -a
```

FIRMWARE UPDATE FOR SPECIFIC CARRIER

Specify which carrier you for which you want to update the firmware.

```
cell-fw-update -c <carrier>
```

Note: Use the `cell-fw-update -l` command to list supported carriers.

MANUAL FIRMWARE UPDATE

Specify a firmware set to download to the modem. This allows you to update the modem with a specific firmware set instead of one provided by Opengear FTP. The path to the firmware set specified must be relative from the directory /mnt/nvram/cellfw/.

Warning: This operation must be used with great caution as can result in the modem becoming *permanently* unavailable or damaged. Use at your own risk.

```
root@om8148-10g-tp2-p35:~# cell-fw-update --unsafe -m SWIX65C_
02.13.08.00.cwe -m SWIX65C_02.13.08.00_GENERIC_030.047_001.nvu
Waiting for clients to stop using the modem...
The modem is now locked
=== INFO ===
The modem is locked by client cellfw
No clients want to use the modem
UIM failover status is disabled
Active UIM slot is 1 (ICCID: 89610180003137049629)
Operator is telstra corp. ltd.
Application version: 1.0.2307.1
Target image Info:
Carrier :GENERIC
FW Version :02.13.08.00
Model ID :SWIX65C
Package ID :001
PRI Version:030.047
SKU :9999999
Switching device into download mode ...
Modem Needs FW
Modem Needs PRI
Downloading: /tmp/cell-fw-update.4045/SWIX65C_02.13.08.00.cwe
Downloading: /tmp/cell-fw-update.4045/SWIX65C_02.13.08.00_GENERIC_
030.047_001.nvu
All image data was downloaded successfully.
Device is about to reset ...
Waiting for modem to come up in ONLINE mode ...
Modem is now in ONLINE mode ...
FW update status: Successful
```

```
FW info from modem:
Model ID : EM7565
FW Version : SWIX65C_02.13.08.00
Carrier Name : GENERIC
Carrier PRI Revision: 030.047_001
Firmware download process completed successfully.
INFO: QDL Port: /dev/wwan0qdl0
INFO: Device Path: /dev/wwan0qmi0
INFO: FW Path: /tmp/cell-fw-update.4045
Waiting for modem to disconnect from the host ...
Modem disconnected from host.
Waiting for modem to come up in BOOT and HOLD mode ...
BOOT and HOLD Mode. Downloading firmware ...
[/dev/wwan0qmi0] Device list of stored images retrieved:
[/dev/wwan0qmi0] Device list of stored images retrieved:
<14>Jan 22 06:05:25 cell-fw-update: The firmware was successfully
stored on the modem
[/dev/wwan0qmi0] Device list of stored images retrieved:
```

MODEM UPDATE TROUBLESHOOTING GUIDE

The following procedure can be used to determine if the cellular modem is ready and available and may provide recovery if necessary if the upgrade or modem repeatedly fails.

DETERMINE IF MODEM IS READY & AVAILABLE

The service `ModemManager` is an essential dependency for all cellular modem operations. Please ensure it is running.

```
root@om8196-10g:~# systemctl start ModemManager
```

If the modem is running correctly, it should be able to be detected by ModemManager within 60 seconds of the service starting.

```
root@om8196-10g:~# mmcli -L
```

If the modem was not detected or is still problematic, the modem needs to be recovered.

DETERMINE IF THE MODEM IS CURRENTLY BEING UPGRADED

The simplest way to determine if the modem is currently being upgraded is to check the currently running processes an look for `cell-fw-update`. This is done through the following check:

```
ps aux | grep cell-fw
```

The following example shows that an upgrade is running:

```
root@om2216-l:~# ps aux | grep cell
root 122965 0.2 0.0 4780 3992 pts/0 S+ 23:42 0:00 /bin/bash
/usr/bin/cell-fw-update -aud
root 125966 0.0 0.0 3332 1756 pts/1 S+ 23:47 0:00 grep cell
```

The following example shows that there is no upgrade running:

```
root@om2216-l:~# ps aux | grep cell-fw
root 126417 0.0 0.0 3332 1776 pts/1 S+ 23:48 0:00 grep cell-fw
```

NETWORK AGGREGATES - BONDS AND BRIDGES

CONFIGURE > NETWORK CONNECTIONS > Network Interfaces > Select the target interface

BRIDGES

Network bridges allow connecting of multiple network segments together so that they may communicate as a single network.

Definitions of the bridge details as in the **Bridge Form Definitions** table later in this topic.

> **Note:** Whether creating a new bridge or editing an existing bridge the page is very similar.

CREATE A NEW BRIDGE

To create a new bridge:

1. Navigate to the **Configure > Network Connections > Network Interfaces** page on the Web GUI.
2. Click on the **New Bridge** button that is located at the top-right of the window.
3. Select which interface will serve as the primary interface for the new bridge.

   Note: When the primary interface is selected, its MAC address is displayed in the MAC address field. This MAC address is inherited by the new bridge interface.

4. Complete the new bridge details form as in the **Bridge Form Definitions** table.
5. Click the **Create** button to finalize the creation of the new bridge.

EDIT AN EXISTING BRIDGE

To edit an existing bridge:

1. Navigate to the **Configure > Network Connections > Network Interfaces** page on the Web GUI.
2. Click on the bridge that you would like to edit, the bridge details are expanded.
3. Click on the bridge **Edit** button that is located next to the Enable / Disable toggle buttons.
4. Select which interface will serve as the primary interface for the new bridge.
5. Change the bridge details as required in accordance with the **Bridge Form Definitions** table.
6. Click the **Update** button to finalize the edit process. Updating the bridge will temporarily interrupt network activity on this interface.

Note: Editing the primary interface will not update its connections.

EDIT BRIDGE - FORM DEFINITIONS

New Bridge Field	Definition
Description	The editable **Description** field allows you to add a description of the interface. If the description field is not completed the field will default to a computed value to describe the interface.
Enable Spanning Tree Protocol	Enable or disable Spanning Tree Protocol. See "Spanning Tree Protocol" on page 175.
Network Interface Selection	Click the checkbox of each network interface you want to include in the bridge. Available interfaces include Ethernet and VLAN interfaces that are not part of another bond or bridge. Bond interfaces can be included in a bridge by using the **ogcli** tool. See Support for Bonds in Bridges in the Knowledge Base.
Primary Interface	Select the interface that is to be used for selecting the MAC address of the aggregate. The new bond inherits the MAC address of the primary interface. On creation, any Network Connections which exist on the Primary Interface will be attached to the Bond/Bridge after it is initially created. When a Bond/Bridge is deleted, any Network Connections which exist on the aggregate interface are handed over to the Primary Interface.
Inherited Connections	When the Primary Interface is selected, the connections inherited by the new bridge are listed here.

Edit	Click to edit the details of an existing interface.

BONDS

Network bonds allow combining two or more network interfaces together into a single logical "bonded" interface for load balancing, redundancy or improved performance depending on the bond mode used.

Definitions of the bond details as in the **Bond Form Definitions** table later in this topic.

> **Note:** Whether creating a new bond or editing an existing bond the page is very similar.

CREATE A NEW BOND

To create a new bond:

1. Navigate to the **Configure > Network Connections > Network Interfaces** page on the Web GUI.
2. Click on the **New Bond** button that is located at the top-right of the window.
3. Select which interface will serve as the primary interface for the new bond.

   > **Note:** When the primary interface is selected, its MAC address is displayed in the MAC address field. This MAC address is inherited by the new bond interface.

4. Complete the new bond details form as in the **Bond Form Definitions** table.

5. Click the **Create** button to finalize the creation of the new bond. Network connections from non-primary interfaces will be deleted when the new bond is created.

EDIT AN EXISTING BOND

To edit an existing bond:

1. Navigate to the **Configure > Network Connections > Network Interfaces** page on the Web GUI.
2. Click on the bond that you would like to edit, the bond details are expanded.
3. Click on the bond **Edit** button that is located next to the Enable / Disable toggle buttons.
4. Change the bond details as required in accordance with the **Edit Bond Form Definitions** table below.
5. Click the **Update** button to finalize the edit process. Updating the bond will temporarily interrupt network activity on this interface.

> **Note:** Editing the primary interface will not update its connections.

EDIT BOND - FORM DEFINITIONS

New Bond Field	Definition
Description	The editable **Description** field allows you to add a description of the interface. If the description field is not completed the field will default to a computed value to describe the interface.
Mode	The mode determines the way in which traffic sent out via the bonded interface is dispersed over the real interfaces. Available modes are:
	Round Robin Balancing - Packets are sequentially transmitted/received through each interface, one by one.
	Active Backup - If the active secondary interface is changed during a failover, the bond interface's MAC address is then changed to match the new active secondary's MAC address.
	XOR Balancing - Balances traffic by splitting up outgoing packets between the Ethernet interfaces, using the same one for each specific destination when possible.
	Broadcast - All network transmissions are sent on all secondary interfaces. This mode provides fault tolerance.
	802.3ad (Dynamic Link Aggregation) - Aggregated NICs act as one NIC, but also provides failover in the case that a NIC fails. Dynamic Link Aggregation requires a switch that supports IEEE 802.3ad.
	Transmit Load Balancing - Outgoing traffic is distributed

	depending on the current load on each secondary interface. Incoming traffic is received by the current secondary interface. If the receiving secondary fails, another secondary takes over the MAC address of the failed secondary.
	Adaptive Load Balancing - Includes transmit load balancing (tlb) and receive load balancing (rlb) for IPv4 traffic and does not require any special switch support.
Poll Interval	The poll interval specifies the MII link monitoring frequency in milliseconds. This determines how often the link state of each secondary is inspected for link failures. A value of zero will disable MII link monitoring.
Network Interface Selection	Click the checkbox of each network interface you want to include in the bridge. Available interfaces include Ethernet and VLAN interfaces that are not part of another bond or bridge.
Primary Interface	Select the interface that is to be used for selecting the MAC address of the aggregate. The new bond inherits the MAC address of the primary interface. On creation, any Network Connections which exist on the Primary Interface will be attached to the Bond/Bridge after it is initially created. When a Bond/Bridge is deleted, any Network Connections which exist on the aggregate interface are handed over to the Primary Interface.
Active Connections	When the Primary Interface is created, the connections inherited by the new bond are listed here. When edited, Active Connections on the aggregate will not be updated if the primary interface is changed.

Edit	Click to edit the details of an existing interface. Updating a bridge will temporarily interrupt network activity on the interface when you click the **Update** button.

SPANNING TREE PROTOCOL

CONFIGURE > NETWORK CONNECTIONS > Network Interfaces > Select the target interface

Spanning Tree Protocol (STP) allows an Console Manager to discover and eliminate loops in network bridge links, preventing broadcast radiation and allowing redundancy.

When STP is implemented on switches to monitor the network topology, every link between switches, and in particular redundant links, are cataloged. The spanning-tree algorithm blocks forwarding on redundant links by setting up one preferred link between switches in the LAN. This preferred link is used for all Ethernet frames unless it fails, in which case a non-preferred redundant link is enabled.

Note:STP Limitations

If multiple bridges are created on the same switch, they should not be used on the same network segment as they have the same MAC addresses; therefore, STP will likely not work correctly as they will have the same bridge id.

Rapid Spanning Tree Protocol (RSTP), Multiple Spanning Tree Protocol (MSTP) and other proprietary protocols are not supported.

The bridge settings relating to STP cannot be changed from the default values shown below:

group_address

forward_delay (default is 15)

hello_time (default is 2)

max_age (default is 20)

priority (default is 32768 (0x8000))

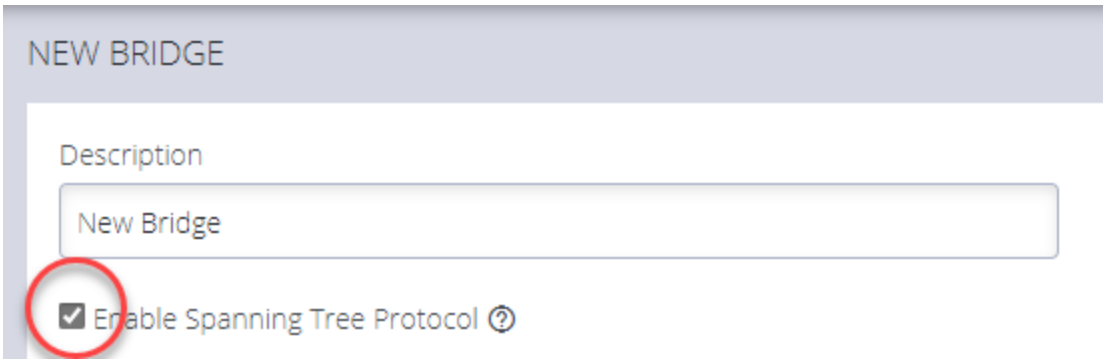
ENABLE STP IN A BRIDGE

To enable STP you can use the UI or CLI. The procedures are:

BRIDGE WITH STP ENABLED - UI

CONFIGURE > NETWORK CONNECTIONS > Network Interfaces > Select the target interface > New Bridge page

1. In the **Network Interfaces** page, click the **Create New Bridge** button.
2. Click to select the **Enable Spanning Tree Protocol** option.



BRIDGE WITH STP ENABLED - OGCLI

```
admin@cm8148:~# ogcli get physif system_net_physifs-5
  bridge_setting.id="system_net_physifs-5"
 bridge_setting.stp_enabled=true
 description="Bridge"
 device="br0"
 enabled=true
 id="system_net_physifs-5"
 media="bridge"
 name="init_br0"
  slaves[0]="net2.3"
```

BRIDGE WITH STP DISABLED - OGCLI

```
admin@cm8148:~# ogcli update physif system_net_physifs-5 bridge_
setting.stp_enabled=false
  bridge_setting.id="system_net_physifs-5"
  bridge_setting.stp_enabled=false
 description="Bridge"
 device="br0"
  enabled=true
 id="system_net_physifs-5"
 media="bridge"
 name="init_br0"
  slaves[0]="net2.3"
```

IPSEC TUNNELS

CONFIGURE > NETWORK CONNECTIONS > IPsec Tunnels

The Opengear Console Manager (CM) can use IPsec to securely connect and route between two or more LANs (sometimes referred to as site to site, LAN-to-LAN, L2L VPN), or as a single client endpoint connecting to a central LAN or endpoint (sometimes referred to as host to site, or host to host).

IPsec does not make a formal distinction between initiator and responder, however the Opengear CM can both initiate tunnels (as the "initiator") and have other devices initiate tunnels to it (as a "responder").

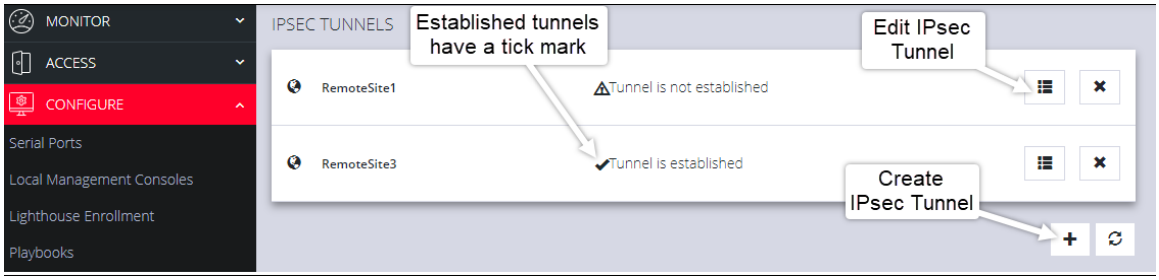
CREATE, ADD OR EDIT IPSEC TUNNELS

On the IPsec Tunnels page, you can create, edit, and delete IPsec tunnels.

To create an IPsec tunnel:

1. Click **CONFIGURE > NETWORK CONNECTIONS > IPsec Tunnels.**

   The IPsec Tunnels page with two tunnels previously created.

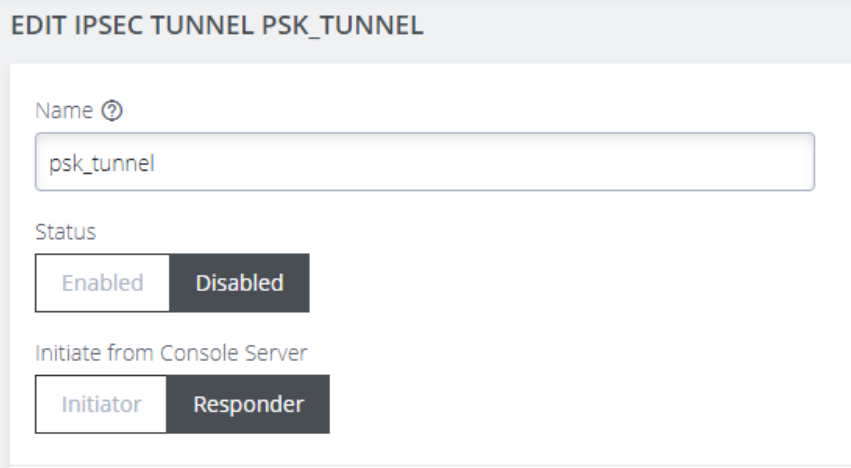


   *If there are no existing tunnels, this **Create Tunnel** button is displayed:*

2. Click **CREATE TUNNEL**. This opens the **EDIT IPSEC TUNNEL** page.

NAME and STATUS

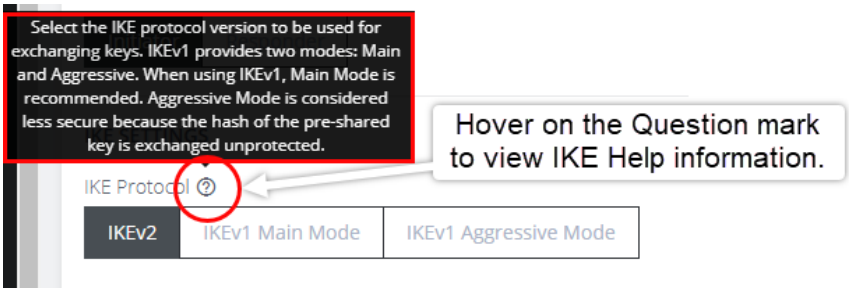
3. In the **Name** section of the page, give your new tunnel a unique name and click the **Enabled** button.



4. Set the Console Server to be the **Initiator** or **Responder**.

> **Note:**When **Initiator**is selected, the node will actively initiate the tunnel by sending IKE negotiation packets to the remote end.

IKE SETTINGS



Continued...

5. Select an **IKE Protocol** version to use for exchanging keys. IKEv1 provides two modes: **Main** and **Aggressive**. When using IKEv1, Main Mode is recommended. Aggressive Mode is considered less secure because the hash of the pre-shared key is exchanged unprotected.
6. Select the **Algorithm Proposal**. This is a set of algorithms used for negotiation when attempting to establish the IPsec tunnel. By default, the node will attempt to negotiate the tunnel using a list of common algorithms which are considered safe. Alternatively, a set of default proposals that guarantee Perfect Forward Secrecy (PFS) can be selected.
7. Select **Initiate** to actively initiate the tunnel by sending IKE negotiation packets to the remote end.
8. Set up the **Phase 1** and **Phase 2** time interval between the key material refresh of the IKE and Child.

AUTHENTICATION

CM Authentication can use PSK or PKI.

9. **For pre-shared key (PSK) authentication**, enter a pre-shared secret key; both ends of the tunnel must use the same key.

   Tip:
   To construct ID_USER_FQDN identities, use `user@example.com`
   To construct ID_FQDN type identities, use `@host.example.com`
   If left blank, the outer local IP address of the tunnel is used as the identity.

10. Enter a **Local ID** Identity or IP address for the local end of the tunnel. If left blank, the outer-local IP address is used as the source address of the tunnel.
11. **For Public Key Infrastructure (PKI) authentication**, upload the certification bundle file or, drag and drop the file into the Certificate Bundle field.

TUNNEL SETTINGS

12. Select **Enabled** if enforced UDP encapsulation is required. When enabled, the IKE daemon can simulate the NAT detection payload.

ADDRESSING

13. Enter the **Local Address** to be used as the source address of the tunnel. If left blank, IPsec will automatically use a default.
14. Enter a **Local Subnet**. Specify local traffic to be tunneled. When no subnets are specified, only traffic originating from this device will be tunneled.
15. Enter the **Remote Address** or hostname for the remote end of the tunnel. If left blank, IPsec will accept initiation packets from any address.
16. Enter the **Remote Subnet**. Specify addresses or subnets that are behind the remote end of this tunnel. If no subnet is specified, only traffic originating from the outer remote address will be accepted.

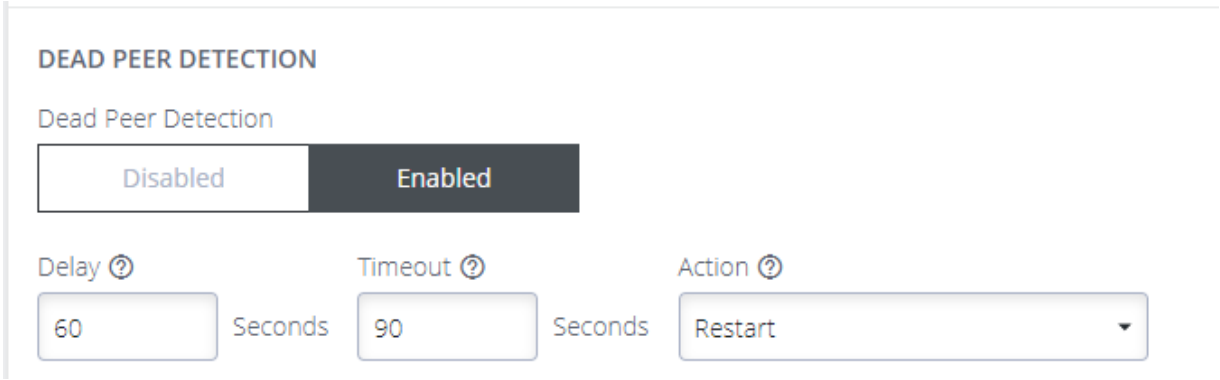
DEAD PEER DETECTION

> **Tip:** Dead Peer Detection may be used to support long-lived tunnels.

Dead Peer Detection (DPD) is a method used by nodes to verify the current existence and availability of IPsec peers. A node performs this verification by sending encrypted IKE Phase 1 notification payloads (R-U-THERE messages) to a peer and waiting for DPD acknowledgments (R-U-THERE-ACK messages) from the peer.

Continued...

You can enable DPD and configure the various options to fine-tune the functionality:



- **Delay** - the time interval between polling the peer (default is 60 seconds).
- **Timeout** - the waiting time before deciding that a peer connection is not live (default is 90 seconds).
- **Action** - the action to be performed when a connection is timed-out. (default is Restart).
  - **Restart** will immediately attempt to renegotiate the tunnel.
  - **Clear** will close the CHILD_SA.
  - **Trap** will catch matching traffic.

ENABLE the IPsec TUNNEL

17. When you have completed the IPsec Tunnel set-up process, ensure the IPsec tunnel status is set to **Enabled**, then, click **Save**.

    The new tunnel is now listed on the **CONFIGURE > NETWORK CONNECTIONS > IPsec Tunnels** page.

STATIC ROUTES

CONFIGURE > NETWORK CONNECTIONS > Static Routes

Static routes are predefined paths that traffic can be configured to take through the network for purposes such as security, cost or to override the default route.

The list of configured static routes are displayed in a table with their current status indicated by the status column.

Status	Meaning
Installed	The route is installed in the routing table.
Not Installed	The route may not be currently installed, but should update in a moment.
Error	The route failed to be installed.
Failed to fetch status	There is an error with the system and status failed to be obtained. This is a temporary error and should update in a moment.
The network interface is disabled	The static route is bound to an interface which is not enabled.
The network interface is disconnected	The static route is bound to an interface which is not connected.
The network	The route cannot be installed as there are no active connections on this interface.

Status	Meaning
interface has no active connections	

CONFIGURE STATIC ROUTES

On the Static Routes page you can add, edit or delete static routes.

Note: Only basic validation is performed when static routes are saved. Check the status column to ensure your route is installed and working correctly.

CREATE A STATIC ROUTE

1. Click the **Add** button to navigate to the creation page.
2. Enter a valid IPv4 or IPv6 destination address or network, followed by the netmask in CIDR notation. The destination address/network must be unique.
3. Enter the gateway or select an interface for the static route to use.
4. Optionally, provide a metric for the route. Routes with a lower metric value are higher priority.

Destination Address	Default Metric
IPv4	0

Destination Address	Default Metric
IPv6	1024

5. Click the **Apply** button to save the changes.
6. If the changes are saved successfully you are returned to the Static Routes list page.
   - If there is an error with the configuration and the route fails to install, a red banner is displayed.
   - If the route installed successfully, a green success banner is displayed.
7. The current status of the configured route is displayed in the table, which may change depending on the status of the network configuration.

EDIT A STATIC ROUTE

1. Click the description of the desired static route in the list to access the **Edit** page.
2. Update the details of the static route.
3. Click apply to save the changes.

DELETE A STATIC ROUTE

1. Click the description of the desired static route in the list to access the **Edit** page.
2. Click the **Delete** button at the top-right of the page.
3. Click **Yes** to confirm the action.

4. If the route was removed from the routing table as expected, a green success banner is displayed.

MANAGING STATIC ROUTES VIA COMMAND LINE

Administrative users can also view the status and perform configuration of static routes via the command line interface.

After creating or modifying a route via the command line, you should take note of the route id and confirm that it has been installed successfully in the routing table.

Description	Command
Display IPv4 installed routes	`ip route`
Display IPv6 installed routes	`ip -6 route`
Display all route information	`ip route show table all`
Show status of configured routes via ogcli	`ogcli get monitor/static_routes/status`
Get static route configuration via	`ogcli get static_routes`

Description	Command
ogcli	

Create static route via ogcli	`ogcli create static_route << END`
	` destination_address="1.1.1.1"`
	` destination_netmask=32`
	` gateway_address="1.1.1.1"`
	` interface="net1"`
	` metric=0`
	`END`

Update static route via ogcli	`ogcli update static_route "1.1.1.1" << END`
	` interface="net2"`
	` metric=100`
	`END`

Delete static route via ogcli	`ogcli delete static_route "1.1.1.1"`

NETWORK RESILIENCE

CONFIGURE > NETWORK RESILIENCE >

Under the NETWORK RESILIENCE menu, you can manage Out-of-Band (OOB) settings.

OUT OF BAND FAILOVER

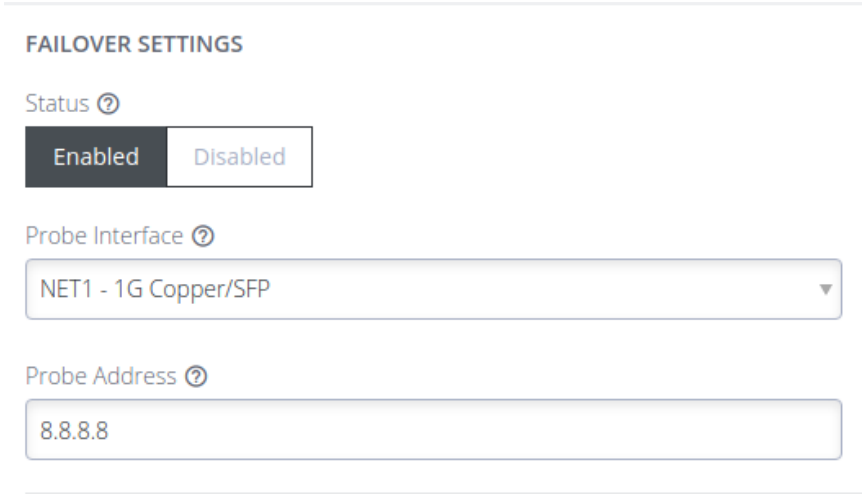
CONFIGURE > NETWORK RESILIENCE > OOB Failover

Out-of-Band (OOB) Failover detects network disruption via the probe interface, and automatically activates a cellular or ethernet interface connection to re-establish network access.

OOB failover requires an IPv4 address (in dotted decimal format), or an IPv6 address, or a domain name, which is always reachable and unlikely to change. When OOB failover is enabled, the node regularly pings this address, using the probe interface, to check for network connectivity.

ENABLE OUT-OF-BAND FAILOVER

1. To manage Out-of-Band Failover, navigate to the **CONFIGURE > NETWORK RESILIENCE > OOB Failover** page



Probe Interface: this is the interface that will be used to test if ping can reach the configured address

Probe Address: the ipv4 or ipv6 or domain name of the address that will be “pinged”

2. In the **Failover Interface** section, select the failover interface from the drop-down list.

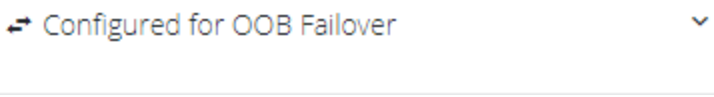
Configurable probe (failover from) and failover (failover to) interfaces are shown below:

NET1 - the default probe interface.

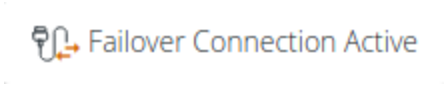
Cellular - the default failover interface for cellular-capable models.

NET2 - the default failover interface for non-cellular models.

3. When you have completed the OOB Failover set-up, ensure the OOB Failover status is set to **Enabled**, then, click **Apply**, a confirmation is displayed.
4. On the **Network Interfaces** page the Failover Interface will display "Configured for OOB Failover" beside the interface name.



5. When failover is triggered, the interface will be marked with the warning: **OOB Failover Active** to an Admin user when logged in.

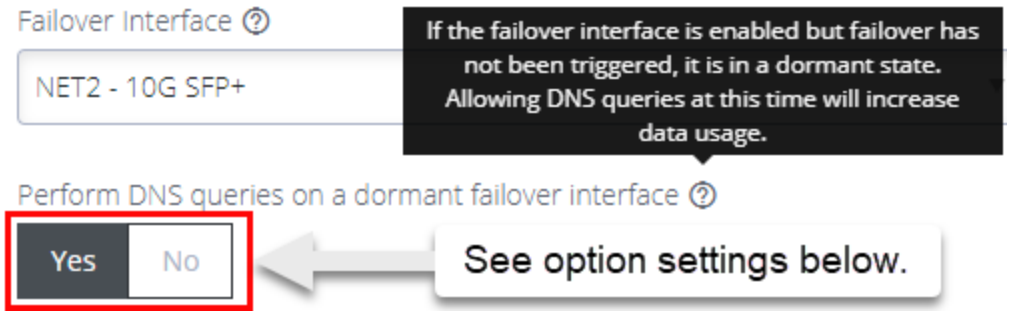


> **Note:** It may take up to five minutes for a failover to actually occur once the probe stops connecting to the probe address.

> **Note:** The shortcut button **Enabled/Disabled** is disabled or removed when an interface is in active failover.

DNS QUERIES ON A DORMANT FAILOVER INTERFACE

The Dormant DNS option allows DNS queries on the failover interface to be disabled in normal operation so that DNS queries can be paused.



The option configures how the DNS name servers and search domains configured for the failover interface are used by the system.

- If set to **Yes**, the DNS name servers and search domains configured for the failover interface will always be available to the system for DNS name resolution. Allowing DNS queries while failover has not been triggered make it more likely that DNS requests will be made over the cellular interface which will increase

data usage.

- If set to **No**, the DNS name servers and search domains will be made available to the system only when the failover state is active.

To configure the DNS name servers and search domains, see "DNS Configuration" on page 138.

OOB FAILOVER TYPES & FAILOVER BEHAVIOR

OOB Setting	Failover Interface	Mode	Description
Disabled	Enabled	Always up OOB	When OOB Failover is disabled, the default outgoing interface cannot be specified, the default route is selected automatically. Outbound network connections (e.g. VPN client tunnels, SNMP alerts) are established according to the main static routing table, regardless of network state.
Enabled	Disabled	Failover mode	Failover detection is enabled on the selected “probe” interface. The network or cellular interface remains in a down state with no network configuration. When failover is ini-

			tiated, the network or cellular interface is started and configured. If a default route is installed on the interface, it takes precedence over the default route on the failed “probe” interface. Outbound network traffic (e.g. VPN client tunnels, SNMP alerts) are established or re-established over network or cellular connection during failover. The advantage of this mode is the secondary connection is completely inactive during normal operation which may be advantageous where the goal is to keep the interface off the Internet as much as possible, e.g. a cellular plan with expensive data rates and no carrier-grade NAT.
Enabled	Enabled	Dormant failover	Failover detection is enabled. Only inbound connections on the network or cellular interface are routed back out the network or cellular interface, to enable OOB access from remote networks (e.g. incoming SSH). Otherwise, outbound network con-

			nections (e.g. VPN client tunnels, SNMP alerts) are established according to the main static routing table, regardless of network state. When failover is initiated, the default route of the network or cellular interface takes precedence over the failed “probe” interface. Outbound network traffic (e.g. VPN client tunnels, SNMP alerts) are established or re-established over the network or cellular connection during failover. The advantage of this mode is the network or cellular connection is available for inbound out-of-band access during normal operation.

USER MANAGEMENT

CONFIGURE > USER MANAGEMENT

Under the User Management menu, you can create, edit, and delete groups and users, as well as assign users to groups. You can also set up remote user authentication.

GROUPS

CONFIGURE > USER MANAGEMENT > Groups

Groups are used to grant privileges to users. When a user is a member of a group, defined privileges may be granted to the group by an administrator. When editing a group, the (authorised) user selects from a list of devices, all of which are under the heading SERIALLY CONNECTED DEVICES.

PERMISSION CHANGES IN THE WEB UI

A new feature change called Access Rights is introduced in release 22.11 replaces the previous concept of a user *Role* and instead uses a set of configurable *Access Rights* for each group. Each access right governs access to a particular feature (or set of highly related features), with a user only having access to features for which they have an assigned access right.

> **Tip:** To support the new permissions model several rest API endpoints have been updated for the new functionality. Wherever possible, these changes backwards compatible. See the release noted for details.

UNDERSTANDING ACCESS RIGHTS

An access right is a permit authorizing access to a feature or collection of related features. Holders of the permit (i.e. the access right) are given access to the feature.

A user gains access rights by the following:

- Access Rights are assigned to Groups.
- Users are members of zero or more Groups.
- A User inherits all Access Rights from all the Groups they are a member of.

Some features may require the user to hold multiple access rights to access the feature through a specific interface. For example, a user needs the “right to use the web UI” and the “right to configure serial ports” to make configuration changes to a serial port through the web UI.

DEFINED ACCESS RIGHTS

There are four *defined* rights (admin, web_ui, pmshell, and port_config) as summarized in the following table.

Access Rights	Description
admin	The admin access right grants a holder access to everything; every feature and every user interface.
web_ui	Permits access for an authenticated user to basic status information via the web interface and rest API. Users can: • Make requests to the subset of endpoints that provide this same information. In both cases the user must be authenticated. • See information about their own user and groups. • See serial port status information for the specific ports to which the user is granted access.
pmshell Restricted CLI	Permits access to devices connected to serial ports. Does not give permission to configure all serial ports, only to those that are added to the same group containing the pmshell rights.

Port Config	Permits access to configure serial ports. This access right gives the holder the ability to configure serial ports. This right does not give the holder the ability to access the serial port.

Tip: A right may be combined with another right for a feature to be accessible by a user. For example, `web_ui` to login and port_config to configure a serial port. The `port_config` right by itself is not useful.

Admin Access Right (`admin`)

Any user who was previously an `Administrator` role now inherits the `admin` access right, giving that user the same “can do everything” permission.

Tip: The **Admin Access** toggle switch in the Web UI hides other rights selections as Admin Access overrides all other rights.

Web UI Access Right (`web_ui`)

Any user who was previously a Console User role now inherits the `web_ui` and `pmshell` access rights and there are no functional changes for this user.

Tip: From release 22.11 in the Web UI, the **Rights** checkbox replaces the **Roles** drop-down selection.

The `web_ui` access right grants the user the ability to

- log into the Web UI,
- see a listing of serial ports (The “Access → Serial Ports” menu item) and to
- edit a restricted set of user configuration such as changing their own password.

Portmanager Shell Access Right (`pmshell`)

Any user who was previously a Console User role now inherits the `pmshell` access rights and there are no functional changes for this user.

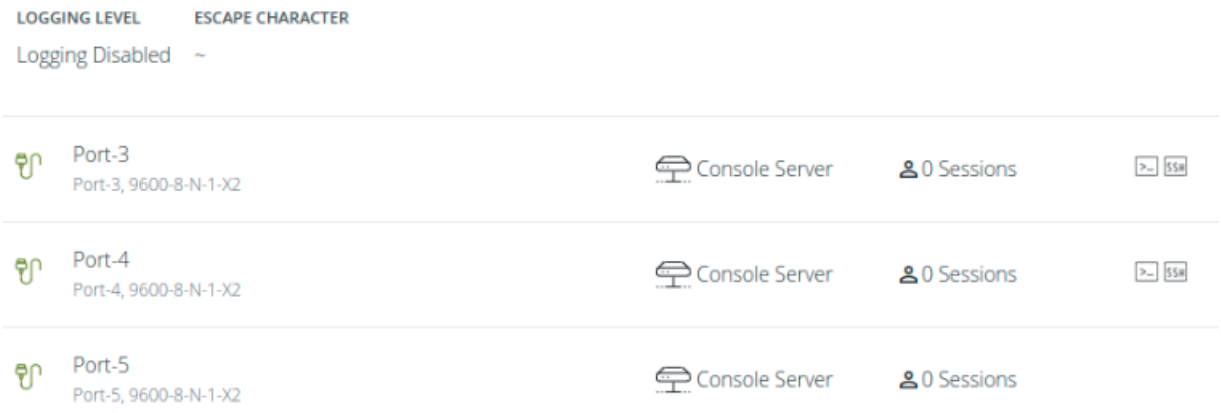
The `pmshell` access right grants the user access to the serial port web terminals and the ability to use `pmshell` over SSH. These rights are applied only to the access ports to which they have been granted rights.

Port Configuration Access Right

The `port_config` access right grants the holder of this right the ability to make configuration changes to the serial ports they have been assigned. Note that a user without the `web_ui` right cannot login to the web UI to configure serial ports, so a user must inherit the `web_ui` from at least one group.

Access > Serial Ports View

Users with the `port_config` access right to some serial ports are able to see the **Edit** link on the **Access > Serial Ports** page for those ports only. Non-admin users with the `port_config` role are able to see any active sessions on a port, but are not able to terminate the session.



Configure > Serial Ports View

The Configure Serial Ports page is accessible to users with the `port_config` and `web_ui` access rights appear in the navigation sidebar menu. This page lists ports that the user has both `port_config` and `web_ui` access rights.

Tip: It is possible to edit all details on these ports, however, changing the “mode” of a port will disconnect any sessions.

Non-Admin Users

Non-admin users with `port_config` access right are able to perform Serial Port Autodiscovery on the ports that they are able to configure. If autodiscovery is already running, they will be able to see the banner but will not be able to view the autodiscovery logs or cancel the running job. Non-admin users are not able to configure the Serial Port Autodiscovery Schedule and the icon is hidden, but are able to see which ports are configured of the ports to which they have access.

PROTECTED GROUPS AND USERS

Certain types of groups and users have protected status, meaning that they cannot be changed or deleted. Protected groups comprise the following:

`root` - The root user is hard-coded member of the admin group. As such, the root user cannot be deleted.

`admin` - The admin group cannot be disabled or changed to a non-admin group.

`netgrp` - The special ‘netgrp’ also cannot be deleted. This group is assigned to users from AAA auth that don’t have a group assigned from the authentication server.

Tip: For these protected groups no 'Delete' button appears beside them in the Web UI.

UNDERSTANDING SERIAL PORT ACCESS

Serial ports are assigned to a group in the same way as access rights are assigned to a group, however, it is the access rights that are assigned to the same group that determine what a user can actually do with those serial ports. The access rights assigned to one group will only apply to the serial ports assigned to that same group, they do not apply to the serial ports of another group.

For example, a user in a group with `port_config` and `port-01` can configure that port but not access the device (as that requires `pmshell` access rights).

Consider the following two groups, *Accounts Admin* and *Port #03 User*:

Group Name	Accounts Admin	Port #03 User
Access Rights	`port_config` `web_ui`	`pmshell` `web_ui`
Serial Ports	`port-01` `port-02`	`port-03`

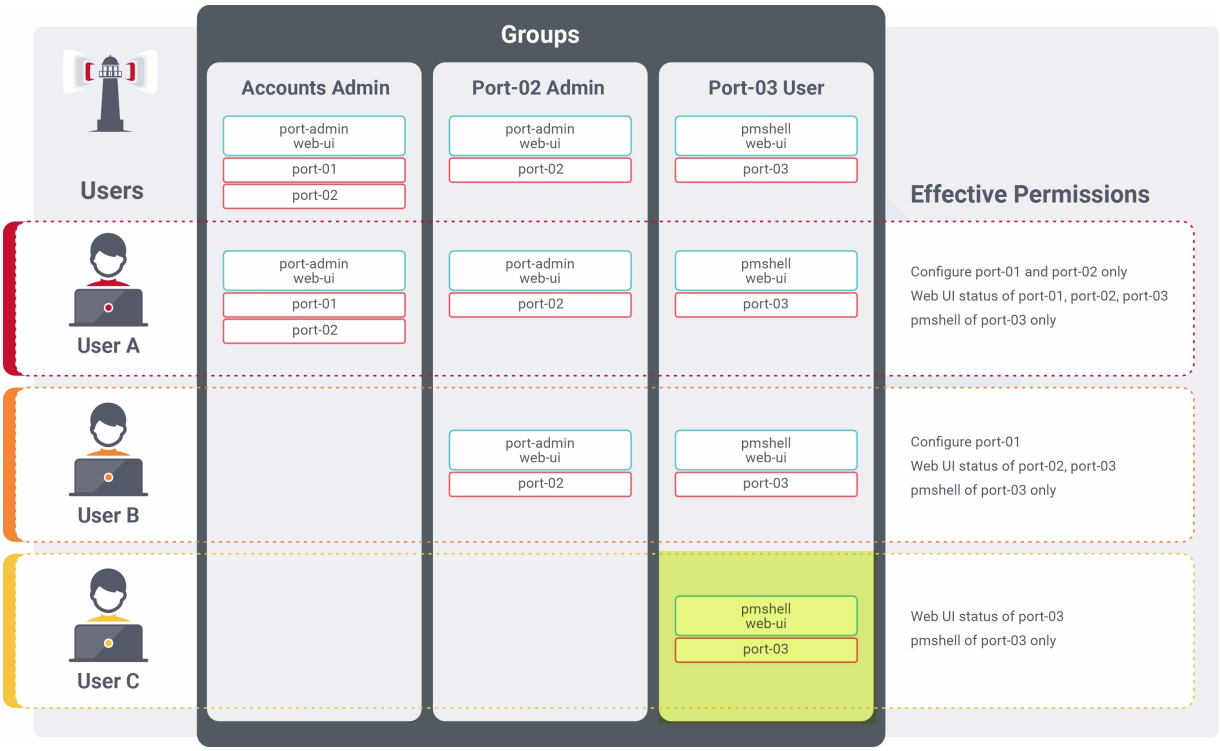
The effective rights for a user in one or both of those groups is shown in the following table. It shows how access rights assigned to one group will only apply to the serial ports assigned to that same group:

The following table shows the effective rights for a user in one or both of those groups, *Accounts Admin* and *Port #03 User*:

Group Membership **Action**	***Accounts Admin***	***Port #03 User***	***Accounts Admin & Port #03 User***
Configure `port-01`	✓	✗	✓
Configure `port-02`	✓	✗	✓
Configure `port-03`	✗	✗	✗
Access `port-01`	✗	✗	✗
Access `port-02`	✗	✗	✗
Access `port-03`	✗	✓	✓

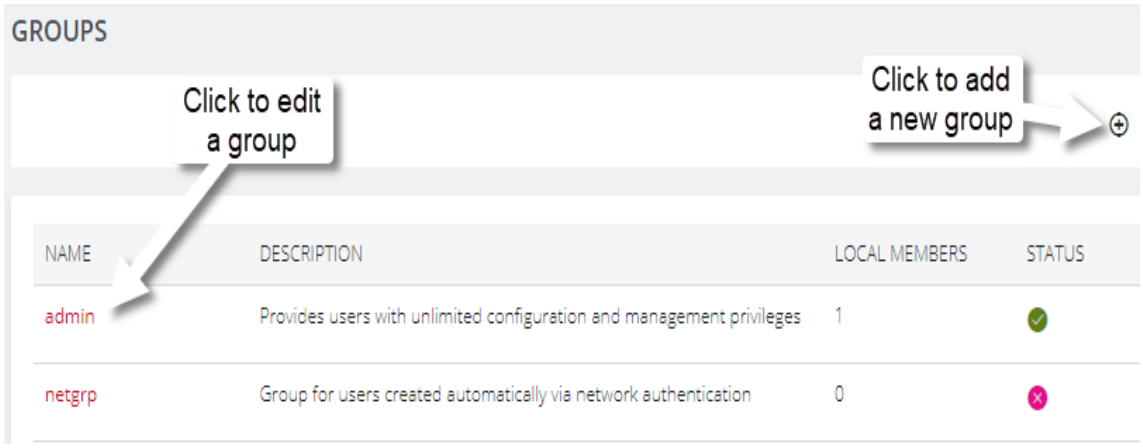
Note: Note the highlighted cell; a user with `pmshell` access to `port-03` (from the *Port #03* user group) does not also get `port_config` for that port, even though that access right is inherited from the *Accounts Admin* group. The access rights of a group *only apply to the serial ports in that same group*. This principle is illustrated in the following figure:

The figure below shows how access rights assigned to one group only apply to the serial ports assigned to that same group.



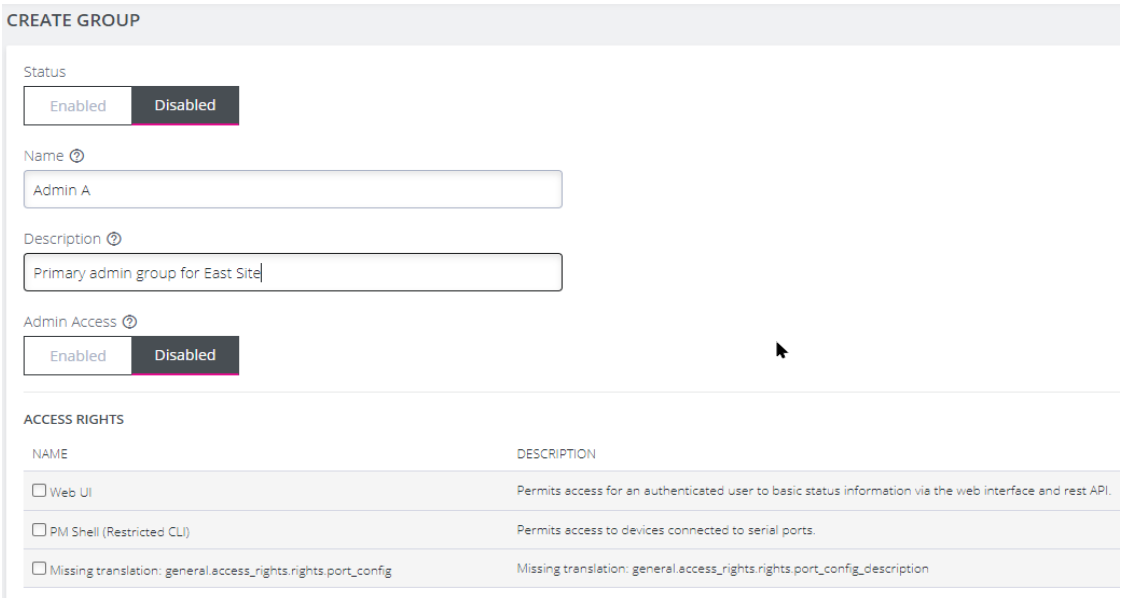
CREATE A NEW GROUP

1. Select **CONFIGURE > USER MANAGEMENT > Groups.**



⊕	**Add** a new group.
admin	Click on the group name to edit an existing group.
Status Enabled Disabled	In the EDIT GROUP window - Enable/Disable an existing group.
Admin Access ⓘ Enabled Disabled	Grant admin access rights and full control of this console, and all attached devices, to all users of this group.
🗑	Delete a group (or delete selected groups).

2. Click the **Add New Group** button. The **CREATE GROUP** page opens.



3. Enter a **Group Name**, **Description,** and set **Admin Access** to **Enabled** or **Disabled**. Specific access rights can be selected in the ACCESS RIGHTS area.

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

> **Note:**If **Admin Acess** is Enabled, members of the group will have full access to and control of selected managed devices, and the rights that are selected under **ACCESS RIGHTS** for that group.

4. Select the applicable **Access Rights** for the group (see the below table).
5. If the new group is to be activated immediately, set the group Status to **Enabled**.

6. Click the **Submit** button to save the group. After creation, group **Status** and **Admin Access** may be enabled or disabled from the **CONFIGURE > USER MANAGEMENT > Groups > EDIT GROUP** page.

EDIT AN EXISTING GROUP

1. Select **CONFIGURE > USER MANAGEMENT > Groups**.
2. Click on the name of the group to be modified and make desired changes.
3. Click **Submit** to save the changes

The **CONFIGURE > User Management > Groups** page also allows administrators to delete a group. 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. By default, netgrp has the Administrator role and is disabled. It must be enabled to take effect for remote AAA users.

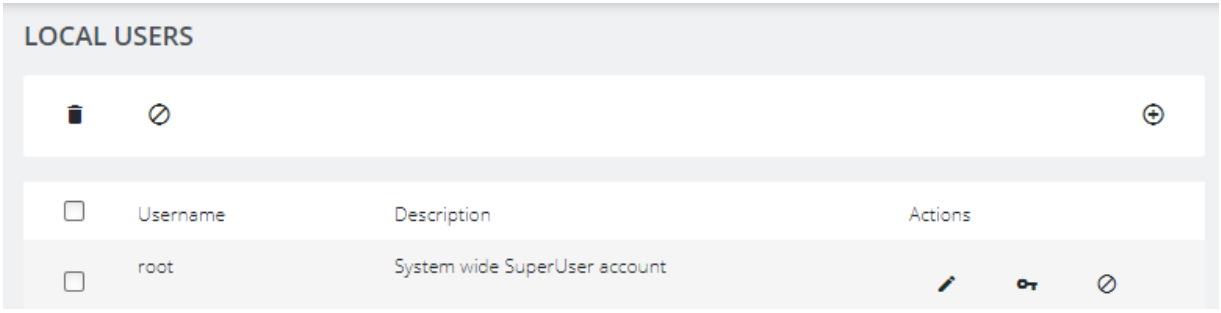
Note: For users that don't have any group, they are still part of netgrp, even if the netgrp membership is not explicitly enabled for the user.
The permissions for the netgrp members is a union of the permissions that have been given in the netgrp AND the permission for the user in AAA (TACACS+, RADIUS, etc).
If your netgrp "role" says "Console User" and you have priv-lvl 13 in TACACS+ (level 15 being the highest), then the union of that is like an admin already, so setting "console user" in netgrp does not matter.

LOCAL USERS

CONFIGURE > USER MANAGEMENT > Local Users

The Local Users feature allows a single point for the creation or management of local user accounts. The Local Users feature can use SSH authorized keys to control user access by using their local password; it is a point of control for:

- Authentication and authorization.
- Creating and editing user descriptions.
- Local passwords.
- User roles (admin or co sole user).
- Accessible ports.



See the Button Action Definitions table on the following page:

Button Action Definitions:

	Add a new local user.
	Edit an existing user.
	Enable an existing user.
	Manage SSH Authorized Keys.
	Disable an existing user (or disable selected users).
	Delete a user (or delete selected users).

CREATE A NEW USER WITH PASSWORD

Note: Users are prevented from using the word “default” as their password. The factory default password automatically expires after a factory reset and users must choose a new password. This password policy applies to the WebUI, Config Shell and CLI. users configured on the system using software versions prior to 23.10 with password “default” are forced to change the user password to something other than “default” after upgrading to 23.10. This password feature update applies to configured boxes with existing users, not just factory defaulted software.

1. Navigate to the **CONFIGURE > USER MANAGEMENT > Local Users** page.

2. Click the **Add User** ⊕ button. The **New User** dialog appears.
3. Enter a Username, Description, and Password that the new user will use.
4. Re-enter the **Password** in the **Confirm Password** field.
5. Select the **Enabled** checkbox.
6. Click **Apply**. A banner will confirm that the data has been saved.

CREATE A NEW USER WITH NO PASSWORD (REMOTE AUTHENTICATION)

To create a new user with no password.

> **Note:** If a new user is created with no password, this will cause the user to fall-back use remote authentication.

1. Select **CONFIGURE > User Management > Remote Authentication**
2. Select a Mode.
3. Enter Settings and click **Apply**.
4. Select **CONFIGURE > USER MANAGEMENT > Local Users**
5. Click the **Add User** button. The **New User** dialog loads.
6. Enter a **Username**, **Description**.
7. Select the **Remote PasswordOnly** checkbox.
8. Select the **Enabled** checkbox.
9. Click **Apply**. A banner will confirm that the data has been saved.

MODIFY AN EXISTING USER ACCOUNT WITH PASSWORD

1. Select **CONFIGURE > USER MANAGEMENT > Local Users**
2. Click the **Edit User** button and make the required changes.
3. Click **Save User**. A banner will confirm the changes have been saved.

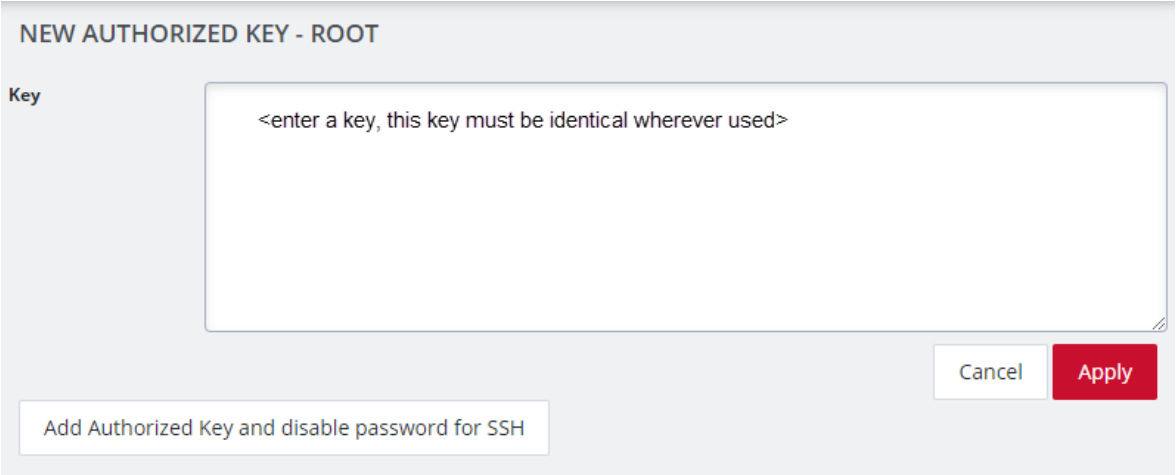
The **Edit Users** dialog allows the user's **Description** to be changed, **Group Memberships** modified, and the user's **Password** to be reset. The username cannot be changed. To disable a user, uncheck the **Enabled** checkbox.

Note: Users of disabled accounts cannot log in to the Console Manager using either the Web-based interface or via shell-based logins.

MANAGE SSH AUTHORIZED KEYS FOR A USER ACCOUNT

To manage SSH authorized keys for a user:

1. Select **CONFIGURE > USER MANAGEMENT > Local Users**
2. Click the **Manage SSH Authorized Keys** button for that user.
3. Click the **Add Authorized Key** button to add a new key. This opens the **NEW AUTHORIZED KEY** page for this user.



4. Enter the key and click **Apply**. You can also click on **Add Authorized Key** and disable password for SSH for this user from this page.
5. To delete a key, click **CONFIGURE > USER MANAGEMENT > Local Users** and click the **Manage SSH Authorized Key** button for the user.
6. Click the **Delete** button next to the key you wish to remove.

DELETE A USER'S ACCOUNT

To delete a user's account:

1. Select **CONFIGURE > USER MANAGEMENT > Local Users**
2. Click the **Delete User** button in the **Actions** section next to the user to be deleted.
3. Click **Yes** in the **Confirmation** dialog.

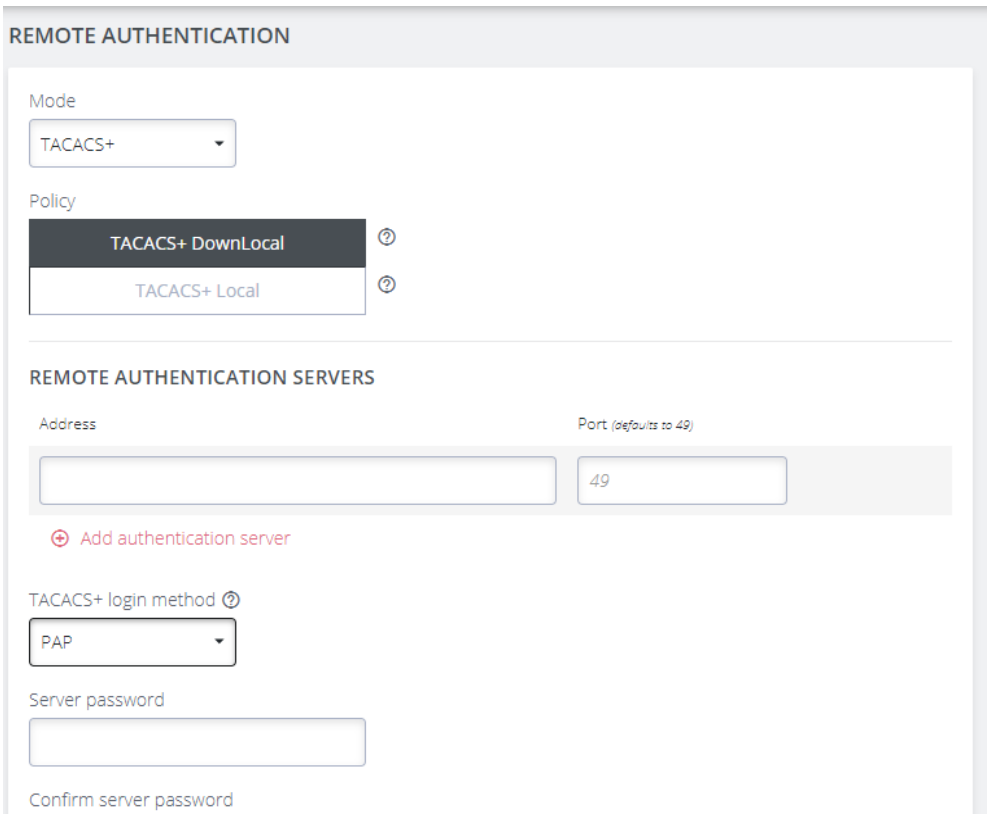
REMOTE AUTHENTICATION

CONFIGURE > USER MANAGEMENT > Remote Authentication

The Console Manager supports three AAA systems. Select the remote authentication mode to be applied (DownLocal, or Local apply for all modes):

- RADIUS
- TACACS+
- LDAP

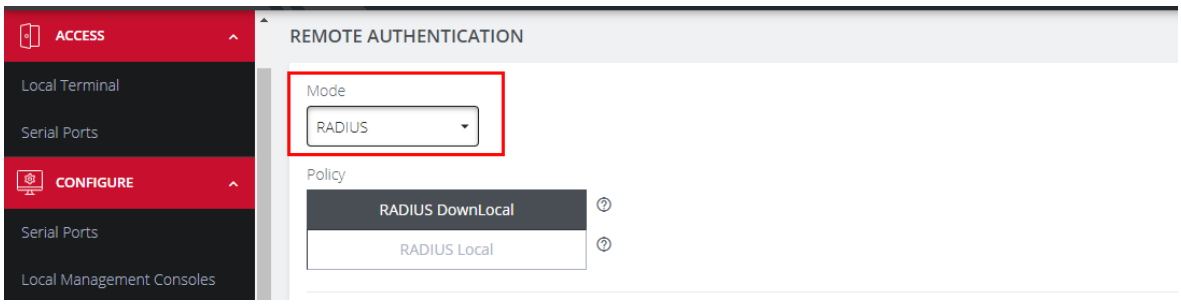
Navigate to **CONFIGURE > USER MANAGEMENT > Remote Authentication**, the Remote Authentication Home page is displayed.



Tip: All fields in the Remote Authentication form have tooltips that provide additional information to assist with completing the form fields.

CONFIGURE RADIUS AUTHENTICATION

1. Under **CONFIGURE > User Management > Remote Authentication**, select **RADIUS** from the **Mode** drop-down menu.



2. Select the preferred Radius Remote Authentication policy to be applied: **Radius DownLocal**, or **Radius Local** (see the tips below).

   Tip: RADIUS DownLocal: Users are authenticated through their local account only if the remote AAA server is unreachable or down. If the credentials provided at login are incorrect or if the account does not exist on the remote server, the user is denied access.

   Tip: RADIUS Local: If remote authentication fails because the user account does not exist on the remote AAA server, the CM attempts to authenticate the user using a local account as per a regular local login

3. Add the **Address** and optionally the **Port** of the authentication server.
4. Add the **Address** and optionally the **Port** of the RADIUS accounting server.
5. Add and confirm the **Server password,** also known as the RADIUS Secret.
6. Click **Apply**.

Note: Multiple servers can be added. The RADIUS subsystem will query 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.

CONFIGURE TACACS+ AUTHENTICATION

1. Under **CONFIGURE > USER MANAGEMENT > Remote Authentication**, select TACACS+ from the **Mode** drop-down menu.
2. Select the preferred TACACS+ Remote Authentication policy to be applied: **TACACS+ DownLocal**, or **TACACS+ Local** (see the tips below).

   > **Tip: TACACS+ DownLocal:** Users are authenticated through their local account only if the remote AAA server is unreachable or down. If the credentials provided at login are incorrect or if the account does not exist on the remote server, the user is denied access.

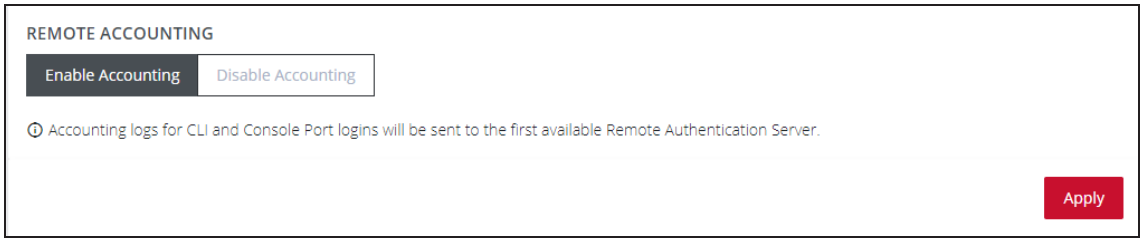
   > **Tip: TACACS+ Local:** If remote authentication fails because the user account does not exist on the remote AAA server, the CM attempts to authenticate the user using a local account as per a regular local login.

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

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

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

7. Enable or Disable **Remote Accounting**.
   TACACS Accounting is enabled by default, the Remote Auth Server is used as the Accounting server. However one or more Accounting Servers can be specified.
   a. To disable Remote Accounting, select **Disable**
   b. To enable Remote Accounting, select **Enable**.



8. Click **Apply.**

Note: For Cisco ACS, see Setting up permissions with Cisco ACS 5 and TACACS+ on the Opengear Help Desk.

CONFIGURE LDAP AUTHENTICATION

1. Under **CONFIGURE > User Management > Remote Authentication**, select **LDAP** from the **Mode** drop-down menu.

2. Select the preferred LDAP Remote Authentication policy to be applied: **LDAP DownLocal**, or **LDAP Local** (see the tips below for explanation).

   Tip: LDAP DownLocal: Users are authenticated through their local account only if the remote AAA server is unreachable or down. If the credentials provided at login are incorrect or if the account does not exist on the remote server, the user is denied access.

   Tip: LDAP Local: If remote authentication fails because the user account does not exist on the remote AAA server, the **CM** will attempt to authenticate the user using a local account as per a regular local login.

2. Add the **Address** and optionally the **Port** of the LDAP server to query.
3. Add the **LDAP Base DN** that corresponds to the LDAP system being queried. For example:

   ```
   CN=example-user,CN=Users,DC=example-domain,DC=com
   ```

4. Add the **LDAP Bind DN**. This is the distinguished name of a user with privileges on the LDAP system to perform the lookups required for retrieving the username of the users, and a list of the groups they are members of.
5. Input the password for the **LDAP Bind DN** user and confirm the password.
6. Add the **LDAP Username Attribute**. This depends on the underlying LDAP system. Use sAMAccountName for Active Directory systems, and uid for OpenLDAP based systems.
7. Add the **LDAP Group Membership Attribute**. This is only needed for Active Directory and is generally memberOf.

8. If desired, check **Ignore referrals** option. When checked, LDAP will not follow referrals to other remote authentication servers when logging users in. If multiple remote authentication servers exist on the network, checking this option may improve log in times.

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

LOCAL PASSWORD POLICY

CONFIGURE > USER MANAGEMENT > Local Password Policy

A Password Complexity policy allows network administrators to implement and enforce a password policy that meets the customers' security standards for local users (including root). This functionality enables administrators to mandate the setting of complex passwords thus making it difficult for malicious agents to succeed in password attacks.

Enabling this feature will:

- Enforce the use of complex passwords so as to improve security.
- Schedule expiry of passwords to enforce regular password updates.

Note: Password policy such as complexity and expiry can only be configured by an administrator. Password requirements are applied to all accounts.

Tip: Password policy may be enabled and configured via the Web GUI, rest-api and ogcli. The password policy also applies to underlying CLI tools.

SET PASSWORD COMPLEXITY REQUIREMENTS

CONFIGURE > USER MANAGEMENT > Local Password Policy

Note: Some password complexity rules are required, other rules are optional. Optional rules can be selected by clicking on the relevant checkbox.

Note: Users are prevented from using the word “default” as their password. The factory default password automatically expires after a factory reset and users must choose a new password. This password policy applies to the WebUI, Config Shell and CLI. users configured on the system using software versions prior to 23.10 with password “default” are forced to change the user password to something other than “default” after upgrading to 23.10. This password feature update applies to configured boxes with existing users, not just factory defaulted software.

See also "Password Policy Implementation Rules" on page 221

To set the password complexity requirements:

1. Navigate to CONFIGURE > USER MANAGEMENT > Local Password Policy.
2. Click the **Enforced** button to implement the password complexity policy (the policy is not activated until the **Apply** button is clicked).
3. Enter the information required to form the password complexity rules to comply with your company policy:
    - Password cannot be a palindrome (required)
    - Minimum length (required)

- Must contain an upper case letter (optional)
- Must contain a numeric character (optional)
- Must contain a special character (non-alphanumeric eg. e.g. #,$,%)
- Disallow user names in passwords (optional)

See "Password Policy Implementation Rules" on page 221

4. Click the **Apply** button to activate the password complexity policy.

SET PASSWORD EXPIRATION INTERVAL

CONFIGURE > USER MANAGEMENT > Local Password Policy

See also "Password Policy Implementation Rules" on the next page

Password Expiration schedules the expiry of passwords to enforce regular password updates. When this feature is applied and a password becomes expired, an expired password prompt is displayed at log-in.

> **Note:**The Password Expiration policy affects local passwords only and does not apply to remote authentication modes.

To set the password expiration interval:

1. Navigate to CONFIGURE > USER MANAGEMENT > Local Password Policy.
2. Click the **Enabled** button to implement the password expiration policy (the policy is not activated until the **Apply** button is clicked).
3. Input a number to represent the desired number of days between mandatory password updates. The default time is 90 days and the minimum is 1 day.
4. Click the **Apply** button to activate the password interval policy.

PASSWORD POLICY IMPLEMENTATION RULES

Rule	Policy
Expiry Rules	The expiry time is measured in number of whole days. When the expiry period is reached users are required to update their password on their next login. The default expiry period is 90 days and the minimum is one (1) day.
	If there are existing user passwords when the expiry is enabled, the expiry time will be applied from when the password was initially set by the user. If a password falls outside the new expiry period the user will be immediately prompted to change the password.
	Local Password policy is only applied to local passwords and does not apply to remote authentication modes.
	When local password policy is enabled it will remain in force until the feature is turned off.
	If the minimum password length is modified and then the password complexity feature is disabled, the minimum length requirement is not updated.
Complexity Rules	The password cannot be a palindrome (this requirement cannot be disabled except by disabling password complexity entirely). (A palindrome is a word or other sequence of characters that reads the same backward as forward, such as *madam* or *racecar*).
	The minimum length (enforced) must be at least 8 char-

	acters (this requirement cannot be disabled except by disabling password complexity entirely).
	The password should contain at least one upper case alphabetic character (enabled or disabled separately).
	The password must contain at least one numeric character (enabled/disabled separately).
	The password should contain at least one special character (e.g. #,$,%) (enabled/disabled separately).
	The password cannot contain your user-name.
	Complexity requirements will apply when a user next tries to update their password.
	An administrator can force the expiry of a users password by running the ogCLI command: `passwd --expire {username}` to force a user to change their password.
	The operations `ogadduser`, `ogpasswd` and `ogsshaddsshkey` have been removed. You should instead use ogCLI for these operations.

SERVICES

CONFIGURE > SERVICES

The **CONFIGURE > SERVICES** menu lets you manage services that work with the Console Manager.

FIPS COMPLIANCE

The Federal Information Processing Standard Publication 140-2 (FIPS 140-2) is a U.S. government computer security standard that is used to approve cryptographic modules. Opengear appliances operating in FIPS mode provide FIPS 140-2 level one compliance by utilizing FIPS validated OpenSSL 3.0.8 cryptographic library while in FIPS mode.

Note: The default provider will be 3.0.10, however, the FIPS provider remains on 3.0.8 in release 23.10.4. See the example of list providers later in this topic under the section "Verify that FIPS is enabled" on the next page.

CONFIGURE FIPS

Enable FIPS mode at the CLI as follows:

ENABLE FIPS

Enable FIPS via config shell:

```
root@<device name>:~# config
Welcome to the Opengear interactive config shell. Type ? or help for
help.
config: system/fips
config(system/fips): enabled true
config(system/fips): apply
Updating entity system/fips.
```

Enable FIPS via ogcli:

```
ogcli update system/fips enabled=true
```

DISABLE FIPS

Disable FIPS via config shell:

```
root@<device name>:~# config
Welcome to the Opengear interactive config shell. Type ? or help for
help.
config: system/fips
config(system/fips): enabled false
config(system/fips): apply
Updating entity system/fips.
```

Disable FIPS via ogcli:

```
ogcli update system/fips enabled=false
```

VERIFY THAT FIPS IS ENABLED

1. Check the OpenSSL FIPS providers.

```
root@<device name>:~# openssl list -providers
Providers:
 default
  name: OpenSSL Default Provider
  version: 3.0.10
  status: active
 fips
```

```
name: OpenSSL FIPS Provider
version: 3.0.8
status: active
```

2. Check that the digest algorithms provided by OpenSSL is limited to FIPS compliant ciphers/algorithms.

```
root@<device name>:~# openssl list -digest-algorithms
...
Provided:
  { 2.16.840.1.101.3.4.2.1, SHA-256, SHA2-256, SHA256 } @ default
  { 2.16.840.1.101.3.4.2.10, SHA3-512 } @ default
  { 2.16.840.1.101.3.4.2.8, SHA3-256 } @ default
  { 2.16.840.1.101.3.4.2.7, SHA3-224 } @ default
  { 2.16.840.1.101.3.4.2.2, SHA-384, SHA2-384, SHA384 } @ default
  { 2.16.840.1.101.3.4.2.3, SHA-512, SHA2-512, SHA512 } @ default
  { 2.16.840.1.101.3.4.2.5, SHA-512/224, SHA2-512/224, SHA512-224 } @
default
  { 2.16.840.1.101.3.4.2.12, SHAKE-256, SHAKE256 } @ default
  { 1.3.14.3.2.26, SHA-1, SHA1, SSL3-SHA1 } @ default
  { 2.16.840.1.101.3.4.2.9, SHA3-384 } @ default
  { 2.16.840.1.101.3.4.2.11, SHAKE-128, SHAKE128 } @ default
  { 2.16.840.1.101.3.4.2.4, SHA-224, SHA2-224, SHA224 } @ default
  { 2.16.840.1.101.3.4.2.6, SHA-512/256, SHA2-512/256, SHA512-256 } @
default
  { KECCAK-KMAC-128, KECCAK-KMAC128 } @ default
  { KECCAK-KMAC-256, KECCAK-KMAC256 } @ default
  { 2.16.840.1.101.3.4.2.1, SHA-256, SHA2-256, SHA256 } @ fips
  { 2.16.840.1.101.3.4.2.10, SHA3-512 } @ fips
  { 2.16.840.1.101.3.4.2.8, SHA3-256 } @ fips
```

```
  { 2.16.840.1.101.3.4.2.7, SHA3-224 } @ fips
  { 2.16.840.1.101.3.4.2.2, SHA-384, SHA2-384, SHA384 } @ fips
  { 2.16.840.1.101.3.4.2.3, SHA-512, SHA2-512, SHA512 } @ fips
  { 2.16.840.1.101.3.4.2.5, SHA-512/224, SHA2-512/224, SHA512-224 } @
fips
  { 2.16.840.1.101.3.4.2.12, SHAKE-256, SHAKE256 } @ fips
  { 1.3.14.3.2.26, SHA-1, SHA1, SSL3-SHA1 } @ fips
  { 2.16.840.1.101.3.4.2.9, SHA3-384 } @ fips
  { 2.16.840.1.101.3.4.2.11, SHAKE-128, SHAKE128 } @ fips
  { 2.16.840.1.101.3.4.2.4, SHA-224, SHA2-224, SHA224 } @ fips
  { 2.16.840.1.101.3.4.2.6, SHA-512/256, SHA2-512/256, SHA512-256 } @
fips
  { KECCAK-KMAC-128, KECCAK-KMAC128 } @ fips
  { KECCAK-KMAC-256, KECCAK-KMAC256 } @ fips
```

CONSIDERATIONS FOR USING THE FIPS FEATURE

In organizations where FIPS is required, the following points should be noted:

- OpenSSL 3.0.8 FIPS provider limits the available cryptography ciphers/algorithms only those that have been validated by laboratory to be FIPS compliant.

Caution: Configuration backup should be taken before enabling or disabling FIPS.

Caution: FIPS has the potential to break any service with secure connectivity, including services listed in the following table:

Feature	Affected Process/Service	Impact
Lighthouse enrollment	OpenVPN	OpenVPN is not compliant with FIPS standards; this issue is a recognized problem specifically when OpenSSL 3.x is being used. Once OpenVPN addresses this issue, it will also meet FIPS compliance standards. However, for compatibility with Lighthouse enrollment, this feature remains enabled although it is non-compliant.
IPsec	Strongswan	Needs to be operated in FIPS mode to be FIPS compliant. The other end of the tunnel does not need to be oper-

		ating FIPS mode to connect.
Remote authentication	freeradius, tacacs, ldap	These are not FIPS compliant.
NTP	chrony	Authenticated NTP servers with MD5 will not connect. Use an algorithm that is FIPS compliant.
SNMP	ogtrapd, snmpd, snmptrapd	Authentication and Encryption should be used as the security policy as V1 and V2 have no encryption. SNMPv3 with MD5 encryption will fail. Use an algorithm that is FIPS compliant. It is recommended that authPriv security policy is used when in FIPS mode for SNMPv3.
LDAP	OpenSSL	LDAP has no encryption, therefore it does not use OpenSSL. For FIPS compliance it is recommended that it is not used.
OpenSSL	OpenSSL MD5	When OpenSSL

		MD5 is not available, pam_tacplus uses its own implementation of MD5. When FIPS is enabled it does not use OpenSSL(but will continue to work). Therefore, it is recommended that it is not used in FIPS mode.
SMF	SMF	Use of the SMF feature will render the device non-compliant for FIPS.
SSH connections	SSH	For SSH connections, a FIPS compliant algorithm must specified as part of the command to connect. See the note below:
NetOps Modules	gre (Secure Provisioning) nom-ipaccess-lhvpn (IP access) nom-ag-lhvpn (Access Gateway)	Opengear NetOps Modules are not functional when FIPS mode is enabled.
Note: SSH will require the cipher to be manually specified when FIPS is enabled. e.g. ssh root@10.0.0.1 -c aes256-gcm@openssh.com		
Wireguard		Wireguard is not FIPS compliant and should not be used in FIPS mode.

Routing protocols		Routing protocols (eg. BGP), should not select an MD5 cipher.

BRUTE FORCE PROTECTION

CONFIGURE > SERVICES > Brute Force Protection

A brute-force attack is an attempt to discover a password by systematically trying every possible combination of letters, numbers, and symbols until the one correct combination that works.

Brute Force Protection offers an essential defense mechanism by automatically blocking access from offending source IP addresses.

> **Caution:** Brute Force Protection may prevent access to the system during an emergency.

CONFIGURE BRUTE FORCE PROTECTION

> **Note:**Brute Force Protection is enabled by default for SSH and Web UI.

To configure Brute Force Protection:

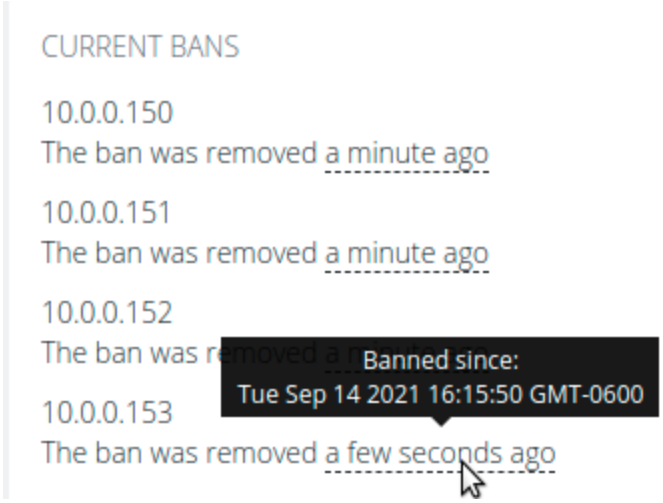
1. Navigate to **CONFIGURE > SERVICES > Brute Force Protection**.
2. Choose the desired settings as described below.
3. Click **Apply** to save the changes.

Field	Values	Description
SSH Protection	Enabled / Disabled	Enable Brute Force Protection for SSH login attempts.
HTTPS Protection	Enabled / Disabled	Enable Brute Force Protection for Web UI login attempts.
Maximum failed attempts	Attempts: 3 (minimum) Time period in minutes: 1 (minimum)	The number of failed access attempts permitted within the given time period before preventing access.
Lockout period	60 (minimum)	The number of seconds that an IP address will be banned after violating the Brute Force Protection policies.

VIEWING CURRENT BANS

IP addresses that are currently blocked appear in the CURRENT BANS section of the Web UI, displaying the address and remaining duration of the ban or how long ago the ban was lifted.

Hover over the ban time for more detailed information.



MANAGING BRUTE FORCE PROTECTION VIA COMMAND LINE

For more control over Brute Force Protection, administrative users can use the command line to configure the service and remove bans manually.

Description	Command	Notes
Display Brute Force Protection configuration	`ogcli get services/brute_force_protection`	
Update Brute Force Protection configuration	`ogcli replace services/brute_force_protection << END` `ban_time=180`	Ban time in seconds. Find time in minutes.

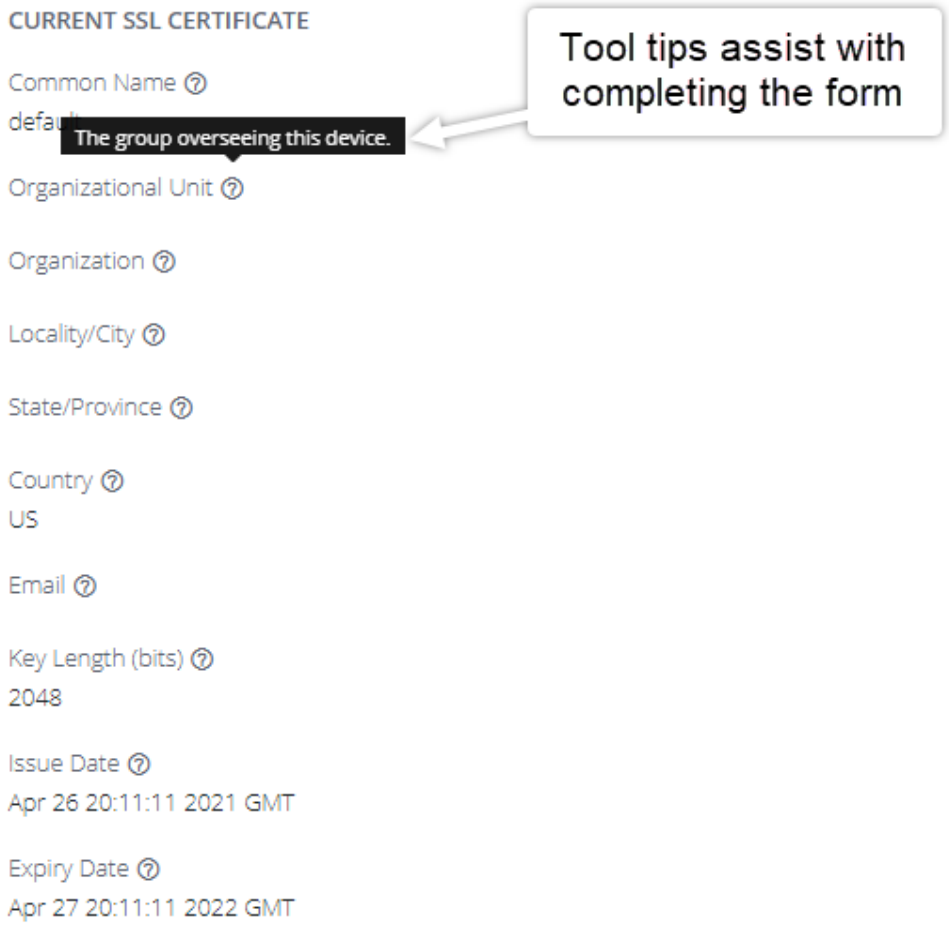
Description	Command	Notes
	find_time=1 https_enabled=false max_retry=4 ssh_enabled=true END	
Un-ban an IP address	fail2ban-client unban <ipaddress>	
Un-ban all current bans	fail2ban-client unban --all	
List SSH bans	fail2ban-client status sshd	SSH protection must be enabled.
List HTTPs bans	fail2ban-client status https	HTTPs protection must be enabled.
List all bans with ogcli	ogcli get monitor/brute_force_protection/bans	

HTTPS CERTIFICATE

CONFIGURE > SERVICES > HTTPS Certificate

The Console Manager ships with a private SSL Certificate that encrypts communications between it and the browser.

To examine this certificate or generate a new Certificate Signing Request, select **CONFIGURE > SERVICES > HTTPS Certificate**. The details of the **Current SSL Certificate** are shown on the landing page.



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

NETWORK DISCOVERY PROTOCOLS

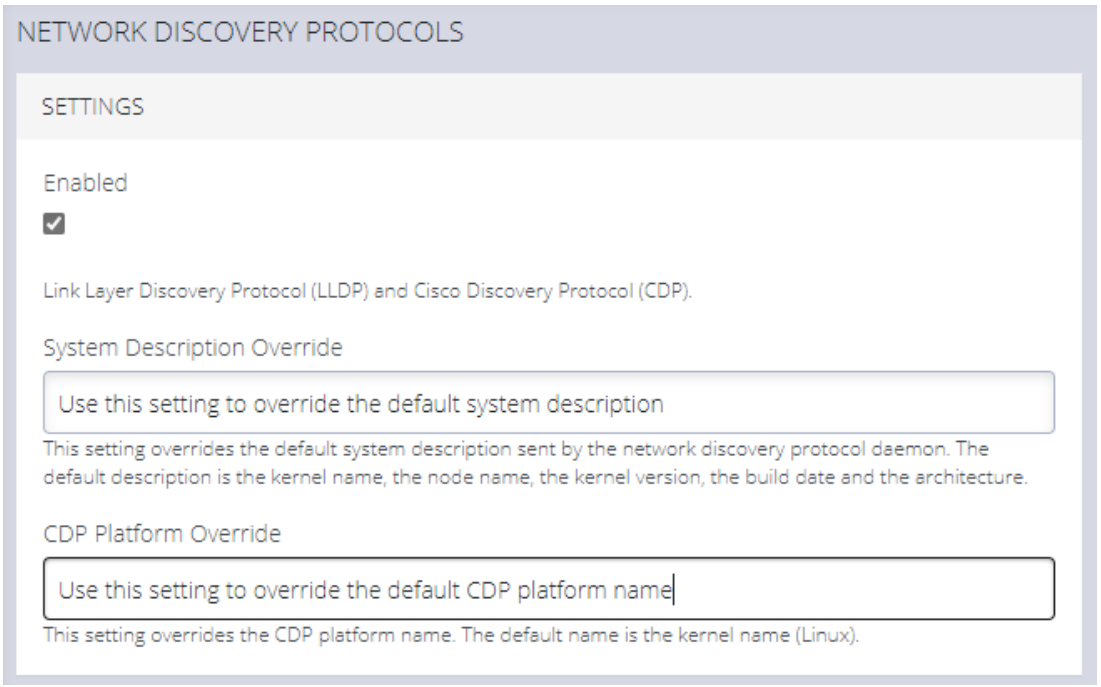
CONFIGURE > SERVICES > Network Discovery Protocols

The Console Manager displays LLDP/CDP Neighbors when enabled for a connection. See **CONFIGURE > SERVICES > Network Discovery Protocols** to enable/disable.

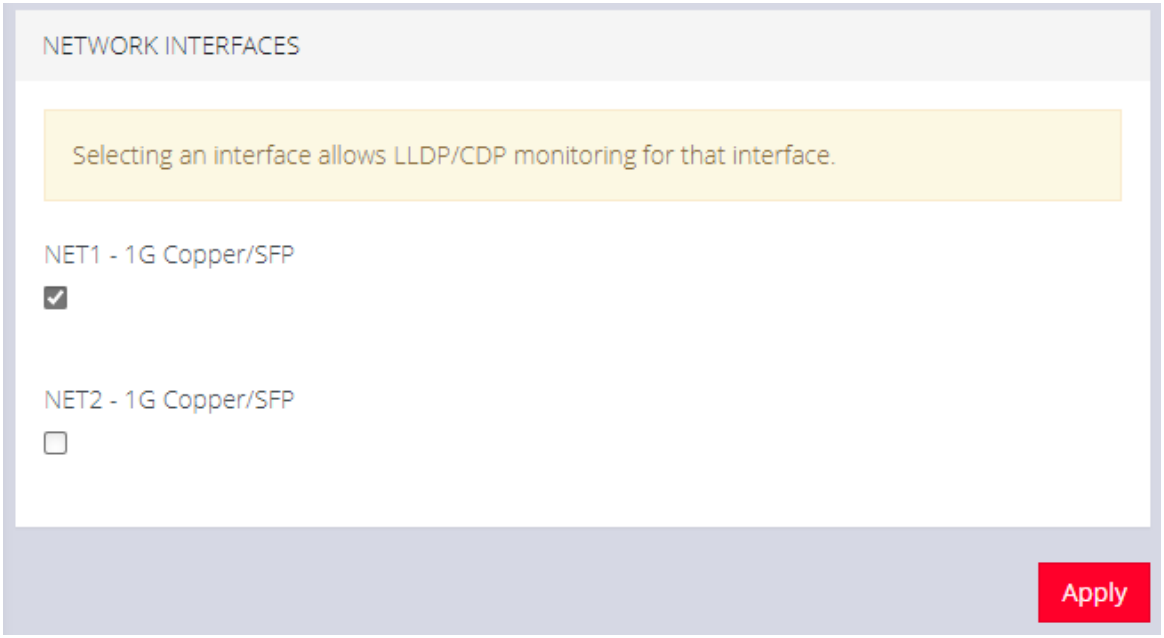
The **CONFIGURE > SERVICES > Network Discovery Protocols > LLDP/CDP NEIGHBORS** page allows you to enable this service by clicking the **Enabled** checkbox.

You can set a System Description that overrides the default system description sent by the network discovery protocol daemon. The default description is the kernel name, the node name, the kernel version, the build date and the architecture.

A value can be entered in the CDP Platform Override to override the CDP platform name. The default name is the kernel name (Linux).



Select one or more checkboxes in the **NETWORK INTERFACES** section of the page and click **Apply**.



FILE SERVER

CONFIGURE > SERVICES > File Server

The Console Manager can be configured to serve files to clients via Trivial File Transfer Protocol (TFTP).

TFTP can be used by nodes on the network to perform a network boot, or to allow backup and restore of configuration files.

Note: Limitations

- The user is responsible for disk space management.
- User permissions cannot be set on files at this time.

ENABLE TFTP SERVICE

Note: The TFTP service is disabled by default.

To enable the TFTP service:

- Click the **TFTP Enabled** button.

- Click **Apply** to save the changes.
- The TFTP service is now running with a default location of **/mnt/nvram/srv**.

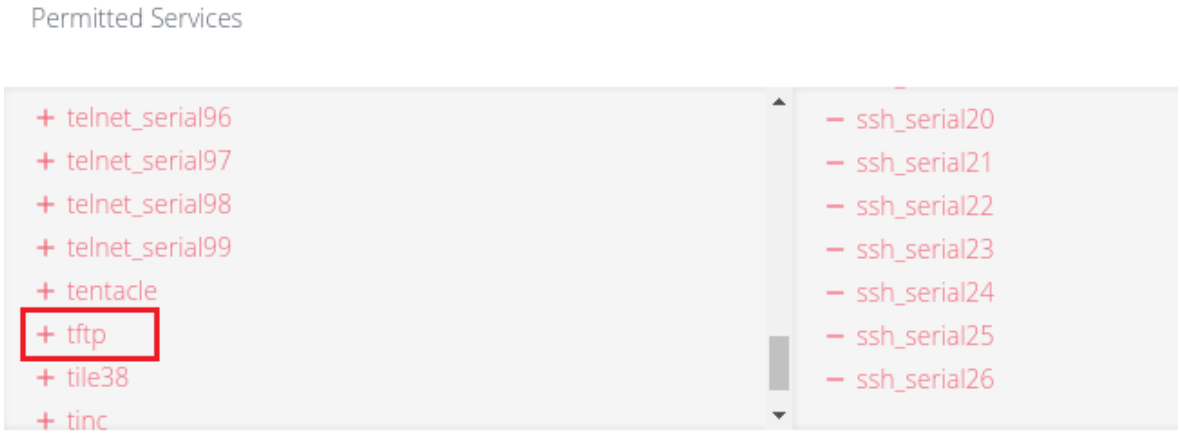
  This location is where all files uploaded to the TFTP server will be stored.

Note: The disk space usage information displayed on the page indicates the usage of the whole storage volume.

MODIFY FIREWALL ZONES TO ALLOW THE TFTP SERVICE TO BE USED

The TFTP service must be allowed through a firewall zone so that clients may upload and retrieve files.

- Navigate to the Firewall Management page via **CONFIGURE > FIREWALL > Management**.
- Expand the desired firewall zone and click the **Edit Zone** button.
- Allow the "tftp" service from the list of Permitted Services.



- Click **Apply** to save the changes.
- On the File Server page, the zones with TFTP enabled are now displayed.

UPDATE THE TFTP SERVICE STORAGE LOCATION

The location used by the TFTP service can be updated using the **ogcli** tool.

Note: The storage location must be an existing directory before running ogcli update.

Caution: Using a storage volume other than **/mnt/nvram** is not recommended. Data may be lost after reboot, or be inaccessible when switching boot slots.

- As an administrative user, run:

```
ogcli update services/tftp path=\"<new path>\"
```

ROUTING

CONFIGURE > SERVICES > Routing

The Console Manager supports Static Routing and Dynamic Routing. Static Routing and OSPF are currently configured via the ogcli interface, however, CLI configuration will not persist after a device reboot. Dynamic Routing can be enabled via the UI and configured using standard Free Range Routing interfaces (e.g. vtysh).

Note: Configuration set via vtysh (or other FRR interfaces) will need to be manually backed up in addition to a standard Opengear configuration export.

DYNAMIC ROUTING

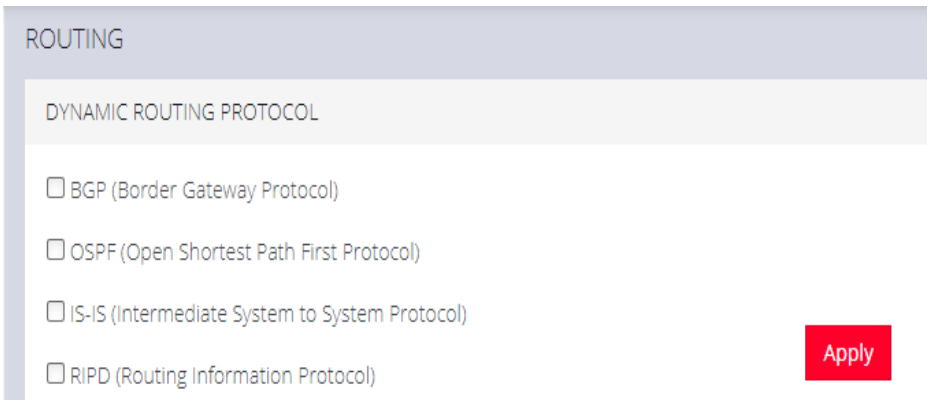
To enable Dynamic Routing on the CM, navigate to the **CONFIGURE > SERVICES > Routing** page.

Dynamic Routing supports four routing protocols, these are:

- BGP (Border Gateway Protocol)
- OSPF (Open Shortest Path First Protocol) (see "Routing" above later in this topic).
- IS-IS (Intermediate System to System Protocol)
- RIPD (Routing Information Protocol)

Select the preferred routing protocol then click **Apply**.

Note: If no protocol is selected, no route sharing services are run on the CM.



STATIC ROUTING (VIA THE OGCLI)

To enable Static Routing on the CM, open an ogcli terminal by navigating to **ACCESS > Local Terminal**.

STATIC ROUTING OGCLI HELP

For Help on implementing a Static Route protocol via ogcli, enter the command:

```
ogcli help static_routes
```

CREATE STATIC ROUTE - EXAMPLE:

```
ogcli create static_route << 'END'
destination_address="10.1.45.0"
destination_netmask=24
gateway_address="192.168.1.1"
interface="system_net_physifs-1"
metric=100
END
```

STATIC ROUTING ARGUMENTS

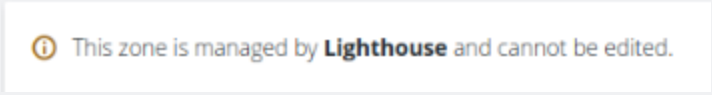
Argument	Description
`get`	Get a list of static routes.
`create`	Add a static route.
`replace`	Similar to the "Create Static Route" example given on the previous page. Creates a single static route by specifying its UUID; or a list of static routes. Overwrites existing routes.
`delete`	Delete all static routes.
`merge`	Merge the existing configuration list with a new list.

OSPF CONFIGURATION

Open Shortest Path First (OSPF) is a link-state routing protocol used to discover routes on a network. It is used to dynamically adjust routes on the Console Server so that subnets connected to different interfaces can reach each other by routing through the Console Server.

Support for OSPF configuration and WireGuard was added to the REST API and Config Shell at release 23.02.

Caution: Users are discouraged from editing OSPF configuration when it has been marked as managed by a Lighthouse. A warning message is displayed when an attempt is made to edit any configuration pushed down from Lighthouse through config shell. After being warned of the risk users may continue to edit configuration with a **managed_by** field set through config shell.



MANAGED CONFIGURATION ITEMS

Certain items in the configuration can contain an optional **managed_by** field. Configuration items that have the managed_by field set are considered to be "managed". The managed_by field is set by a managing entity such as lighthouse, when the network plan is being managed by a remote node.

The following features can have managed configuration:

- Firewall Zones
- Firewall Policies
- Routing OSPF
- WireGuard Tunnels

If a firewall zone, policy or WireGuard tunnel is managed, this does not affect sister contexts, for example, if the WireGuard tunnel is managed, any other WireGuard tunnels configured separately by the user are not managed. However, there is only one OSPF configuration file and users will need to bypass the **managed_by** field in config shell in order to edit the configuration.

NEW FIELDS IN REST API & CONFIG SHELL

REST API

The OSPF sub-object now has a number of new fields:

```
"services": {
  "routing": {
    "bgpd": {
      "enabled": true
    },
    "isisd": {
      "enabled": false
    },
    "ripd": {
      "enabled": true
    },
    "ospfd": {
      "enabled": false,
      "router_id": "",
      "redistribute_connected": false,
      "redistribute_static": false,
      "redistribute_kernel": false,
      "interfaces": [],
      "neighbors": [],
      "networks": []
    }
  }
}
```

CONFIG SHELL

The services/routing OSPF context has new fields similar to the REST API:

```
config(services/routing ospfd): show
Entity services/routing field ospfd
  enabled                              false
  redistribute_connected  false
  redistribute_static          false
  router_id                             ""
  interfaces (array)
  neighbors (array)
  networks (array)
```

Field	Condition	Definition
enabled	(true / false)	When set to true, the OSPF service is started.
redistribute_connected	(true / false)	If this option is enabled, any directly connected network routes will be broadcast to OSPF neighbours.
redistribute_static	(true / false)	Network routes can be statically defined (in OSPF, not the Linux Kernel) by editing the ospfd.conf file or through `vtysh`. If this option is enabled, redistribute_ routes broadcasts any static routes that are managed by OSPF.

redistribute_kernel	(true / false)	If this option is enabled, network routes that are configured in the Linux kernel via DHCP or static definition will be shared with OSPF neighbors.
router_id		The router id (RID) is a 32-bit number which must be expressed as a dotted quad (i.e. in the format A.B.C.D). The RID is used to identify the router. It must be unique within the OSPF network. The highest RID in the network will be used to determine which OSPF node is the designated router.

INTERFACES, NEIGHBORS AND NETWORKS.

There are a number of sub-objects under the ospfd context: interfaces, neighbors and networks.

INTERFACES CONTEXT

The services/routing OSPF interfaces context is an array in which each element holds the specific individual interface related parameters for OSPF. Each interface has the following fields:

```
Entity services/routing field ospfd interfaces 0
  auth_method     ""    (required)
  cost                     ""
  priority                ""
  name                    "" (required)
  non_broadcast   ""    (required)
  passive                 ""    (required)
```

Definitions of interface related parameters for OSPF:

Parameter	Definition
auth_method	The authentication method to use for communications on this interface. Should be one of 'no_auth', 'cleartext' or 'md5'. If authentication is enabled (i.e. not no_auth), one or multiple authentication keys can be configured depending on your authentication method chosen. • Cleartext authentication only needs one authentication key. • Md5 authentication can use multiple authentication keys, each of which needs a unique id.
cost	The link cost of the interface used in OSPF route calculations. It is normally auto-calculated, but can be specified manually in the range of 1 to 65535.
priority	The priority of a router on an OSPF interface mainly is used to determine the designated router/backup designated router (DR/BDR) for a network. OSPF forwards all messages to the designated router, reducing the amount of repetitive routing traffic on the network. The priority is in the range of 0 to 255. The default priority for each router is 1 unless specified. Selecting a priority of 0 makes the router unable to become a DR/BDR. The higher the priority, the higher chance a OSPF router has of winning the DR/BDR election.
name	The name of the interface these settings apply to. This should match the name of an interface on the device.
non_broadcast	May be true or false. If true, the interface will be marked as

	non broadcast for OSPF purposes. This would mean OSPF would not use multicast on this interface, and static neighbours would need to be defined.
passive	May be true or false. If true, the interface should be marked as passive for OSPF purposes. This would mean LSAs are not traded on this link.

NEIGHBORS CONTEXT

The services/routing OSPF neighbors context is an array where each element holds details about adjacent static neighbor devices. Neighbors must be specified for non-broadcast networks.

```
config(services/routing ospfd neighbors): add
config(services/routing ospfd neighbors 0): show
Entity services/routing field ospfd neighbors 0
address "" (required)
```

Where `address` is an IPv4 host address of the static neighbor.

NETWORKS CONTEXT

The services/routing OSPF networks context is an array where each element holds IP network configurations to enable the system OSPF service for:

```
config(services/routing ospfd networks): add
config(services/routing ospfd networks 0): show
Entity services/routing field ospfd networks 0
address_with_mask "" (required)
area "" (required)
```

Network Configuration	Definition
address_with_mask	An IPv4 network address with CIDR subnet mask to enable OSPF for (e.g. A.B.C.D/E). No host bits should be set.
area	An OSPF network can be divided into sub-domains or groupings called areas which limit the scope of route information distribution. We specify the area number/id we want the interface to be in. This can be an integer between 0.0.0.0 and 255.255.255.255 or can take a form similar to an IP address A.B.C.D. All routers inside an area must be a part of the same OSPF network and have the same area number/id to become OSPF neighbours.

INTERACTION WITH CONFIGURATION FILES

The first line of /etc/quagga/ospfd.conf controls whether the console server configuration system will overwrite the file with new content or keep custom user configuration. This supports customers who want to upload a custom configuration file for OSPF. If the first line contains only the text `! autogen`, the configuration system will overwrite the file, otherwise, the configuration system will have no effect.

To verify the OSPF configuration, the configuration file generated can be found in /etc/quagga/ospfd.conf:

```
! autogen
! This configuration file has been autogenerated. Any changes made
within
! will be overwritten. To stop this and allow for manual editing,
remove
! or change the first line of this file to something other than '!
```

```
autogen'.
! The behaviour can be reenabled by restoring the first line to this
or by
! completely removing this contents of this file.
!
interface wg-smf-1
ip ospf network non-broadcast
!
interface net1
!
router ospf
ospf router-id 0.0.0.1
log-adjacency-changes
redistribute connected
redistribute static
network 10.0.0.0/24 area 0.0.0.0
network 192.168.41.0/24 area 0.0.0.0
neighbor 10.0.0.1
!
line vty
!
```

CONFIRM OSPF NEIGHBOURS

Use the `vtysh` command line tool to see if OSPF neighbours have been discovered:

```
root@<device name>-q:~# vtysh -c 'show ip ospf neighbor'
Neighbor ID       Pri State Dead Time Address            Interface
RXmtL     RqstL        DBsmL
- 0 Attempt/DROther          33.007s 10.0.0.1 wg-smf-1:10.0.0.2
0 0 0
```

(Where `wg-smf-1` is a user-named interface).

WIREGUARD CONFIGURATION

WireGuard is an open source encrypted VPN solution; WireGuard configuration support was added to the REST API and Config Shell at release 23.8. WireGuard facilitates communication between two peer devices; in order to communicate with a peer, both devices must have a virtual WireGuard interface configured over the physical or virtual interface they are connected over.

Note: Users who have pre-existing configuration files for WireGuard will not have their configurations overwritten as the configurator will only modify those files if they are initially missing or are prefixed with a disclaimer that manual edits will be overwritten.

VIEWING A WIREGUARD CONFIGURATION

WireGuard installs the **wg** tool which can be used to control, configure and monitor WireGuard . Refer to the WireGuard online tools index page: index : wireguard-tools

Note: OpenGear does not own or operate the WireGuard tools web page and is not responsible for its content or maintenance. The link is provided only for the reader's convenience.

CONFIGURE WIREGUARD THROUGH CONFIG SHELL OR REST API

WireGuard is configured through Config Shell or REST API. The minimum configuration of WireGuard is shown in the following:

1. Provide a name for the interface (wg0 in the example below).
2. Set enabled.
3. Set the private_key of your WireGuard interface.
4. Add an address (at least one) for your WireGuard interface (10.0.0.1/24 in this case).
5. Add a peer with the following parameters: endpoint_address, endpoint_port, public_key.
6. Add an allowed_ip for your peer. At least one - this is the WireGuard address(es) (as it can also accept an address range) of the other interface to which you are connected.

For example:

```
config: wireguard
config(wireguard): add wg0
config(wireguard wg0): private_key
AGiZvFHY+r/dD0rHSKU5ZCrHNdLM0W/h29VxobxWgFo=
config(wireguard wg0): enabled true
config(wireguard wg0): addresses
config(wireguard wg0 addresses): add 10.0.0.1/24
config(wireguard wg0 addresses): up
config(wireguard wg0): peers
config(wireguard wg0 peers): add
```

```
config(wireguard wg0 peers 0): public_key
o+quB4sbUAG2hEGSPpMNTnO0YSaQTP7dD+Q4IVjiCW8=
config(wireguard wg0 peers 0): allowed_ips
config(wireguard wg0 peers 0 allowed_ips): add 10.0.0.2/32
config(wireguard wg0 peers 0 allowed_ips): up
config(wireguard wg0 peers 0): endpoint_address 192.168.1.2
config(wireguard wg0 peers 0): endpoint_port 51820
config(wireguard wg0 peers 0): up
config(wireguard wg0 peers): top
```

CONFIG SHELL WIREGUARD CONFIGURATION

The following shows a typical WireGuard configuration in Config Shell:

```
config: show wireguard wg0
Entity wireguard item wg0
  description ""
  enabled true
  mtu 1420
  name wg0
  port 51820
  private_key AGiZvFHY+r/dD0rHSKU5ZCrHNdLM0W/h29VxobxWgFo=
  public_key ""
  table ""
  addresses (array)
    0 10.0.0.1/24
  peers (array)
    0 (object)
      endpoint_address     192.168.1.2
```

```
      endpoint_port 51820
      keep_alive                    ""
      public_key o+quB4sbUAG2hEGSPpMNTnO0YSaQTP7dD+Q4IVjiCW8=
      allowed_ips (array)
          0 10.0.0.2/32
  post_down_hooks (array)
  post_up_hooks (array)
  pre_down_hooks (array)
  pre_up_hooks (array)
```

REST API WIREGUARD CONFIGURATION

The following shows a typical WireGuard configuration in Config Shell:

```
{
    "wireguards": [
        {
          "enabled": true,
          "post_down_hooks": [],
          "id": "wireguard_tunnels-1",
          "pre_up_hooks": [],
          "post_up_hooks": [],
          "private_key":
"AGiZvFHY+r/dD0rHSKU5ZCrHNdLM0W/h29VxobxWgFo=",
          "name": "wg0",
          "pre_down_hooks": [],
          "addresses": [
              "10.0.0.1/24"
          ],
```

```
            "peers": [
                {
                            "allowed_ips": [
                                  "10.0.0.2/32"
                            ],
                            "public_key":
"o+quB4sbUAG2hEGSPpMNTnO0YSaQTP7dD+Q4IVjiCW8=",
                            "endpoint_address": "192.168.1.2",
                            "endpoint_port": 51820
                }
            ]
        }
    ]
}
```

CONFIGURABLE WIREGUARD FIELDS

The WireGuard <interface-name> context holds the configuration for a WireGuard connection. The following fields can be configured:

WireGuard Field	Description
description	This can be any user text to describe the WireGuard interface.
enabled	Values may be **true** or **false**. When enabled, WireGuard will be started for this configuration.

mtu	Allows customization of the maximum transmission unit (MTU) for the local WireGuard interface. The range is 1280 - 1472 and if not set, WireGuard will use the internal default of 1420.
name	The name of the WireGuard interface used in the Linux kernel. Names must be unique, max 15 characters and only contain letters, numbers, hyphens or underscores.
port	The port the local instance of WireGuard will listen on. The range is 1 to 65535 and defaults to 51820.
private_key	The private key to use to authenticate the local WireGuard interface. This is obtained by running the wg genkey command.
public_key	The public key that corresponds your private key, which WireGuard peers will authenticate with. This is obtained by running the wg pubkey command.
table	The routing table for the WireGuard routes. Can be a table number, 'off' or 'auto'.

WIREGUARD CONTEXT SUB-OBJECTS

There are a number of sub-objects under the WireGuard context: addresses, peers and hooks.

ADDRESSES

The wireguard <interface-name> addresses context is a list that holds the IPv4 CIDR addresses of the local Wireguard interface. These are statically assigned when the WireGuard interface is brought up.

```
config: wireguard
config(wireguard): add wg0
config(wireguard wg0): addresses
config(wireguard wg0 addresses): add 10.0.0.1/24
```

PEERS

The following list defines the WireGuard settings for Wireguard-capable remote peers. Each peer has the following fields:

```
config(wireguard wg0 peers 0): show
Entity wireguard item wg0 field peers 0
    endpoint_address ""
    endpoint_port ""
    keep_alive ""
    public_key "" (required)
    allowed_ips (array) (required)
```

Peer Field	Description
endpoint_address	A reachable IP address or fully-qualified domain name for the remote peer with a WireGuard interface.
endpoint_port	The port number for which the WireGuard instance is listening on the remote peer.
keep_alive	Equivalent to PersistentKeepalive in the WireGuard config, this specifies how often the WireGuard interface must send a keep alive packet. This helps keep the routing entry alive for scenarios where the peer is behind a NAT.

public_key	The public key that will be accepted by the local WireGuard service if offered by a peer for the purpose of mutual authentication during a five step key exchange process.
allowed_ips -	A list which specifies the IP ranges for which a peer routes traffic. For multiple WireGuard interfaces on the same device, the addresses must not overlap. The IP addresses specified here are the addresses of the peer's WireGuard interface(s) - this is where the peer "routes traffic". These are specified as IPv4 addresses in a.b.c.d/<cidr_mask> format.

HOOKS

WireGuard allows for commands to be executed before/after the interface is brought up/down. These can be specified in the following array fields:

Note: Each field is an array of strings that correspond to commands to be executed.

Hook	Description
pre_up_hooks	Run a command before the interface is brought up (optional).
post_up_hooks	Run a command after the interface is brought up (optional).
pre_down_ hooks	Run a command before the interface is brought down (optional).

post_down_hooks	Run a command after the interface is brought down (optional).

ADDING A WIREGUARD INTERFACE TO A FIREWALL ZONE

The WireGuard interface can be added to a firewall zone as in the following example:

```
Entity firewall/zone item zone
description "" (required)
label "" (required)
masquerade "" (required)
name zone
permit_all_traffic "" (required)
address_filters (array)
custom_rules (array)
physifs (array)
port_forwarding_rules (array)
wireguards (array)
```

SSH

CONFIGURE > SERVICES > SSH

To modify the properties of the port used for connecting to serial consoles via SSH, navigate to **CONFIGURE > SERVICES > SSH** .

The following table gives the definitions of the configurable SSH properties.

Parameter	Definition
Serial Port Delimiter	The delimiting character used to separate the username with port selection information. The default delimiter is a plus sign (+). For example, username+port@address.
Port Number for Direct SSH Links	If SSH is configured to be reachable on a non-standard port, the Direct SSH links on the serial ports page will use this port number.
Max Startups Start	The number of unauthenticated connections before they are refused.
Max Startups Rate	This is the percentage of unauthenticated connections refused. This percentage is a probability that increases linearly until the unauthenticated connections reach full.
Max Startups Full	The number of unauthenticated connections allowed.

UNAUTHENTICATED ACCESS TO SERIAL PORTS

For information about Unauthenticated Access to Serial Ports, see "Unauthenticated SSH to Serial Ports" below.

UNAUTHENTICATED SSH TO SERIAL PORTS

Configure > Services > SSH

The Unauthenticated SSH Access feature provides the option to access console ports (using TCP high ports) by establishing per-port SSH connection between a

console and serial ports at a remote device. This allows a single step log-in and avoids the necessity for two log-ins to reach a remote end device within secure, closed networks.

Usually, you would need to authenticate on the Opengear appliance, followed by any log in to a device you are connecting to via the serial port.

When unauthenticated access is enabled SSH is available to all serial ports on the device without requiring a password.

Note: Unauthenticated access can be used with or without IP aliases for serial ports.

Caution: For security, **Unauthenticated SSH** should only be used when operating within a trusted, closed network, for example within a lab. There is a security risk in allowing any kind of unauthenticated access to serial ports and any terminals connected to them.

ENABLE UNAUTHENTICATED SSH

Authenticated or Unauthenticated access is determined via a global configuration option. Unauthenticated access to individual ports is achieved by command such as `ssh -p 300X user@<IP>`.

ENABLE SSH

Note: This feature may be enabled using the default settings without the need for configuration.

1. Open the SSH form, **Configure > Services > SSH > SSH (form)**.
2. Complete the SSH form (if this is the first time Unauthenticated SSH has been used), a description of the input data is provided at "Properties and Settings" on page 265 in this topic.
3. When required, enable the Unauthenticated SSH feature by clicking the **Enabled** button.

   Note: Unauthenticated access to all serial ports will be available through SSH on TCP port 3000+ or Serial Port IP aliases.

ENABLE/DISABLE

Enabling or disabling this feature is done in the user interface.

To **enable** the feature click on the **Enabled** button then click the **Apply** button. The feature is enabled immediately and a pop-up will confirm that the feature is enabled.

Note: Clicking the **Apply** button saves any changes you have made to the SSH form. A Details Saved banner confirms that the changes have been saved.

To **disable** the feature click on the **Disabled** button then click the **Apply** button. There is no confirmation pop-up when the feature is disabled.

CONNECTING DIRECTLY TO SERIAL PORTS

For ports that have been configured with the SSH access service, you can connect directly to a port and start a session, bypassing the chooser, by using one of the conventions described in the following:

Convention	Example

Use a network client to connect to the service network Base Port + serial port number.	`# SSH to serial port 1 by TCP port`
	`ssh -p 3001 -l operator 70.33.235.190`
	
	In this example, the SSH base port is TCP port 3000, so SSH to TCP port 3001 directly connects you to serial port 1

SSH to the Opengear node, log in adding +portXX to your username (e.g. root+port01 or operator+port01).	`# SSH to serial port 1 by port name` `ssh -l operator+port01 70.33.235.190`
SSH to the Opengear node, log in adding the +port-label to your username (e.g. root+Router or operator+Router).	`# SSH to serial port labelled Router` `ssh -l operator+Router 70.33.235.190`

Note: For additional reading on connecting to serial ports see: Communicating with serial port connected devices

Note: Serial ports in the Local Console and Disabled ports modes are not available for SSH connection.

FEATURE PERSIST

If the node has an active console session after closing pmshell, connecting to the node again will resume the session and you are not prompted for the node password.

PROPERTIES AND SETTINGS

Property	Definition/Range
Serial Port Delimiter	A character that separates the User name and port selection information. The default value is the + character. *Default is '+', maximum length is 1.* *The prohibited characters are '\', '"', '`', ' ', '=' and '#'.* **Source: schema**

	required ssh_delimiter: string (default = "+"; minimum = 1; maximum = 1; validator = ("ssh_ url_delimiter")),
	Source: validator
	if (strlen(v) != 1) valid = 0;
	else if (v[0] == '\'') valid = 0;
	else if (v[0] == '"') valid = 0;
	else if (v[0] == '`') valid = 0;
	else if (v[0] == ' ') valid = 0; // breaks sshd_config
	else if (v[0] == '=') valid = 0; // breaks sshd_config
	else if (v[0] == '#') valid = 0; // breaks sshd_config
	else if (!isprint(v[0])) valid = 0;
	else {
	valid = 1;
	}

Port Number for Direct SSH Links	This port number will be used for direct SSH links on the serial ports page. Set this option if you have configured SSH to be reachable on a non-standard port.
Max Startups Start	The number of connections pending authentication before new connections *begin* to

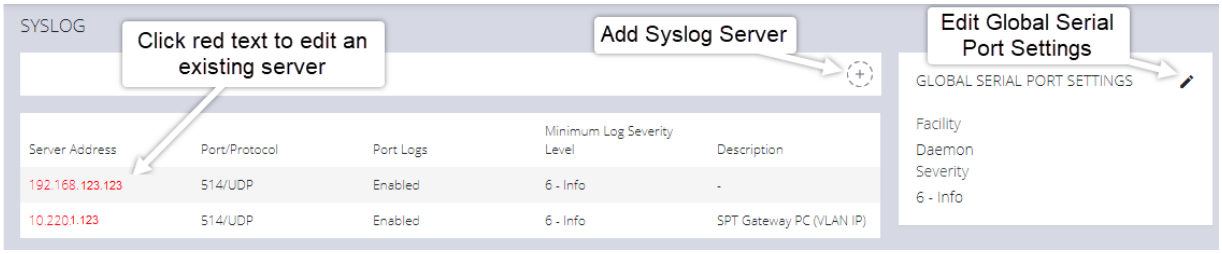
	be refused. *Required start: int (minimum = 1; default = 10)*
Max Startups Full	The number of connections pending authentication before *all* new connections are refused. *Required full: int (minimum = 1; default = 100)*
Max Startups Rate	This is the percentage rate at which new connections are refused once the Max Startups value is reached. The rate is increased to 100% at Max Startup Full. *Required rate: int (minimum = 1; maximum = 100; default = 30),* *The rate at which connections are refused randomly begins at max startup rate and increases linearly until the number of connections pending authentication reach max startups full, in which case 100% of new connections are refused.*
Unauthenticated Access to Serial Ports	This is the feature Enable/Disable button.

SYSLOG

CONFIGURE > SERVICES > Syslog

Administrative users can specify multiple external servers to which the Syslog can be exported via TCP or UDP. There is a drop-down on each serial port to enable the logging and to define the “scope” of logging.

The Syslog page lists any previously added external syslog servers.



ADD A NEW SYSLOG SERVER

Note: The combination of server address, protocol and port should be unique. There can be no duplicates. However, the same server could be used if the other entry is an IPv6 address to the same Syslog server.

Use the following procedure to add a new Syslog Server.

1. Navigate to **CONFIGURE > SERVICES > Syslog**.
2. Click the **Add Syslog Server** button. The **Add Syslog Server** form opens.
3. In the **Description** field, add a suitable description that will help to identify the new server.
4. Enter the **Server Address**.
5. Click the **Protocol** switch to select either **UDP** or **TCP**.

6. Enter the correct **Port**. If no port is entered, UDP defaults to port 514 and TCP defaults to 601.
7. From the drop-down list, select the required severity level to be logged, eight levels of log severity are supported.
8. Click **Add** to complete the process.

GLOBAL SERIAL PORT SETTINGS

Global Serial Port Settings will define the Facility used and the Severity of all Syslog serial port activity sent from this node. There are two setting functions, Facility, and Severity. From the drop-down menus, select the preferred Facility and Severity as required.

GLOBAL SERIAL PORT SETTINGS TAB - FIELD DEFINITIONS

Configure > Services > Syslog > Global Serial Port Settings

Field	Definition
Description	Unique, familiar text description or name given to this syslog server that users will recognize.
Server Address	The IP address of the syslog server you are using for logging.
Protocol	Click to select the required protocol for data transmission to the syslog server.

Port	The Syslog Server IP address.
Minimum Log Severity Level	Log entries with a value equal or greater than the level specified are sent to the server.
Send Serial Port Logs	Click to enable serial port logging.
Add Button	Click to initiate the syslog, wait for confirmation banner.

SYSLOG FACILITY DEFINITIONS

Facility	Definition
Kern	Kernel messages
User	User-level messages
Mail	Mail system
Daemon	System daemons
Auth	Security/authentication messages
Syslog	Messages generated internally by syslogd
lpr	Line printer subsystem
News	Network news subsystem
uucp	UUCP subsystem
Cron	Clock daemon
Authpriv	Security/authentication messages
ftp	FTP daemon
Local	Locally used facilities

SYSLOG SEVERITY DEFINITIONS

Severity	Definition
0- Emergency	System is unusable.
1 - Alert	Action must be taken immediately.
2 - Critical	Critical conditions.
3 - Error	Error conditions.
4 - Warning	Warning conditions.
5 - Notice	Normal but significant conditions.
6 - Info	Informational messages
7- Debug	Debug-level messages

EDIT OR DELETE AN EXISTING SYSLOG SERVER

To edit an existing syslog server, click the hyperlinked **Red Text** server name in the server list (see the Syslog page image on the previous page). Make the required changes, then click the **Submit** button.

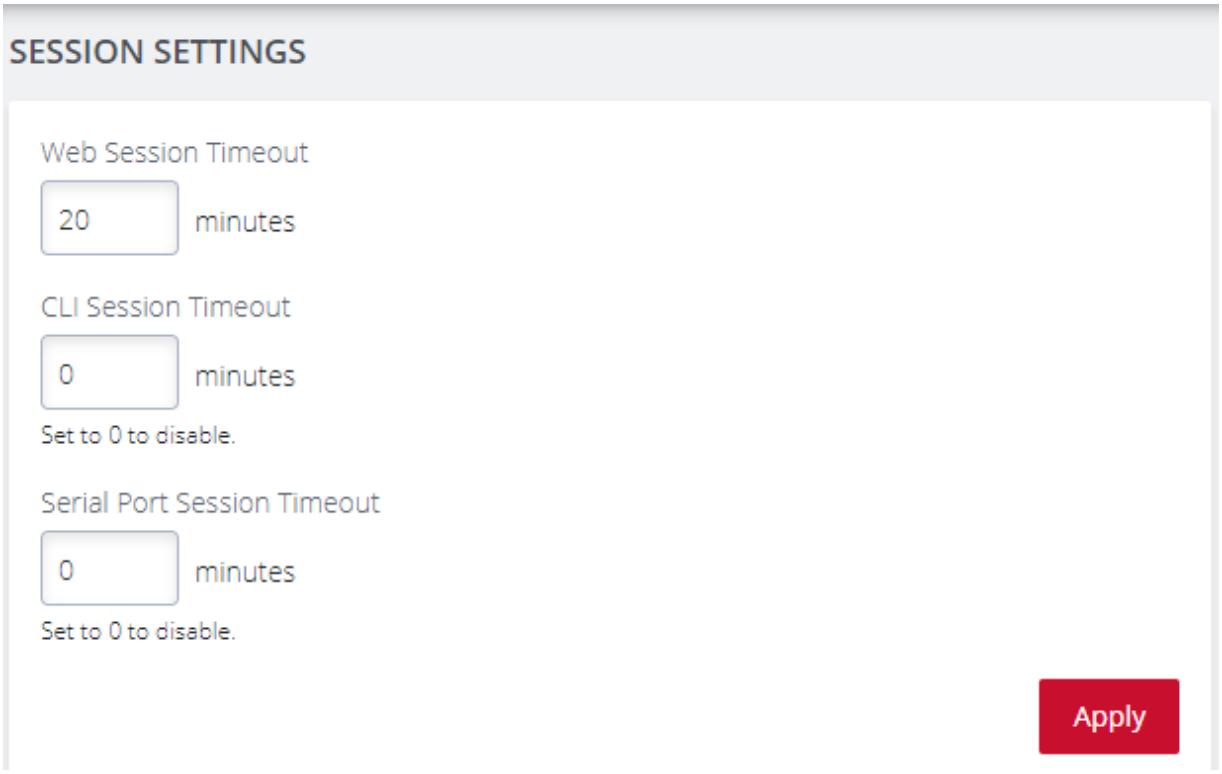
Delete a server by clicking the Delete icon at the top-right of the **Edit Syslog Server** page.

SESSION SETTINGS

SETTINGS > SERVICES > Session Settings

Use **Session Settings** to set timeouts for console sessions where the users have been idle for a specified time. At timeout, the user's web, CLI or Serial Port sessions are terminated, thus excluding authorized users with physical access to the node that has been left connected.

To set the timeouts for Web, CLI or Serial Port sessions settings, navigate to the **SETTINGS > Services > Session Settings** page.



- **Web Session Timeout**: Set the timeout from 1 to 1440 minutes.
- **CLI Session Timeout:** Set the timeout from 1 to 1440 minutes or set it to 0 to disable the timeout. Changes take effect the next time at the next login via the CLI.
- **Serial Port Session Timeout:** Set the timeout from 1 to 1440 minutes or set it to 0 to disable the timeout.

Click the **Apply** button to save the settings.

The new session timeout will take immediate effect on all pmshell sessions, including ones in use.

FIREWALL

CONFIGURE > FIREWALL

In the **CONFIGURE > FIREWALL** menu you can configure:

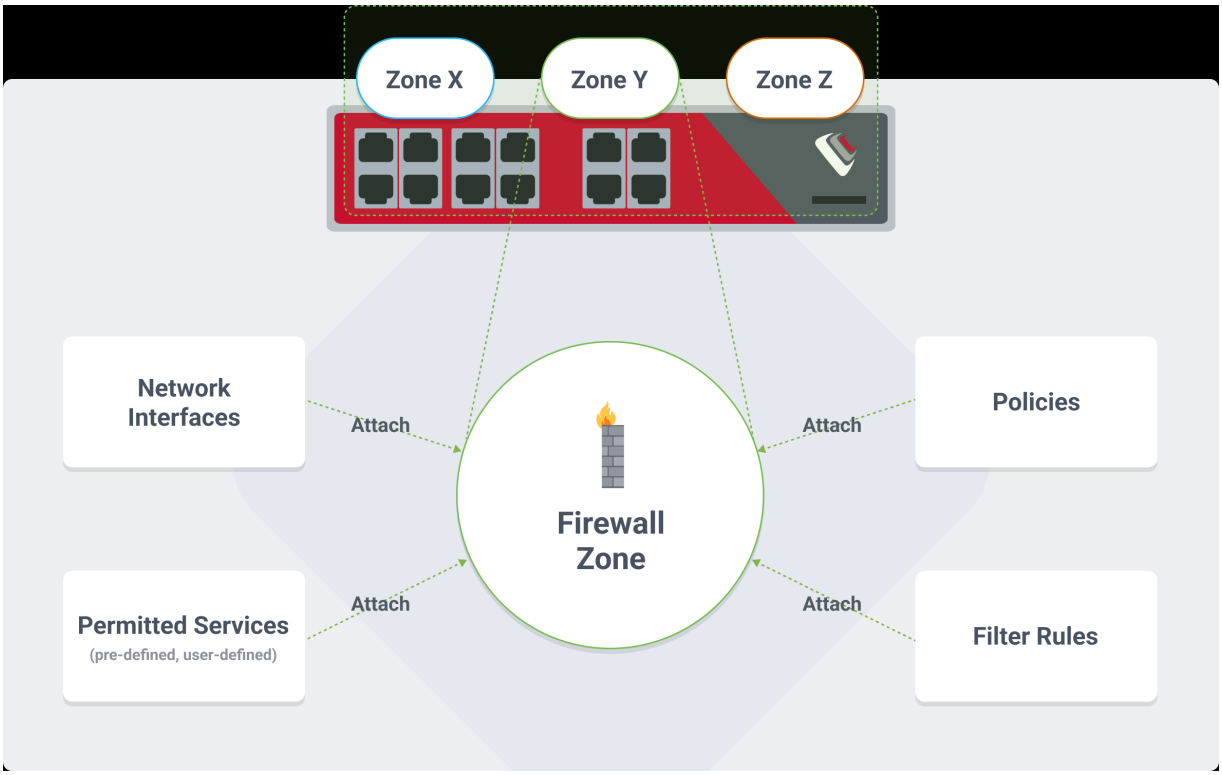
- **Firewall Management**
- **Interzone Policies**
- **Services**

FIREWALL GUIDE

INTRODUCTION

Opengear firmware is equipped with a powerful firewall stack based on leading open source firewalld and nftables tools. The default firewall rule set is configured with a default-deny policy.

The firewall is based on the concept of configurable Zones. Zones enable operators to create multiple “firewall segments” per node and attach network interface(s), services, filtering policies and filtering rules to the zones.



Note: To access services on the device, a user must have both access through the firewall and the appropriate authorization, e.g. via a local user account or remote AAA.

There are several kinds of rules and policies that may be applied to Zones.

Firewall Rules

- Permitted Services Rules allow access to Services for requests arriving on interfaces in the Zone – Services are configurable collections of TCP/UDP port or ports (e.g. TCP port 443 is the device's HTTPS service for WebUI and REST API access). There are pre-defined services, devices also support user-defined services.
- Custom Rules allow the full flexibility of the firewalld rich rule syntax for fine-grained access control and advanced applications.

Firewall Policies

- Interzone Policies control how Zones may forward traffic between each other – by default Zones may not forward between each other (note that interfaces in the same Zone may always forward between themselves).
- Port Forwarding Rules use destination NAT (DNAT) requests arriving on interfaces in the Zone to an external Target IP/Port, e.g. a web server running on another host
- Additionally, you can apply source NAT (SNAT) to traffic going out of a Zone by checking the Masquerade Traffic option.

EXAMPLE WEBUI CONFIGURATION

The following examples use Permitted Services Rules and Custom Rules features

> **Note:**Some aspects of the UI may change in future releases.

Example 1: Disallow WAN Zone access to HTTPS

The default configuration is to allow HTTPS (i.e. the WebUI & API) on the WAN Zone. To disallow this:

> **Note:**Ensure you are accessing the device via an interface outside the WAN Zone (e.g. NET2 which is the LAN Zone by default) otherwise you will lock yourself out.

1. Login to the WebUI as an Administrator user.
2. Select **CONFIGURE > FIREWALL > Management**.
3. Click **WAN** then **Edit Zone**.
4. Scroll down to **Permitted Services**.
5. In the right-hand column, click – to remove **https** service.
6. Any service in the right-hand column allows everyone access to this service from this zone.
7. Click **Apply**.

Example 2: Permit access to WAN Zone HTTPS from a trusted source network only

When a service is permitted using a Permitted Services Rule, connections to the service in that Zone are permitted regardless of the originating network the connection is coming from. To disallow connections from all but a trusted source network, use Custom Rules (examples below) instead.

In this example, HTTPS connections from the 10.12.34.0/24 network to the Operation Manager's WAN Zone are permitted, other HTTPS connections on the WAN Zone are disallowed.

Note: Ensure you are accessing the device via an interface outside the WAN Zone (e.g. NET2 which is the LAN Zone by default) or from the trusted source network, otherwise you will lock yourself out.

1. Login to the WebUI as an Administrator user.
2. Select **CONFIGURE > FIREWALL > Management**.
3. Click **WAN** then **Manage Custom Rules**.
4. Click **Add Custom Rule**.
5. In **Description** enter: *Trusted HTTPS*.
6. In **Rule Content** enter:
   rule family=ipv4 source address=10.12.34.0/24 service name=https accept

   Note: This is supported via firewalld 'rich-rules' option.
7. Click **Apply**.
8. Follow the steps in Example 1 above to remove the HTTPS Permitted Service.

Note: It is not recommended to mix firewall configurations between the UI (WebUI/CLI) and firewalld commands (firewall-cmd) from Linux shell. Commands may be overwritten. Recommended to use either WebUI or CLI for all supported functionality instead of firewall-cmd

CUSTOM RULES (FIREWALLD “RICH-RULES”)

This feature enables users to define fine-grained control of services inside a zone. Users can apply custom filter rules to traffic in a firewall zone based on Layer2 Ethernet MAC, L3 IP fields, layer 4 ports, pre-defined services. Actions to permit, deny, drop the defined packets can be included in the rule. Logging facility is also provided via custom rules.

The following sections provide examples and many sample templates that users can directly use in WebUI or CLI in the rich-rules field/section

Custom Rules Examples:

Example 1: Filter (drop) specific IPv4 source address

```
rule family="ipv4" source address="34.34.36.36" drop
```

Example 2: Permit specific source subnet and list of address

```
rule family="ipv4" source address="34.34.36.0/24" accept
```

Example 3: Permit Specific Service (HTTPS) from a specific source subnet:

```
rule family="ipv4" source address="10.0.0.0/16" service name="https"
accept
```

Example 4: Drop Specific Service (HTTP)

```
rule family="ipv4" service name="http" drop
```

Example 5: Permit specific source subnet and log connection attempts

```
rule family="ipv4" source address="10.0.0.0/16" accept log
```

Example 6: Permit IPv6 packets with source address, TCP port number 4000. Log the packets

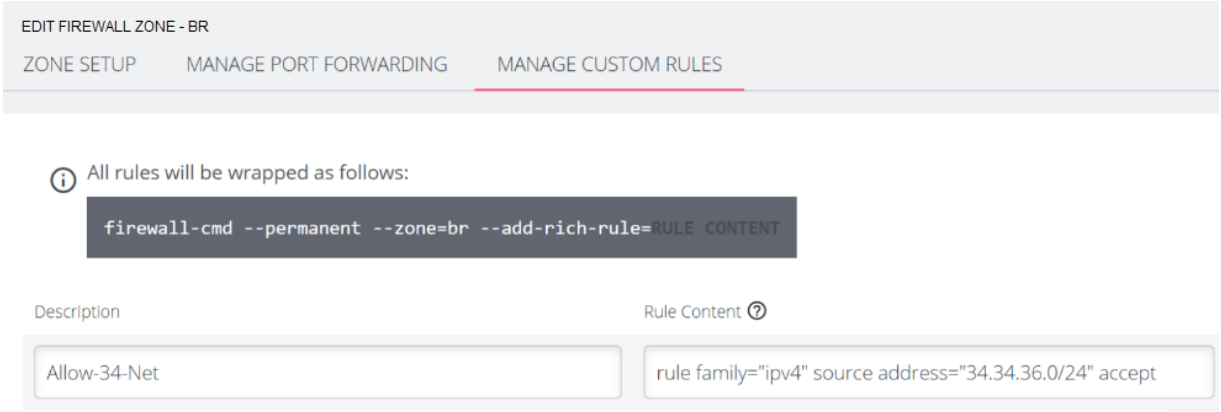
```
rule family="ipv6" source address="1:2:3:4:6::" port port=4000
protocol=tcp accept log
```

Example 7: Permit IPv6 packets with source address, only TCP protocol, from all TCP ports. Log the packets

```
rule family="ipv6" source address="1:2:3:4:6::" protocol value="tcp"
accept log
```

USEFUL TEMPLATES FOR USE IN WEBUI OR CLI

In WebUI:



In ogcli:

ogcli replace firewall/zones << 'END'

firewall_zones[0].custom_rules[0].description="allow rule"

firewall_zones[0].custom_rules[0].rule_content="rule family='ipv4' source address='192.168.67.101/32' service name='telnet' accept"

…

END

SAMPLE RICH RULES TEMPLATES

```
1. rule family="ipv4" source address="<user-to-fill>" accept|drop|reject
```

```
2. rule family="ipv4" destination address="<user-to-fill>" accept|drop|reject
```

```
3. rule family="ipv4" destination address="<user-to-fill>" accept|drop|reject
```

```
4. rule family="ipv4" source address="<user-to-fill>" accept|drop|reject
```

```
5. rule family="ipv4" source address="<user-to-fill>" destination
address="<user-to-fill>" accept|reject|drop log
```

```
6. rule family="ipv4" source address="<user-to-fill>" service name="<user-to-
fill>"" accept|reject|drop
```

```
7. rule family="ipv4" source address="<user-to-fill>" destination
address="<user-to-fill>" accept|reject|drop log
```

```
8. rule family="ipv4" source address="<user-to-fill>" destination
address="<user-to-fill>" accept|reject|drop log
```

```
9. rule family="ipv4" source address="<user-to-fill>" port port=<usr-to-fill>
protocol=tcp|udp accept|reject|drop
```

```
10. rule family="ipv4" source address="<user-to-fill>" protocol
value="tcp|udp" accept|reject|drop
```

Note: Ordering of rules is important. See Firewalld Rich Rules Explained.

In the Template:

- Choose one of the actions accept|reject|drop [Drop action does not send any response back to source, reject does].
- For protocol value, tcp and udp examples are given in template, but many other choices are available.
- For values, source address as example, replace <user-to-fill> with the address. Address can be with or without subnet.

FIREWALL MANAGEMENT

CONFIGURE > FIREWALL > Management

Navigate to the Firewall Management page, **CONFIGURE > FIREWALL > Management**, from here you can:

- Add a new firewall zone.
- Add a firewall service.
- Edit a firewall zone - manage the zone setup.
- Manage port forwarding.
- Manage custom rules for firewalls.

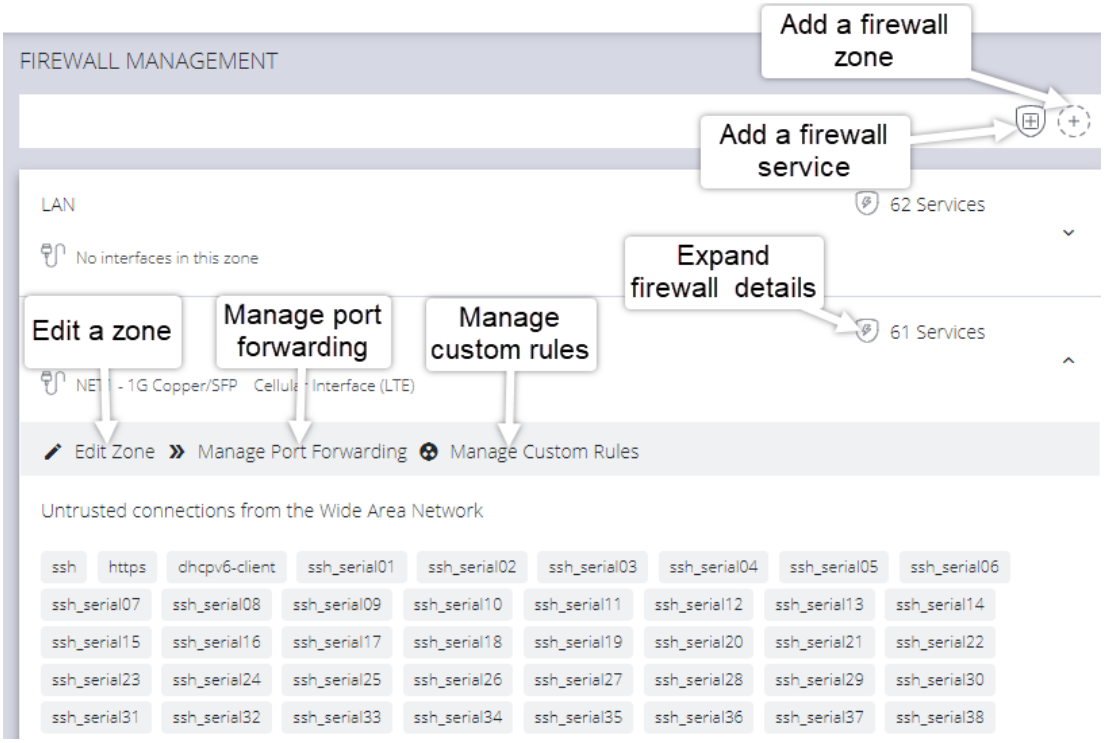


Figure: Firewall Management main page

FIREWALL ZONE SETTINGS

To change firewall management settings navigate to **CONFIGURE > FIREWALL > Management**.

> **Note:**The application of any custom rules will result in **Permit All Traffic** being enabled in a zone.

You can inspect details of any zone by clicking the **Expand** icon to the right of the zone. Once expanded, you can click **Edit Zone** to change settings for a particular zone.

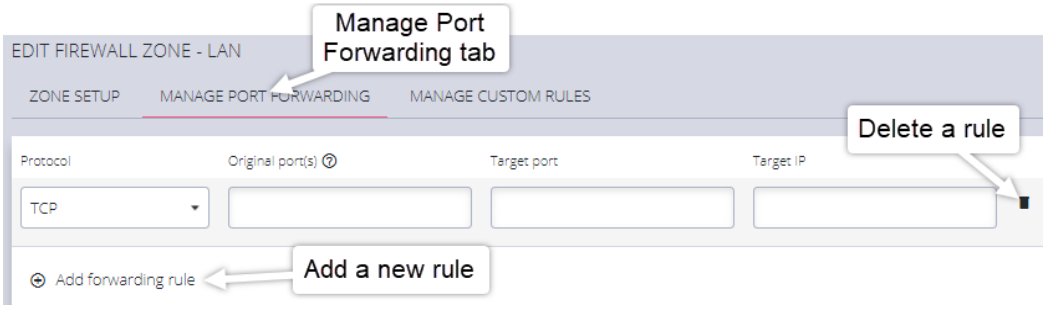
The **Edit Zone** page has three tabs. The **ZONE SETUP** page allows you to:

- Modify the Name of the zone.
- Add a Description for this zone.
- Permit all Traffic.
- Masquerade Traffic.
- Select Physical Interfaces.
- Manage Permitted Services by clicking on Plus or Minus next to each.

> **Tip:** You can use the **Filter Interfaces** and **Filter Available Services** text boxes to limit the list content that is displayed.

PORT FORWARDING

The **MANAGE PORT FORWARDING** tab allows you to add, edit, and delete forwarding rules for the particular zone you are editing.

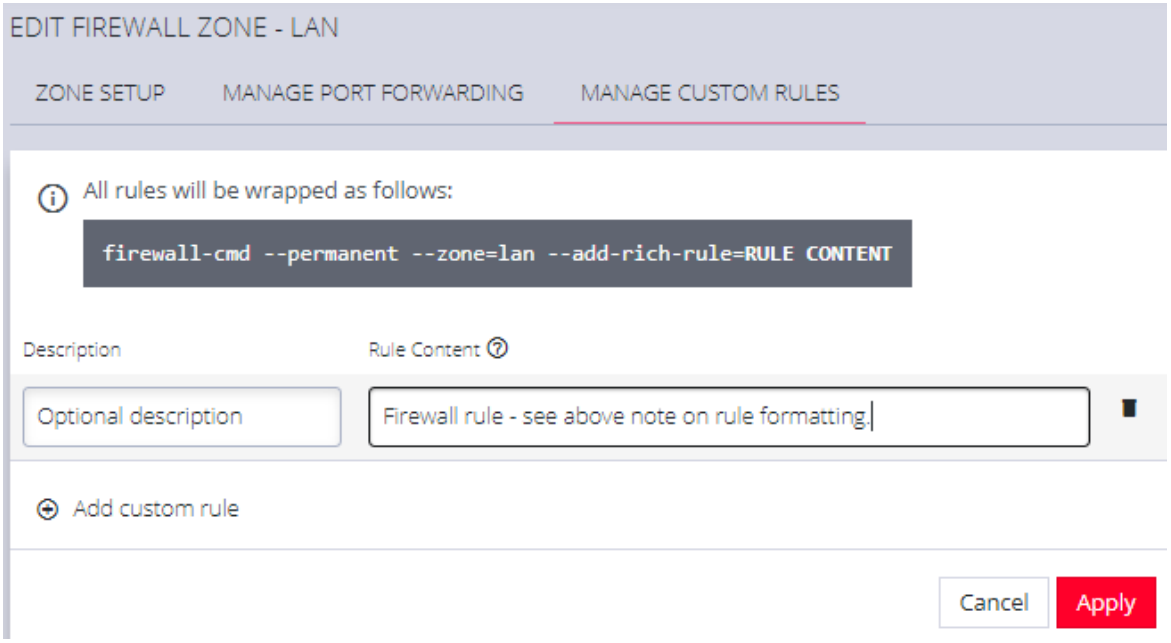


MANAGE CUSTOM RULES

> **Note:**The application of any custom rules will result in **Permit All Traffic** being enabled in a zone.

The third tab, **MANAGE CUSTOM RULES**, allows you to add, edit , and delete custom firewall rules for the zone you are editing. These custom rules continue to exist after reboots, upgrades, and power cycles.

These rules are prioritized by the order they are added.



To add a new custom rule:

1. Click **Add custom rule**.
2. Enter an optional description for this rule.
3. Enter the rule content, custom rule content formatted with firewall-cmd syntax.
4. Click **Apply**.

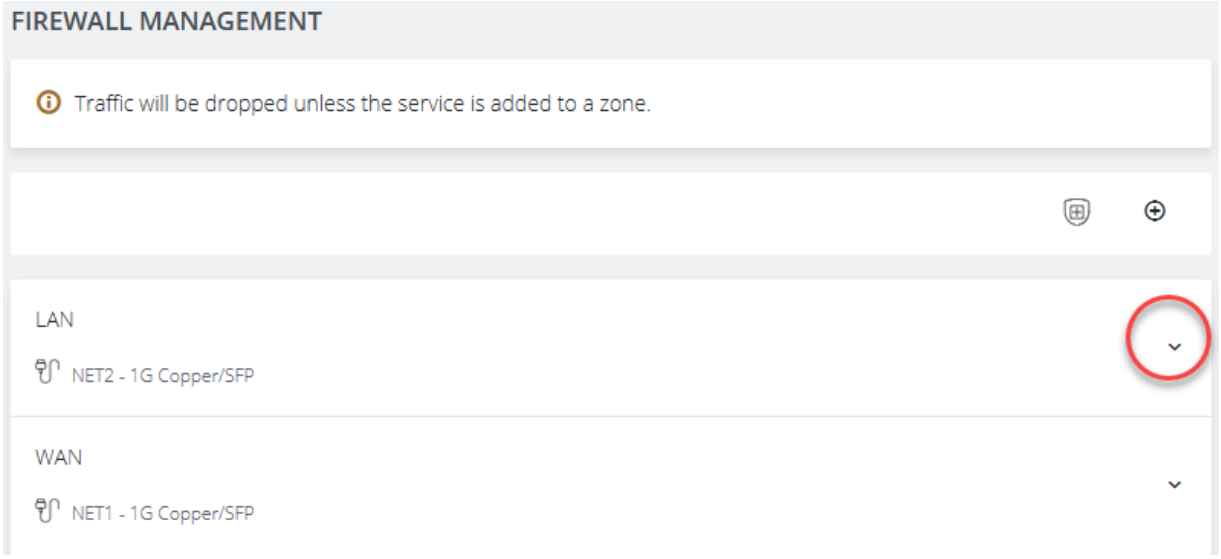
Note: All rules will be wrapped as follows:
firewall-cmd --permanent --zone=lan --add-rich-rule=RULE CONTENT

FIREWALL - SOURCE ADDRESS FILTERING

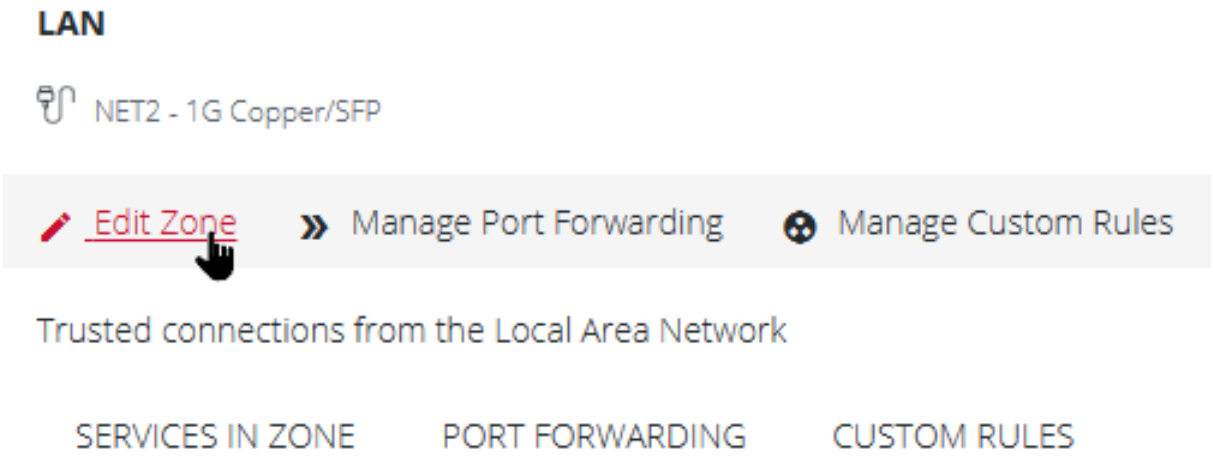
Source address filtering provides an interface by which users can permit access to services (for example, SSH, HTTPS, SNMP) on a device from specific source addresses.

This feature removes generic or global permitted services within firewall zones, and instead allows users to permit a service on a specified source address (or address range) within the firewall zone. Source address filters configured in a zone apply to all the interfaces within that zone.

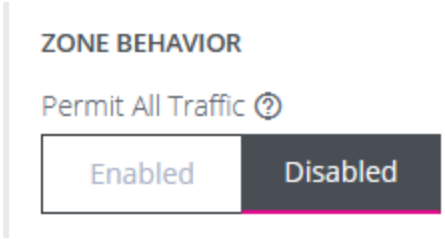
To access the feature, navigate to the **Configure > Firewall > Management** page through the WebUI then select the current source address filter configuration under the **services in zone** tab for each zone.



To add a source address filter for a zone, select the **edit zone** option under the desired zone, which opens the **edit zone page** where source address filters can be configured.



You can choose to enable permit all traffic, which will permit all traffic in the zone (unless there is a custom rule configured overwriting this behavior).



If the permit all traffic option is disabled, you will have the option to configure permitted services for any allowed source address. Permitted services can be added or removed from each source address filter under the "Services" field.

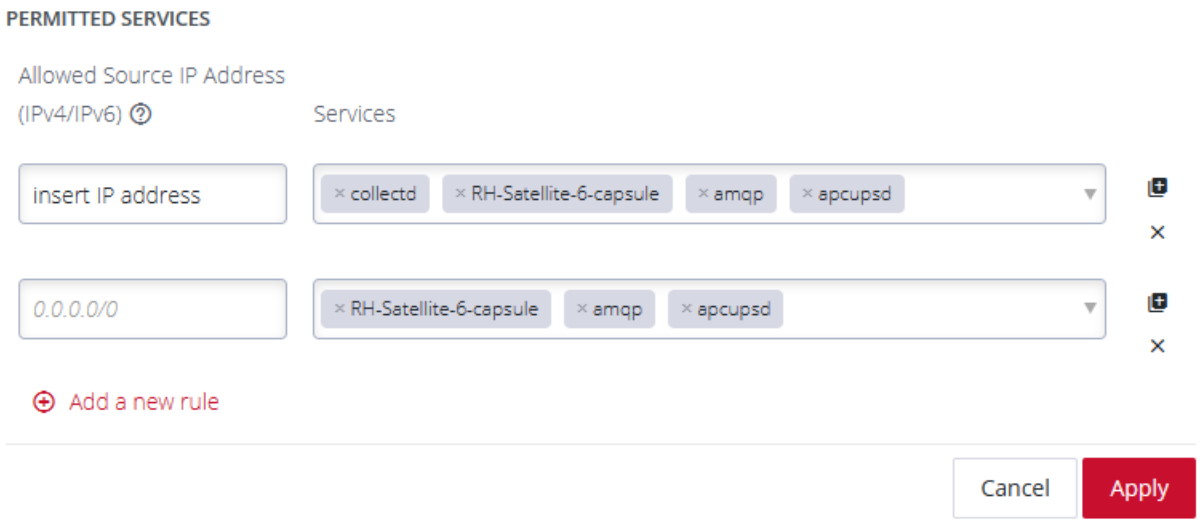
Source address filters can be added, duplicated or deleted by using the buttons below and to the right of the filter. Any new changes to the source address filters can be seen under the **services in zone** tab for each zone on the main firewall management page.

FIREWALL SOURCE ADDRESS BULK SERVICES

Configure > FIREWALL > Management > New Firewall Zone

PERMITTED SERVICES

The firewall source ip field allows you to assign permitted services to specified source ip addresses in bulk rather than needing individual rich rules to add each specific service. This change allows you to easily target specific IP Addresses with permitted services. Enter the target IP address, select services from the drop-down list and click **Apply**.



FIREWALL EGRESS FILTERING

Firewall egress filtering may be used to allow or deny traffic leaving a device. This feature allows you to create firewall egress rules, which govern outgoing traffic leaving the device.

Firewall egress filtering extends the firewall/policies endpoint, allowing customization over both incoming (ingress) and outgoing (egress) traffic, thus allowing greater control of the device's security.

The feature allows you to:

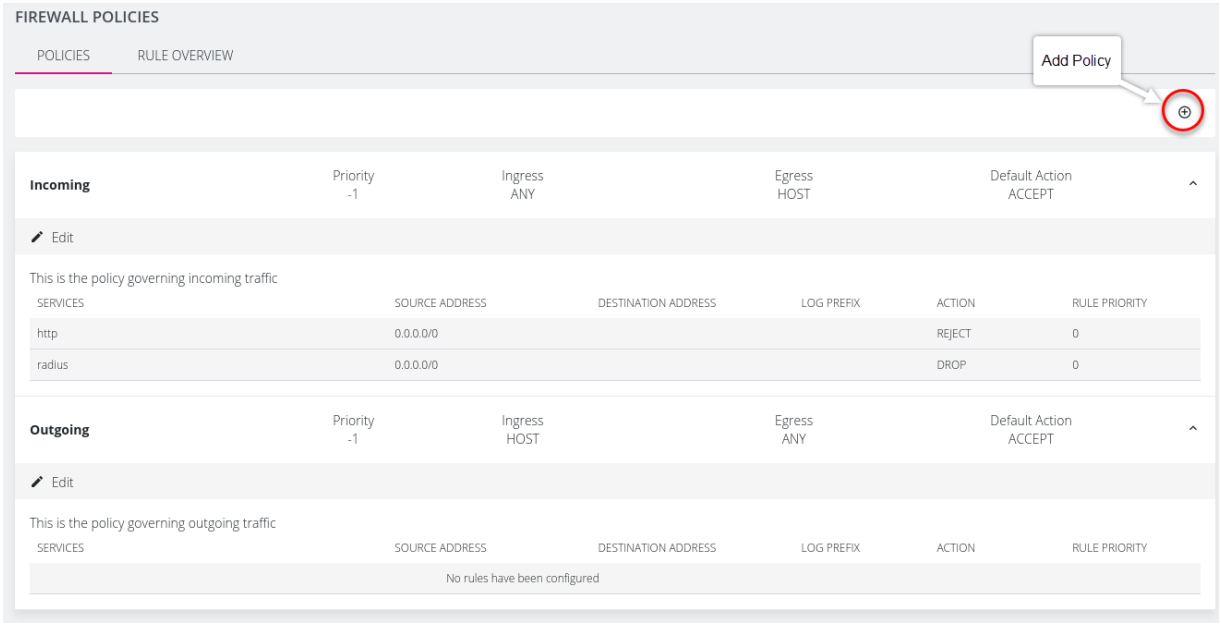
- Change the default behavior of a firewall policy so it can accept or deny traffic moving between zones.

- Create, edit and delete firewall policy rules which allow or block specific service traffic based on IP addresses.
- Configure firewall policy rules through ogcli, config shell or the WebUI.
- Display and inspect rules in a single location in the WebUI.
- Create symbolic zones HOST and ANY which allow the creation of catch-all firewall policies affecting traffic incoming and outgoing all zones.

CREATING EGRESS POLICIES ON THE WEBUI

Configure > FIREWALL > Policies

New policies or edits of existing policies are done from the Firewall Policies page. Navigate to **Configure > Firewall > Policies**, there is now an overview for firewall policies created on the device, as well as an overview page showing firewall policy rules created. To view firewall policy rules, click the drop-down arrow to the right of any policy row.



CREATING NEW EGRESS POLICIES

New policies are created by first clicking on the **Add Policy** button at the top-right of the **Firewall Policies** page of the WebUI. New policies can have a user-defined default action, either ACCEPT, CONTINUE, DROP, or REJECT, which describes how traffic moving through the ingress and egress zones will be treated. The ingress and egress zones may be configured as custom zones on the device through the firewall/zone endpoint, or can be symbolic (ANY/HOST) which represent traffic on all interfaces and the host device itself respectively. These default actions are described below.

Default Action	Outcome
ACCEPT	All packets flowing between ingress and egress zones are accepted by default.
REJECT	Rejects every packet (a message warns that the connection was rejected and that packets will not be allowed through): `ssh: connect to host 10.236.3.7 port 22: Connection refused`
DROP	Drops every packet (users will not get a message, the connection will hang).
CONTINUE	Ongoing packets will be subject to rules in following policies and zones.

CREATE A NEW FIREWALL POLICY

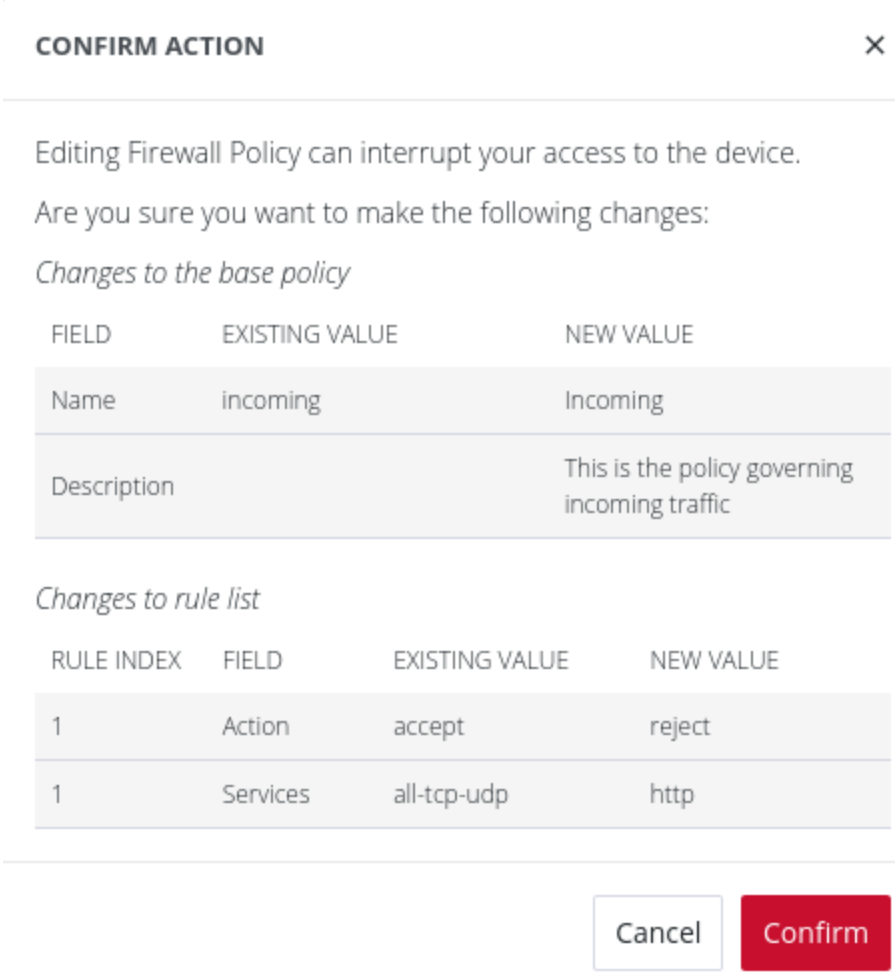
1. Click on the **Add Policy** button at the top-right of the **Firewall Policies** page of the WebUI.
2. Complete the **Name**, **Description**, **Default Action** and **Policy Priority** inputs of the New Policy.

Note: Policy Priority - Policies with negative values are applied before any filtering rules in zones. Policies with positive values are applied after filtering rules in zones. A priority of 0 (zero) cannot be applied.

3. Select the required Ingress and or Egress zones.
4. Click on the **Add New Rule** button and complete the information; Source and Destination address, also Log Prefix are optional.
5. Click **Apply**. The new rule is instated.

EDITING POLICIES OR RULES

Rules associated with a policy can be edited. When saving their changes after editing, you are prompted to double check their changes using the **Confirm Action** window, which presents an overview of the policy changes.



Note: Editing a firewall policy or rule may interrupt access to the device.

CONFIGURE EGRESS POLICIES IN THE CONFIG SHELL

Firewall policies may be created through config shell an example is given below:

```
config: firewall/policy
config(firewall/policy): add incoming
config(firewall/policy incoming): default_action accept
config(firewall/policy incoming): egress_zones
config(firewall/policy incoming egress_zones): add host
config(firewall/policy incoming egress_zones): up
```

```
config(firewall/policy incoming): ingress_zones
config(firewall/policy incoming ingress_zones): add any
config(firewall/policy incoming ingress_zones): up
config(firewall/policy incoming): show
Entity firewall/policy item incoming
   default_action accept    *
   description ""
   name                                    incoming
   priority                                    -1
   egress_zones (array)
      0 host *
   ingress_zones (array)
      0 any *
   rules (array)
```

Policy Configurable Fields	
default_action	The default action that is applied to packets that don't match any rule.
priority	The priority of the policy dictates when it is applied compared to other policies and zones. Policies with negative priorities are applied before rules in zones; policies with positive priorities are applied after. A priority of 0 is reserved for Rules and is not used for policies. The default value is -1.
egress_zones	Traffic directed to the egress zones will be subject to this policy. This was pre-existing but has been expanded to include options for ANY/HOST.

ingress_zones	Traffic originating from the ingress zones will be subject to this policy. This was pre-existing but has been expanded to include options for ANY/HOST.
rules	A list of rules that specify what happens to specific packets as they pass through the firewall policy.

CREATE RULES UNDER A POLICY - CONFIG SHELL

The rules that apply to a firewall policy may be created through Config Shell; an example is given below:

```
config(firewall/policy incoming): rules
config(firewall/policy incoming rules): add
config(firewall/policy incoming rules 0): show
Entity firewall/policy item incoming field rules 0
   action                        "" (required)
   destination_address  ""
   log_prefix                      ""
   priority 0
   source_address             ""
   services (array)
```

Rule Configurable Fields	
action	The action that will be applied to matching packets.
destination_address	The destination address to which this rule will apply.
log_prefix	This sets the prefix of the info level log that is sent

	when this rule is hit. If it is empty, no logs are sent.
priority	The priority given to the selected rule. Rules with negative priorities are applied first. The default value is 0.
source_address	The source address to which this rule will apply. For multiple source addresses, a separate rule must be created for each address.

LOGGING AND DEBUGGING FIREWALL POLICIES

Some logging and debugging tools are provided for resolving firewall policy issues, as below:

- List all firewall policies configured on the device: `firewall-cmd --list-all-policies`.
- Check the xml files which contain the firewall policy configuration information, under the /etc/firewalld/policies/ directory.
- Check the journal for firewall related messages: `journalctl -xeu firewalld`

Note: `firewalld` is used to create firewall rules, `firewalld` is discussed in "Interzone Policies" below and in "Firewall Guide" on page 276.

INTERZONE POLICIES

CONFIGURE > FIREWALL > Interzone Policies > Create Interzone Policy

In the Console Manager, Interzone firewall policy is implemented through Firewalld; this is a zone-based firewall which allows you to define zones and create rules to manage the traffic between the zones.

The firewalld feature provides a dynamically managed firewall with support for network/firewall “zones” to assign a level of trust to a network and its associated connections, interfaces or sources.

The feature allows you to define policies to configure forwarding between zones and can be configured to allow directional forwarding from one or more ingress zones to one or more egress zones.

Rules and filtering may be applied at the zone level. When you add a zone, you select which services are part of that zone. Interzone policy allows these rules and filtering to be applied so as to control the type of traffic allowed to be forwarded.

The default policy, ie. when no zones are added, is that no traffic is forwarded.

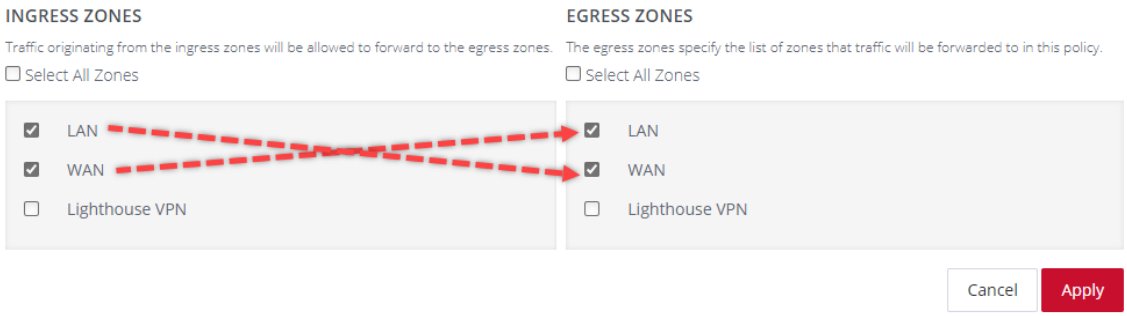
CREATE AN INTERZONE POLICY

CONFIGURE > FIREWALL > Interzone Policies > New Interzone Policy

1. Navigate to the **Interzone Policies** page: CONFIGURE > FIREWALL > Interzone Policies.
2. Click the **Add Firewall Policy** button , the **New Interzone Policy** page opens for editing.
3. In the **Name** field, enter a name that clearly identifies this policy instance to other users.
4. In the **Description** field provide a detailed description of this interzone policy (optional).

5. Click to check the boxes for each Ingress and Egress zone that is to be included in this policy. You can configure traffic in both directions by selecting both zones in the Ingress and Egress as in indicated by the red arrows in the image below:

Two Directional Traffic Interzone Policy:

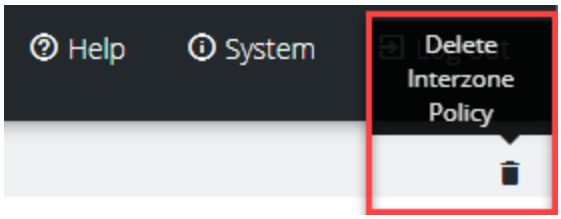


Note: Additional zones may be added to the zones list at: CONFIGURE > FIREWALL > Management > New Firewall Zone.
Zone customized rules may be edited at CONFIGURE > FIREWALL > Management > Firewall Management.

6. Click the **Apply** button to implement the policy, a green banner will inform you that the policy details are saved successfully. The interzone policy is now active.

EDIT OR DELETE AN INTERZONE POLICY

CONFIGURE > FIREWALL > Interzone Policies > Edit Interzone Policy

1. Navigate to the **Interzone Policies** page: CONFIGURE > FIREWALL > Interzone Policies.
2. Click the name of the policy you wish to edit (editable policies are identified by red text). The **Edit Interzone Policy** page opens for editing.
3. Edit the policy details to be changed.
4. If necessary, change the **Description** field to provide a detailed description of the edited interzone policy.
5. To **delete** a policy, click on the **Bin** widget in the top-right corner of the **Edit** page.
6. 

7. Click the **Apply** button to implement the edited policy, a green banner will inform you that the policy details are saved successfully. The edited interzone policy is now active.

CUSTOMIZED ZONE RULES

Customized zone rules may be applied to any zone at CONFIGURE > FIREWALL > Management > Firewall Management: "Firewall Management" on page 284.

ADDING WIREGUARD ZONES TO A FIREWALL

The WireGuard interface can be added to a firewall zone as in the following example:

```
Entity firewall/zone item zone
description "" (required)
label "" (required)
masquerade "" (required)
name zone
permit_all_traffic "" (required)
address_filters (array)
custom_rules (array)
physifs (array)
port_forwarding_rules (array)
wireguards (array)
```

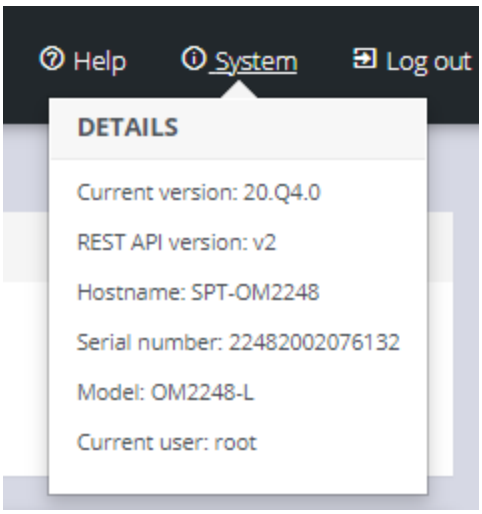
SYSTEM

CONFIGURE > SYSTEM

The **CONFIGURE > SYSTEM** menu lets you change the Console Manager hostname, perform system upgrades, and reset the system.

CHECK SYSTEM DETAILS

To ascertain current system details click on the System link at the top-right of the CM window.

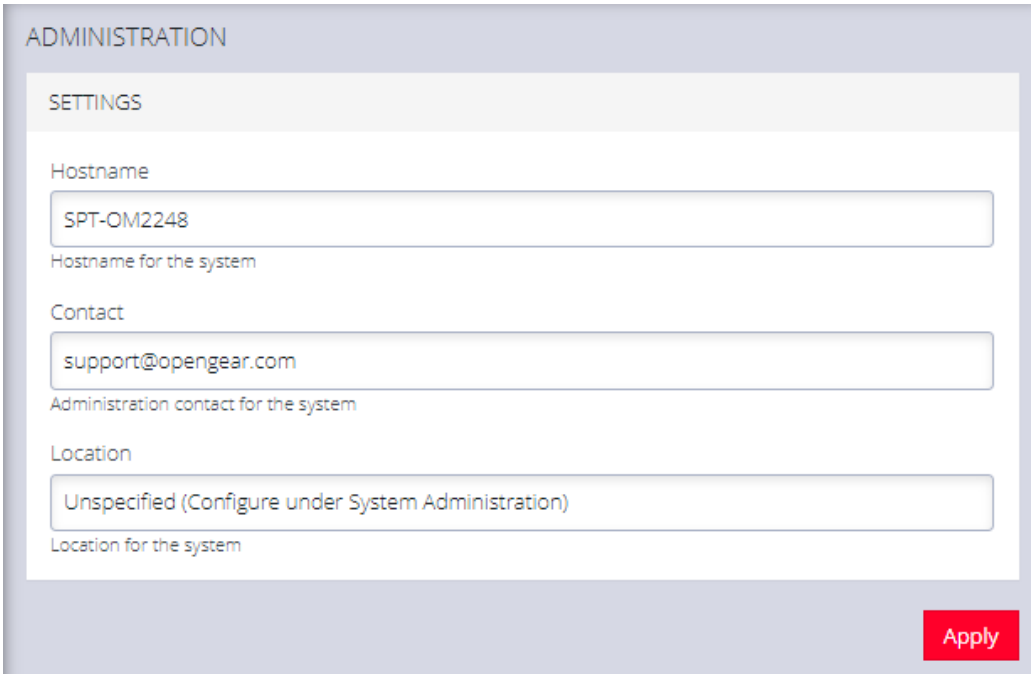


ADMINISTRATION

CONFIGURE > SYSTEM > Administration

To set the hostname, add a contact email, or set a location for the Console Manager:

1. Click **CONFIGURE > SYSTEM > Administration**.
2. Edit the **Hostname** field.



3. Click **Apply**, the new settings are saved.

DATE & TIME

CONFIGURE > SYSTEM > DATE & TIME > TIME SETTINGS

It is important to set the local Date and Time in your Opengear device as soon as it is configured. Features such as Syslog and NFS logging use the system time for time-stamping log entries, while certificate generation depends on a correct Timestamp to check the validity period of the certificate.

Your Opengear device can synchronize its system time with a remote Network Time Protocol (NTP) server. NTP uses Coordinated Universal Time (UTC) for all time synchronizations so it is not affected by different time zones.

You need to specify your local time zone so the system clock shows correct local time. The Date & Time section of the navigation bar provides a means to

- Set the time zone
- Manually set the correct time and date

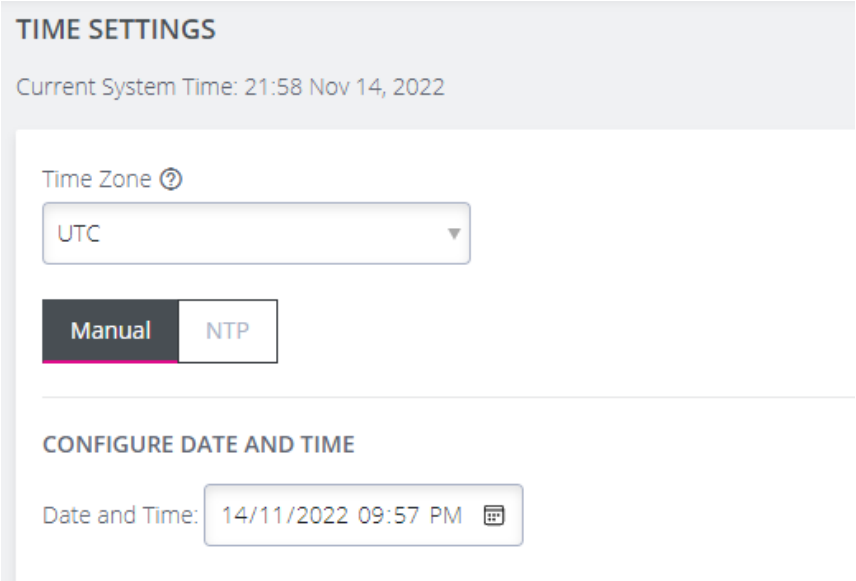
Or

- Set the date and time by NTP Server

Continued:

MANUAL DATE & TIME SET

1. Navigate to CONFIGURE > SYSTEM > DATE & TIME > TIME SETTINGS.
2. Select the applicable local time zone from the global time zone drop-down list, then, select **Manual** in the **Time Zone** section of the page.



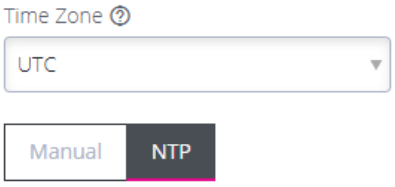
3. Select the correct date and time from the Date/Time Calendar.
4. Click the **Apply Date and Time** button.

NTP CONFIGURATION & AUTHENTICATION

Configuring an NTP server ensures the Opengear device clock is kept accurate (once Internet connection has been established).

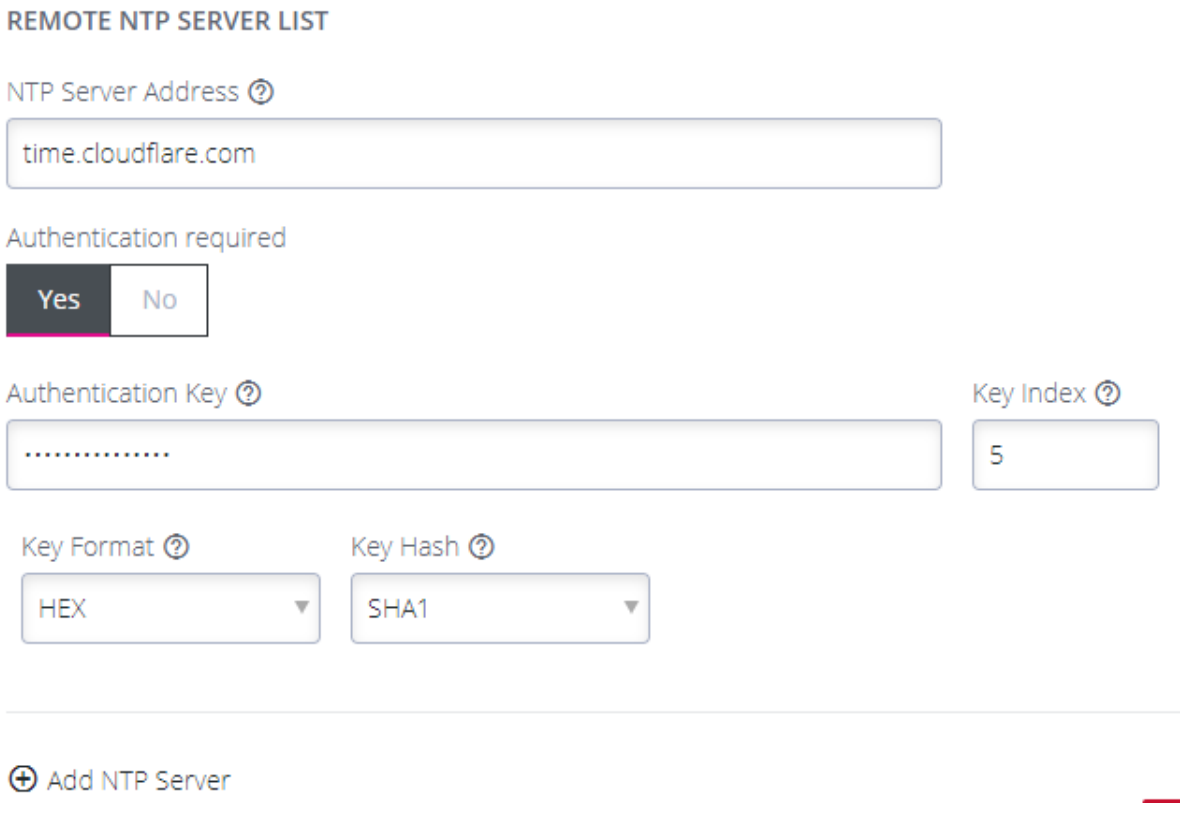
When defining an NTP server you can choose to supply an Authentication Key and Authentication Key Identifier or not to use Authentication. If NTP Authentication keys are in use, the NTP server must be verified using the Authentication Key and Authentication Key Index before synchronizing time with the server.

1. Navigate to CONFIGURE > SYSTEM > DATE & TIME > TIME SETTINGS.
2. Select the applicable time zone from the global time zone drop-down list, then, select **NTP** in the **Time Zone** section of the page.



3. In the **Remote NTP Server List** section of the page, click Add NTP Server. The '**Remote NTP Server List**' opens.

> **Note:**If your external NTP server requires authentication, you need to specify the NTP Authentication Key and the Key Index to use when authenticating with the NTP server.



4. Enter the IP address of the remote NTP Server.
5. If Authentication is required, select **Yes** and complete all sections of the **Authentication Key** form.
6. Click the **Apply NTP Settings** button.

CLI COMMANDS ASSOCIATED WITH NTP CONFIGURATION

Generate a new key:

chronyc keygen $INDEX $ALGORITHM

Examples:

```
chronyc keygen 1 SHA3-512
```

```
chronyc keygen 50 SHA1
```

```
chronyc keygen 2345 AES256
```

Check chronyd service:

```
systemctl status chronyd.service
```

```
journalctl -b 0 --unit chronyd.service
```

Check if the server has clients

```
chronyc clients
```

Check if the client is synchronizing:

`chronyc sources` - shows a list of servers available to the system, status, and offsets from the local clock and the source

`chronyc sourcestats` - show additional statistics for each server

`chronyc tracking` - see what server chrony is tracking with and performance metrics from that server execute

`chronyc activity` - see the number of servers and peers that are connected

`chronyc ntpdata` - returns data about each configured server

Check the NTP packets

```
tcpdump -vvv -i any udp port 123
```

OM specific CLI Commands

`ogcli get services/ntp`

`ogcli help services/ntp`

`ogcli replace services/ntp enabled=false` - disable NTP and clear all servers and keys.

`ogcli update services/ntp enabled=false` - disable NTP, but keep servers and keys settings.

`cat /etc/config/chronyd.conf`

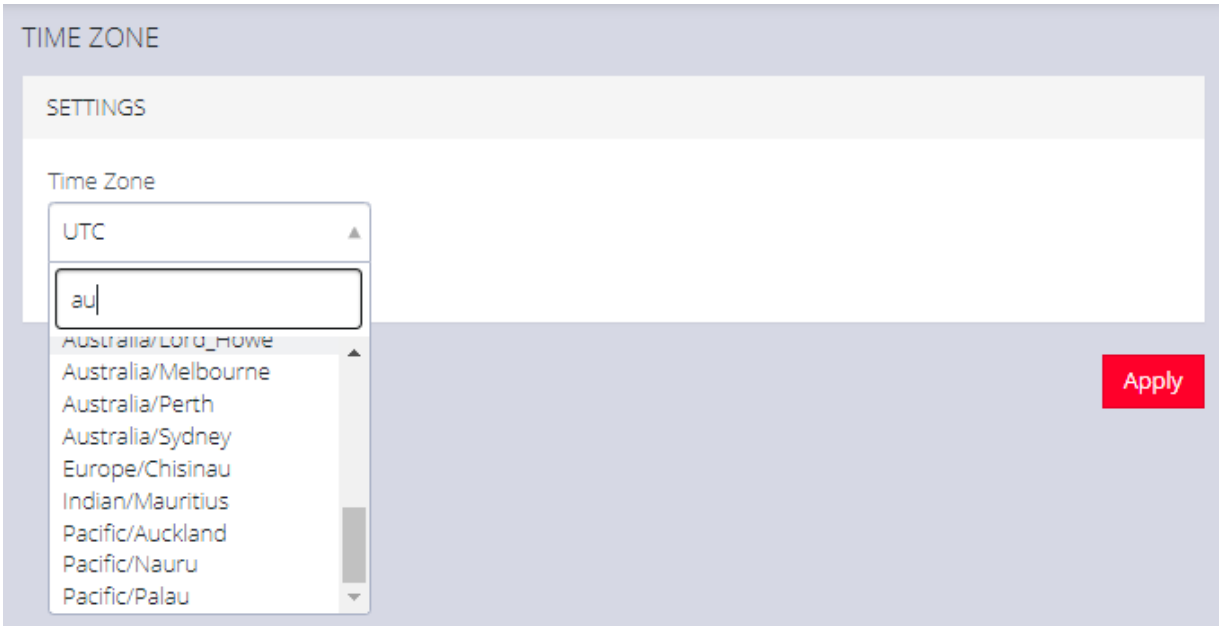
`cat /etc/config/chronyd.keys`

TIME ZONE

CONFIGURE > DATE & TIME > Time Zone

To set the time zone:

1. Navigate to the **CONFIGURE > DATE & TIME > Time Zone** page.
2. Select the Console Manager's time-zone from the **Time Zone** drop-down list. A filter is provided to make selection easier.
3. Click **Apply**.

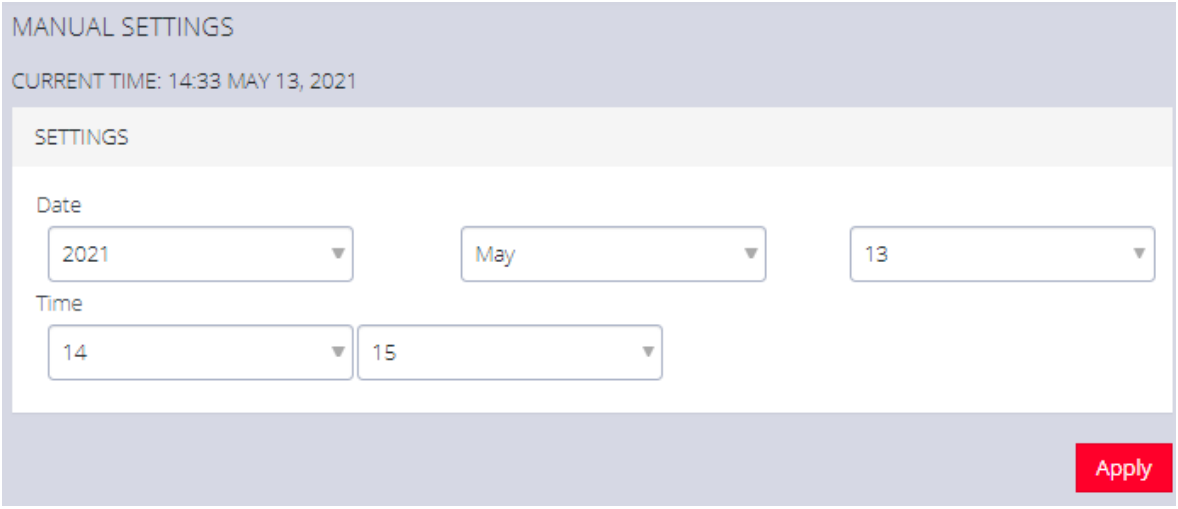


MANUAL SETTINGS

CONFIGURE > DATE & TIME > Manual Settings

To manually set the correct time and date:

1. Click **CONFIGURE > DATE & TIME > Manual Settings**.
2. Enter the current **Date** and **Time**.
3. Click **Apply**.



Note: When Automatic time setting is enabled, the manual settings are ignored and only automatic settings are applied. Nodes enrolled in Lighthouse must be on the same time zone.

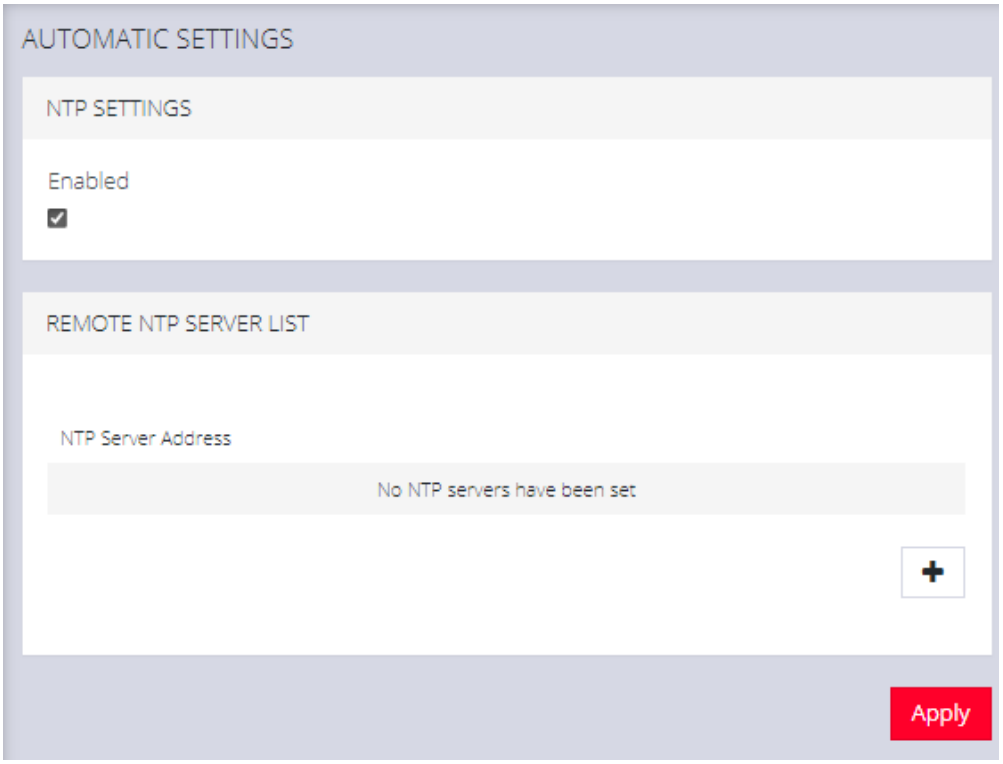
AUTOMATIC SETTINGS

CONFIGURE > DATE & TIME > Automatic Settings

Automatic Setting of the date and time:

1. Click **CONFIGURE > DATE & TIME > Automatic Settings**.
2. Click the **Enabled** checkbox.
3. Enter a working NTP Server address in the **NTP Server Address** field.

4. Click **Apply**.



Note: When Automatic time setting is enabled, the manual settings are ignored and only automatic settings are applied. Nodes enrolled in Lighthouse must be on the same time zone.

FACTORY RESET

CONFIGURE > SYSTEM > Factory Reset

You can perform a factory reset at the UI by pressing the **Factory Reset** button (CONFIGURE > SYSTEM > Factory Reset) or at the external **Erase** button, or from the **CLI**. All three methods are covered in this topic. During a factory reset the device is reset to the factory default.

Note: During the reset process, the software driven LEDs on the front of the device (Power, Heartbeat, Network Activity, Serial Activity, Cellular) may take several seconds to appear ON. The LEDs will behave as described in the LED Status table below. Note that the Power LED may appear to be OFF during part of the reboot process; this is a normal stage of the reboot process.

To reset from the **UI**:

CONFIGURE > SYSTEM > Factory Reset

To return the Console Manager to its factory settings:

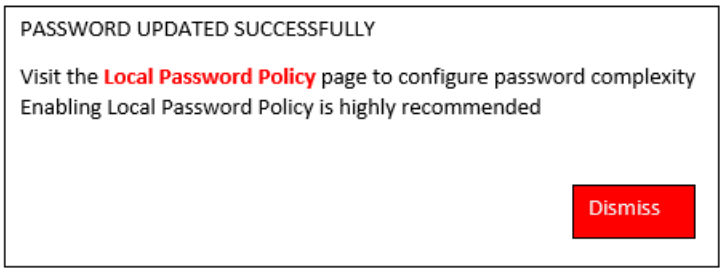
1. Login to the Web UI as a user with admin privileges.
2. Navigate to **CONFIGURE > SYSTEM > Factory Reset**.
3. Read the Factory Reset warning notice.

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

4. If you still wish to proceed with the reset, Select the **Proceed with the factory reset** checkbox.
5. Click **Reset**.

   Warning: This operation performs the same operation as the hard factory erase button. This resets the appliance to its factory default settings. Any modified configuration information is erased. You will be prompted to log in and must enter the default administration username and administration password (Username: root Password: default). You will be required to change this password during the first log in.

6. CONFIRM the message "Factory reset initiated. System will reboot in ten seconds." appears.
7. CONFIRM the appliance is undergoing a system reboot.
8. The 'Power' and 'Heartbeat' LEDs will display orange briefly, then go off after approximately five seconds.
9. Wait for LEDs to indicate the device has rebooted normally (see LED Status below).
10. Login to Web UI or CLI. Use the default password for initial login, then, change password in accordance with the local password policy.



To reset at the external **Erase** Button:

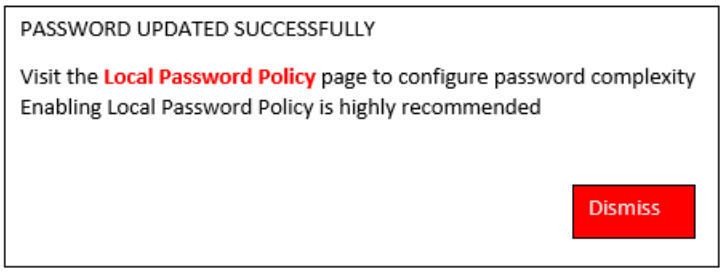
1. Press the external physical **Erase** button on the device once.

   Note: On most devices the button is at the front panel, near the LEDs. On the OM1200 the button is on the rear, near the power inlet).

2. CONFIRM all of the LEDs come on.
3. Press the physical **Erase** button on the device a second time within 5 seconds.

Note: If the **ERASE** button is not depressed within 5 seconds of the LEDs turning on, the appliance will resume normal operation.

4. The 'Power' and 'Heartbeat' LEDs will will display orange briefly, then go off after approximately five seconds.
5. Wait for LEDs to indicate the device has rebooted normally (intermittently flashing heartbeat changes to green, see LED Status below).
6. Login to Web UI or CLI. Use the default password for initial login, then, change password in accordance with the local password policy.



To reset from the **CLI** terminal:

1. Log in at the CLI terminal, then enter:

```
root@om2248-l-tp1-p14:~# factory_reset
```

2. Confirm: Factory reset system? [yes/no]:
3. Follow the procedure from step 2 in the 'Erase button' procedure above.

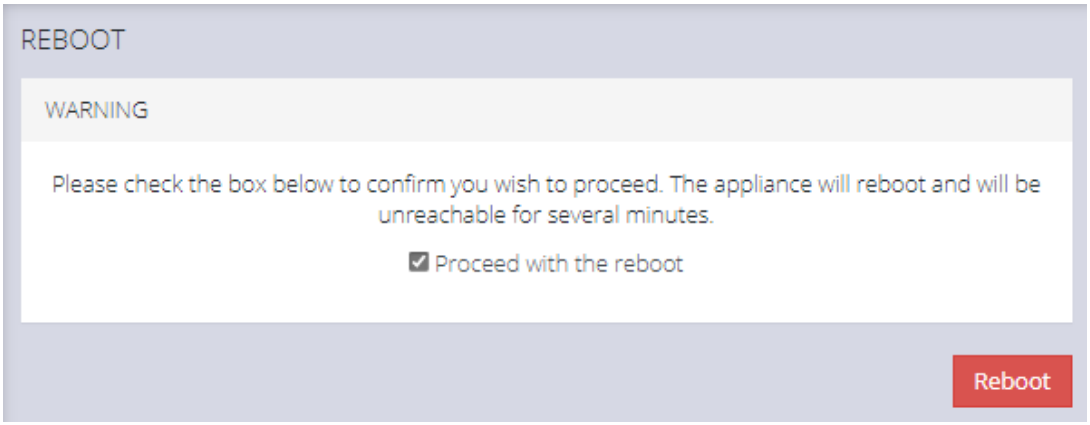
REBOOT

PERFORM A SIMPLE REBOOT FROM THE UI

CONFIGURE > SYSTEM > Reboot

To reboot the Console Manager:

1. Navigate to **CONFIGURE > SYSTEM > Reboot**.
2. Select **Proceed with the reboot**,
3. Click **Reboot**.



See also "Factory Reset" on page 312 for detailed information about device behavior that may occur during a factory reset procedure.

EXPORT CONFIGURATION

The current system configuration can be downloaded as a plain text file. It contains all configuration performed via the Web UI and the ogcli tool.

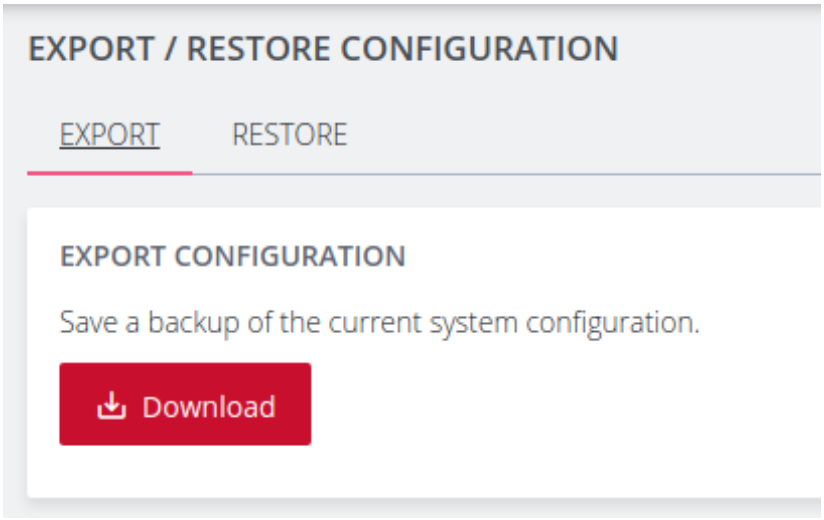
It does not contain log files, user scripts, docker containers, service configuration or other files stored via other means.

The exported configuration may be useful for:

- disaster recovery
  - issues with system upgrades
  - unexpected configuration changes
- replacing devices after RMA
- configuration templating

EXPORT CONFIGURATION VIA WEB UI

CONFIGURE > SYSTEM > Export / Restore Configuration



To export the system configuration, click the **Download** button and save this file.

Sensitive data such as passwords and tokens will be obfuscated in the configuration export.

> **Note:**The default filename includes the system hostname and a timestamp. For example, **cm8148_20210910_config.txtem8000_20210910_config.txt**

EXPORT CONFIGURATION VIA OGCLI

The system configuration can also be exported using the ogcli tool.

As an administrative user, run the following command:

```
ogcli export <file_path>
```

CONTROL THE EXPORT OF SENSITIVE DATA

The display of sensitive data during export via ogcli can be controlled by modifying the ogcli command:

- To display secrets in cleartext, run:

```
ogcli --secrets=cleartext export <file_path>
```

- To display obfuscated secrets, run:

```
ogcli --secrets=obfuscate export <file_path>
```

- To display secrets masked with ********, run:

```
ogcli --secrets=mask export <file_path>
```

Caution: Configuration exported with **--secrets=mask** cannot be used to import configuration.

LIGHTHOUSE NODE BACKUP

Configuration export can be scheduled to be performed periodically using the Lighthouse Node Backup feature.

For more details, consult the Lighthouse User Guide: https://opengear.com/support/documentation/

RESTORE CONFIGURATION

An exported system configuration can be imported to the node using the Web UI or ogcli tool.

Note: If the configuration was exported using **--secrets=mask**, it cannot be used for configuration import.

Note: It may take up to ten minutes to import a config file with a large amount of configuration.

RESTORE CONFIGURATION VIA WEB UI

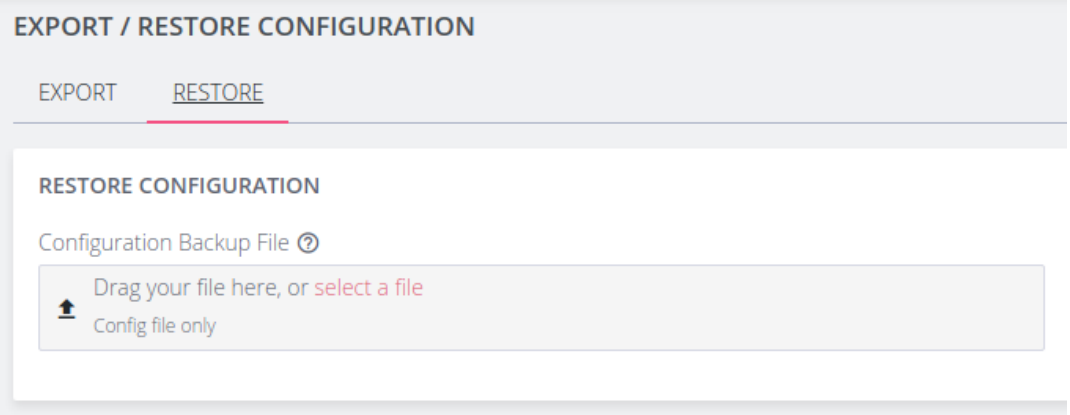
CONFIGURE > SYSTEM > Export / Restore Configuration

Importing configuration using the Web UI will use the restore strategy. Restoring configuration will override all settings on the node.

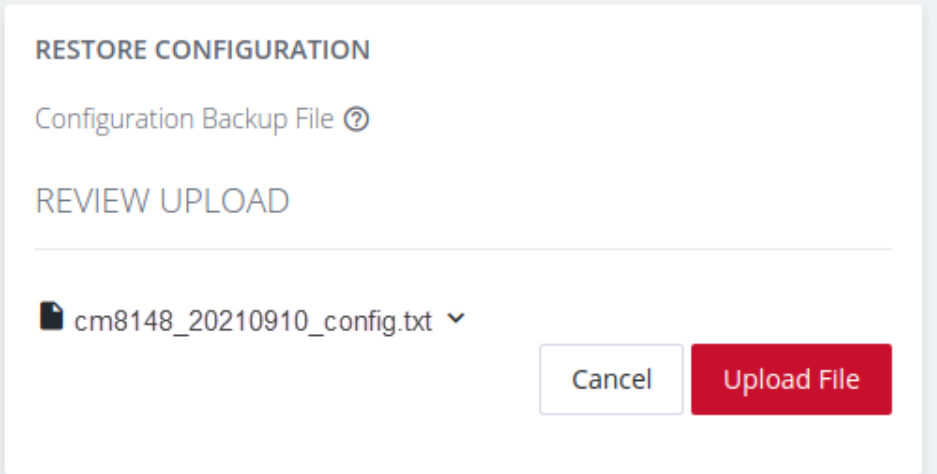
Only configuration from the same version and model can be restored.

To restore the system configuration:

1. Click the **Restore** tab



2. Select the configuration file to import.
3. Review the configuration by clicking the arrow to display the file content.



4. Click the **Upload File** button to start the import process.
5. A green banner will display when the configuration import is successful.

IMPORT CONFIGURATION VIA OGCLI

The system configuration can also be imported using the ogcli tool. Either the import or restore strategies can be used.

IMPORT CONFIGURATION

Configuration that is imported using the `ogcli import` command will be merged with the current system configuration, preserving the current values and adding missing entries from the exported configuration where required.

As an administrative user, run the following command:

```
ogcli import <file_path>
```

RESTORE CONFIGURATION

Configuration that is imported using the `ogcli restore` command will replace the current system configuration. The resulting system configuration will reflect what is in the exported configuration.

> **Note:**Restoring a config file may take up to three minutes for large files.

As an administrative user, run the following command:

```
ogcli restore <file_path>
```

AUTOMATED ROLLBACK TO WORKING CONFIGURATION

Config Rollback provides an automated rollback mechanism that ensures a device will automatically revert to its last known working configuration in case of a failed restore. Automated Rollback is the default configuration of this feature and cannot be overridden or configured for manual operation.

Rollback maintains operational stability, ensuring the system does not become partially upgraded due to some error during upgrade. The ability to roll back to a previously safe configuration minimizes downtime and service disruption, making it a vital addition to the system's resilience.

Rollback behavior in the event of a detected restore failure:

The system automatically detects a configuration update failure. Upon detection of a failure, the system automatically initiates a rollback to the last known working configuration without user intervention until a known working configuration is successfully installed.

```
root@om2248:~# ogcli restore restore.txt
⋮    restoring data

restore failed with the following error(s):
Error(s) detected during REPLACE operation on services/snmp_alert_managers
Adding record 1 from the list of supplied records has failed
Error: 'bad_address' is not a valid network address
Error: Push command failed

⋮    rolling back config
rollback successful
```

Updating the Import/Restore File

The import/restore file has to follow a very specific format; deviation from format should be avoided. Comments and blank lines are allowed but any commands not starting with either ogcli or config need to be in proper heredoc format, see the below example (note the <<'END' format, this is the only heredoc marker allowed).

Note: Inline comments will not work.

```
config replace system/session_timeout <<'END'
  cli_timeout=0
  serial_port_timeout=0
  webui_timeout=20
END
```

Rollback Capabilities

- When the system initiates a rollback, it will log to syslog, print a message in the CLI and display a pop up “toast” notification in the Web UI.
- This system is resilient to network issues; once Rollback is started it will continue without the user being connected to the network.
- If a user sends a ctrl-c signal during restore/import the system will also begin a rollback.
- If a user sends a ctrl-c signal during the rollback it will be ignored. This is to ensure that the system does not enter a bad state.
- Users cannot start another restore/rollback if there is already one running on the system, a warning will be issued.

Rollback Limitations

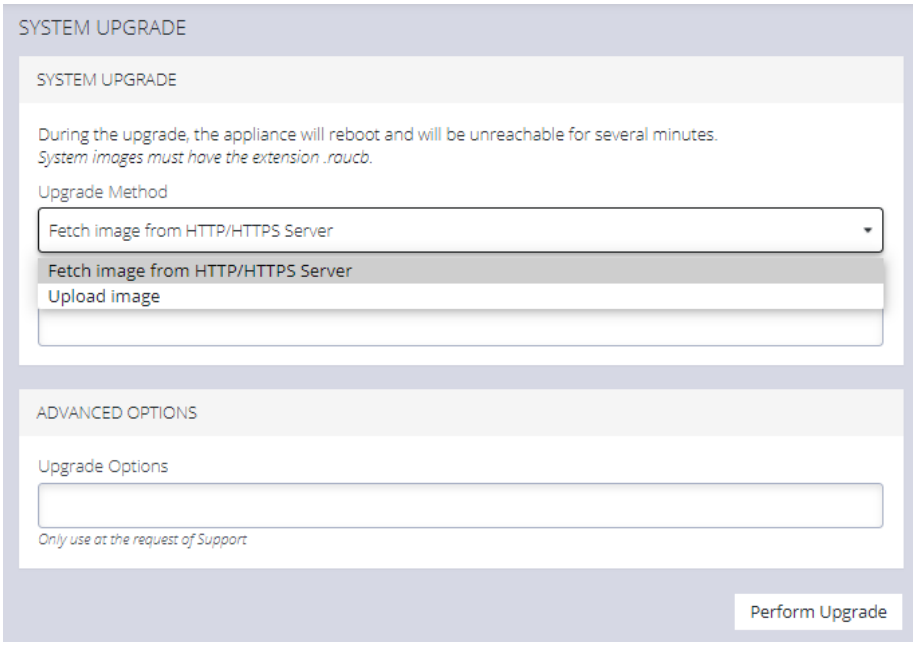
- Config diff and Rollback can be used by any user with administrator permission and access to the shell. It is initiated via the web-ui or command line.

- Only one import/restore and rollback is permitted on the box at once.
- Rollback cannot be initiated without a failing import/restore.
- Rollback cannot be initiated to a specific version.
- Rollback does not support manual intervention and once rollback is initiated it cannot be stopped.
- Rollback cannot be initiated once import/restore is complete.

SYSTEM UPGRADE

CONFIGURE > SYSTEM > System Upgrade

You can perform a system upgrade when new firmware is released. After specifying the location of the firmware and beginning the upgrade process, the system will unavailable for several minutes and then reboot. Unlike a factory reset, users, and other configuration data is maintained after the upgrade.



To perform a system upgrade:

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

> **Note:**See https://opengear.com/support/device-updates/ for firmware updates.

UPGRADE VIA FETCH FROM SERVER

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

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

UPGRADE VIA UPLOAD

If upgrading via **Upload Image**:

1. Click the **Choose file** button.
2. Navigate to the directory containing the file.
3. Select the file and press **Return**.
4. Click **Perform Upgrade**.

> **Note:**The **Advanced Options** section should only be used if a system upgrade is being performed as part of an Opengear Support call.

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

SNMP

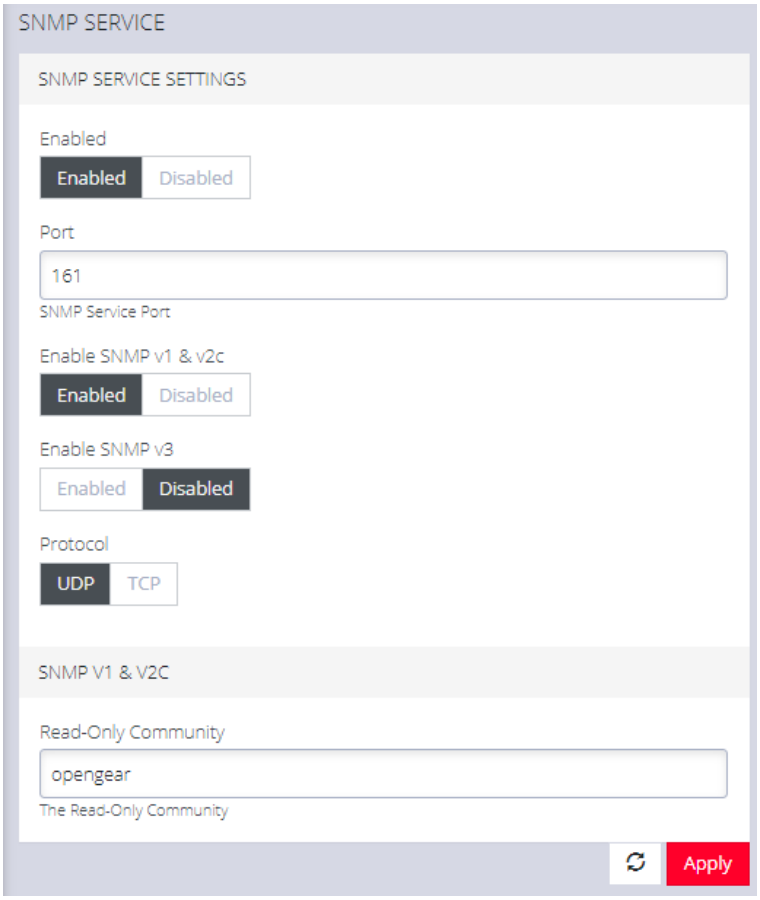
CONFIGURE > SNMP

The **CONFIGURE > SNMP** menu has two options, **SNMP Service** and **SNMP Alert Managers**.

SNMP SERVICE

CONFIGURE > SNMP > SNMP Service

Navigate to the **CONFIGURE > SNMP > SNMP Service** to open the **SNMP Service** page.



SNMP Service allows you to specify which SNMP services to enable. When you click on **ENABLED** for **SNMP V1 & V2** or **SNMP V3**, a detail form appears where you can add service specific settings.

You can also specify the **SNMP Service Port** and choose between **UDP** or **TCP** for the **Protocol**.

SNMP ALERT MANAGERS

CONFIGURE > SNMP > SNMP Alert Managers

Navigate to **CONFIGURE > SNMP > SNMP Alert Managers** to open the **SNMP Alert Managers** page.

See the "Multiple SNMP Alert Managers" on the next page feature for information about configuring more than one SNMP manager.

On this page, you can set the following:

- **Address**: The IPv4 Address or domain name of the computer acting as the SNMP Manager.
- **Version**: The version of SNMP to use. The default is v2c.
- **Port**: The listening port used by the SNMP Manager. The default value is 162.
- **Manager Protocol**: The transport protocol used to deliver traps to the SNMP Manager. The default value is UDP.
- **SNMP Message Type**: The type of SNMP message to send to the SNMP manager. The INFORM option will receive an acknowledgment from the SNMP manager and will retransmit if required. The TRAP option does not expect acknowledgments.

For SNMP V1 & V2C, you can specify a **Community**. This is a group name authorized to send traps by the SNMP manager configuration for SNMP versions 1 and 2c. This must match the information that is setup in the SNMP Manager. Examples of commonly used values are log, execute, net and public.

MULTIPLE SNMP ALERT MANAGERS

CONFIGURE > SNMP > SNMP Alert Managers > Add New SNMP Alert Manager

The Multiple SNMP Alert Managers feature provides the option to configure more than one SNMP manager. Multiple SNMP Alert Managers can receive trap and inform events that can be used to trigger remedial action; events can be sent to multiple SNMP Alert Managers. The AR functionality sends traps to all configured SNMP Alert Managers for a reaction of type SNMP. Whether you input an IPv6 address or a domain name, the correct protocol needs to be selected.

CREATE OR DELETE AN SNMP MANAGER

To create a new SNMP manager:

1. Navigate to **Configure > SNMP > SNMP Alert Managers**.
2. Click the **Add New SNMP Manager** button (a plus character in the top-right of the window)
3. Complete the new **SNMP Alert Manager Form** as per the **Definitions** table below.
4. Click the **Submit** button. A banner appears confirming that the new SNMP Manager has been successfully created.
5. The new manager appears in the list of SNMP Alert Managers.
6. To delete an SNMP manager, click on the IP address of the item to open the **Edit SNMP Manager** page for that SNMP Manager.
7. Click on the **Delete SNMP Manager** widget in the top-right of the page.

Note: If you would like to use an IPv6 Address, then you need to select either UDP6 or TCP6 from the list of protocols. Whether you input an IPv6 address or a domain name, the correct protocol needs to be selected.

Note: For SNMP V3 TRAPS, an Engine ID will be provided by default if none is specified. This is generated by the snmpd service and can be found in the SNMPD RUNTIME CONF /var/lib/net-snmp/snmpd.conf. Traps will be sent for Alerts added in **Configure > SNMP Alerts.** Traps will also be sent to all the configured SNMP Alert Managers for a Playbook SNMP Reaction.

NEW SNMP ALERT MANAGER PAGE DEFINITIONS

New SNMP Alert Manager Field	Definition
Description	The editable Description field allows you to add a description of the SNMP Alert Manager.
Server Address	The IPv4/IPv6 address or domain name of the computer acting as the SNMP Alert Manager.
Port	The listening port used by the SNMP Alert Manager. The default value is 162.
Protocol	The transport protocol used to deliver traps or informs (for SNMP v3).
	UDP - Speeds up transmissions by enabling the transfer of data before an agreement is provided by the receiving party.

	TCP - A commonly used protocol used to transmit data from other higher-level protocols that require all transmitted data to arrive.
	UDP6 - Similar to UDP but uses IPv6.
	TCP6 - Similar to TCP but uses IPv6.
Version	The version of SNMP protocol to use. The default value is v2c. For further reading on SNMP versions we suggest: https://en.wikipedia.org/wiki/Simple_Network_Management_Protocol#Protocol_versions
SNMP V1 & V2C Community	A group name authorized to send traps by the SNMP alert manager configuration for SNMP versions 1 and 2c. This will need to match what is setup in the SNMP alert manager. Examples of commonly used values are log, execute, net and public.
	Click the **Submit** button to finalize the New SNMP Manger process.
	Click the bin widget to **Delete** an SNMP Manager (in the Edit SNMP Manager page).

ADVANCED OPTIONS

The Console Manager supports a number of command line interface (CLI) options and REST API.

address : Primary Lighthouse address to enroll with

api_port : Optional port to use for the primary address when requesting enrollment

password : LH global or bundle enrollment password

bundle : Name of LH enrollment bundle

CONFIG CLI GUIDE

The Config Command Line Interface(CLI) provides users with an interactive and familiar environment similar to other networking devices that users may be familiar with. The result is a user-experience that feels like an Interactive CLI .

Advantages of the Config CLI are:

- Interactive CLI makes everyday operations such as configuration changes and troubleshooting activities easier for users.
- Items can be created or updated without being applied immediately.
- Items that are not applied are indicated by an asterisk (*) beside them when viewing information..
- Tab complete is supported for many commands.
- Built-in context sensitive help.
- Has a structured, tabular view when displaying lists of data.

NAVIGATION IN CONFIG CLI

STARTING A SESSION IN CONFIG CLI

Start the config shell by typing `config` at a bash prompt. The bash prompt is presented to root and admin users when they log in via SSH or on the management or local console.

EXITING A CONFIG CLI SESSION

You can exit the Interactive CLI by in any of the following ways:

- Type `exit` to end the session.
- Send an EOF (Control+D).
- Send an INT (Control+C).

> **Note:**The session is prevented from exiting if there are un-committed changes, this condition is indicated by a message. However, you can force an exit by immediately executing an exit command again, any un-committed changes will be discarded.

NAVIGATING THE CONFIG CLI

The Config CLI operates using a hierarchy . Due to the variety of endpoints, there are several ways to get to a place where you may want to make changes.

- Starting at the root, enter endpoint names to descend down to lower endpoints.
- Similarly, type 'up' to ascend towards the root or type 'top' to reset to the root context.

Note: Every endpoint name is an operation that descends into that endpoint.

When using the config CLI, it is possible to navigate ‘downwards’ through multiple contexts with a single command line.

HIERARCHICAL IDENTIFIERS

This section outlines the identifiers needed to navigate the CLI.

Identifier	Description
Singleton endpoints	These require only the endpoint name to be uniquely identified.
List/item endpoints	The first level is the endpoint name, the second level is the item identifier (the identifier is the same identifier used by ogcli).
Multiple identifiers	A single endpoint (ssh/authorized_keys) requires an extra identifier. In this case, the hierarchy is: ssh/authorized_keys > userid > [key_id]
Nested fields	The interactive CLI treats nested fields as additional hierarchy levels. This applies both to arrays and maps. For arrays of complex values, each value shall also be a hierarchy level.

UNDERSTANDING FIELDS, ENTITIES AND CONTEXTS

The Config CLI allows you to configure the device settings through a number of required fields, which provide the settings for the device.

These fields are grouped in *entities* that describe a small set of functionality, for example, there is a ‘user’ entity which is used to access user settings. Entities can contain sub-entities as well as simple fields.

HOW CONTEXT OPERATES IN THE CONFIG CLI

Description

The *context* is the current entity that is the focus of the config shell. When the shell is first started, the context is a special parent context from which sub-entities can be seen. Within the config shell, a number of commands are available, depending on the current context.

When config shell is started the context is at the “top context” which lists all the entities when the show command is used. If the name of an entity is typed, then the context moves 'down' into that entity. When simple commands such as `show`, `help` or `apply` are used, they will act on the current context. The context can be moved down further by typing the name of an item.

Entities can contain sub-entities as well as simple fields. For example, there is a ‘user’ entity which is used to access user settings. Fields are grouped within entities that describe a small set of functionality.

Navigating Using Context

You select a context by typing the name of the target entity and pressing Enter/Return; the new context is shown in the prompt between brackets. In the following example, the 'user' context is accessed and then the 'john' sub-entity is accessed causing the context to become 'user john'.

The 'show' command is used to list the entities and fields that descend from the current context.

```
config: user
config(user): show
Item names for entity user
  john matt myuser netgrp root
config(user): john
config(user john):
Entity user item john
  description
  enabled         true
  no_password     false
  password
  ssh_password_enabled true
  groups (array)
config(user john):
```

The following example will navigate the context to the root user object without first having to navigate to the user context:

```
config: user root
config(user root):
```

Sub-objects are supported. In the following example, power_supply_voltage_alert and syslog are nested sub-objects of the onitoring/alerts/power entity:

```
config: monitoring/alerts/power power_supply_voltage_alert syslog
config(monitoring/alerts/power power_supply_voltage_alert syslog):
```

GLOBAL & ENTITY-CONTEXT COMMANDS

GLOBAL CONTEXT COMMANDS

The table below lists commands available on any context:

Global Command	Description
`help (or '?')`	Show help which is context sensitive. It will list some special details about the current context, the list of sub entities (or fields) and a list of available commands.
`help <entity>`	Displays short-form help for the specific entity.
`show`	Lists the available entities and fields.
`<entity>`	Inputing the name of an entity changes the context to focus on the named entity.
`exit`	Exit the command shell.

ENTITY CONTEXT COMMANDS

In addition to the global context commands, once an entity context is selected then further, entity context, commands become available.

Entity Command	Description
`<field>`	Show the value of a field.
`help <entity>`	Displays short-form help for the specific entity.
`<field> <value>`	Set the value of a field.
`delete`	Deletes the current enitity. This is available when the context entity is an item in a list.
`add`	Append a sub-entity or field to the current entity. This is only available when the context entity is a list.

CONFIG CLI ENTITIES

The config shell allows the user to configure a number of fields which are the settings for the device. These fields are grouped in entities that describe a small set of functionality. For example, there is a ‘user’ entity which is used to access user settings. Entities can contain sub-entities as well as simple fields.

Once in the shell, a number of commands are available depending on the current context. The context is the current entity that is the focus of the config shell. When the shell is first started, the context is a special parent context from which sub-entities can be seen.

Once a context is selected by typing the name of the entity, it is shown in the prompt between brackets. e.g. In the following snippet, the ‘user’ context is accessed and then the ‘john’ sub-entity is accessed causing the context to become ‘user john’. The ‘show’ command is used to list the entities and fields that descend from the current context.

SUPPORTED ENTITIES

Entity	Definition
access_right	An access right is a permit that grants the holder access to a feature or collection of related features.
auth	Configure remote authentication, authorization, accounting (AAA) servers.
auto_response/beacon	Read and manipulate the Auto-Response beacons on the NetOps Console Server appliance.

auto_response/reaction	Read and manipulate the Auto-Response reactions on the NetOps Console Server appliance.
auto_response/status	Read the AutoResponse Status on the NetOps Console Server appliance.
auto_response/status/beacon-module	Read the AutoResponse Status of Beacon Modules on the NetOps Console Server appliance.
cellfw/info	Retrieve cellular modem version and related information.
cellmodem	Retrieve information about the cell modem.
cellmodem/sim	Cell modem SIM status.
conn	Read and manipulate the network connections on the NetOps Console Server appliance.
diff	config diff performs a comparison of the active configuration and an input configuration file, which is the product of the ogcli export <template-file> operation. Config diff shows additions, removals and changes clearly in the a streamlined format with only functional differences between the input and running configurations. See also ogcli diff.
failover/settings	failover/settings endpoint is to check and update failover settings. When failover is enabled, this device will consume from 1MB to 1.6 MB of bandwidth per day on the probe_physif connection. If the probe addresses are unreachable, this device will take from 108 to 156 seconds to enter the failover state.
failover/status	failover/status endpoint is to check current failover

	status.
firewall/policy	A collection of policies defined for the NetOps Console Server appliance's firewall. A policy specifies which zones traffic is allowed to route between.
firewall/predefined_service	A collection of predefined services for the NetOps Console Server appliance's firewall. A service is a named grouping of one or more TCP or UDP ports for a particular networking protocol. For example, the 'https' service refers to TCP port 443. This collection contains predefined services for common protocols and doesn't include the services added by the administrator.
firewall/service	A collection of custom services defined for the NetOps Console Server appliance's firewall. A service is a named grouping of one or more TCP or UDP ports for a particular networking protocol. For example, the 'https' service refers to TCP port 443. The appliance includes many predefined services for common protocols (see /firewall/predefined_services). This collection contains only custom services which have been defined by the administrator.
firewall/zone	Collection of zones defined for the NetOps Console Server appliance's firewall. A zone includes 1 or more interfaces.
group	Retrieve or update user group information.
ip_passthrough	IP Passthrough endpoints are for retrieving / changing IP Passthrough settings.
ip_passthrough/status	The IP Passthrough status endpoint provides

	information about what part of the IP Passthrough connection process the device is currently at and information about the connected downstream device.
ipsec_tunnel	Read and manipulate the IPsec tunnels on the NetOps Console Server appliance.
lighthouse_enrollment	View and control enrollment to a lighthouse.
local_password_policy	Configure the password policy for local users. This includes expiry and complexity settings.
logs/portlog	None
logs/portlog_settings	Check and update port log settings.
managementport	Used for working with local management console information.
monitor/brute_force_protection/ban	Used for monitoring addresses banned by Brute Force Protection.
monitor/lldp/chassis	Get the current status of the network discovery (LLDP/CDP) protocols on this device.
monitor/lldp/neighbor	Get the list of neighboring devices (peers) that have been discovered by the LLDP protocol.
monitor/static_routes/ status	Used for monitoring the status of static routes. Only IPv4 static routes are supported.

monitoring/ alerts/networking	Retrieve and configure Networking Alert Group settings.
monitoring/alerts/power	Retrieve and configure Power Alert Group settings.
monitoring/alerts/system	Retrieve and configure System Alert Group settings.
pdu	Configure, monitor and control PDUs connected to the device.
pdus/drivers	Read the PDU driver list.
physif	Read and manipulate the network physical interfaces on the NetOps Console Server appliance.
port	Configuring and viewing ports information.
port_session	None.
ports/ auto_discover/schedule	Manage Port Auto-Discovery Scheduling.
ports/status_port	Provides information about the serial pin status and Tx & Rx counters for each of this device's serial ports.
system/admin_info	Retrieve or change the appliance system's information (hostname, contact and location).
services/ brute_force_protection	Provides access to the Brute Force Protection configuration on the system. When this service is enabled, the system watches for multiple failed login attempts and temporarily bans the offending IP Address for the configured amount of time.

services/lldp	Provides access to the Network Discovery Protocols (LLDP/CDP) configuration.
services/ntp	Provides access to the NTP client configuration on the system.
services/routing	Retrieve and configure routing services on the NetOps Console Server appliance.
services/ snmp_alert_manager	SNMP Alert Managers are used to receive and log SNMP TRAP and INFORM messages sent by the NetOps Console Server. To receive SNMP alerts generated by the system at least one SNMP Alert Manager must be configured.
services/snmpd	Simple Network Management Protocol (SNMP) is an Internet Standard protocol for collecting and organizing information about managed devices on IP networks and for modifying that information to change device behaviour. This entity allows configuration of the SNMP service.
services/ssh	Configure the Secure Shell Protocol (SSH) service.
services/syslog_server	Provides access to the remote syslog server configuration.
services/tftp	Trivial File Transfer Protocol (TFTP) is a service that allows files to be transferred to or from the NetOps Console Server appliance. This entity provides access to the TFTP server configuration on the system.
single_session	Can be enabled on a given port to prevent multiple users from connecting to that port or limit the port to a single concurrent connection.

ssh/authorized_key	Configure the SSH authorized keys for a specific user.
static_route	Configuring and viewing static routes.
system/admin_info	Retrieve or change the NetOps Console Server appliance system's information (hostname, contact and location).
system/banner	Retrieve or change the appliance system's banner text.
system/cell_reliability_test	None.
system/cellular_logging	Cellular logging provides the ability to capture the RRC connection messages from the EM7565 cellular module. This entity allows configuration of cellular logging and is only to be used during compliance testing.
system/cloud_connect	Retrieve or change the appliance system's cloud connect configuration.
system/diskspace	Retrieve the system's Disk Space usage.
system/info	Retrieve basic system information.
system/model_name	Retrieve the appliance's Model Name.
system/serial_number	Retrieve the appliance's Serial Number.
system/session_timeout	Retrieve or change the appliance session timeouts.

system/ssh_port	The SSH port used in Direct SSH links.
system/ system_authorized_key	Configure the SSH authorized keys for all users.
system/time	Retrieve and update the NetOps Console Server's time.
system/timezone	Retrieve and update the system's timezone.
system/version	Retrieve the appliance's most recent firmware and REST API version.
user	Retrieve and update user information.

CONFIG CLI COMMANDS

Command	Definition
add	Add a new item for an entity.
apply	Apply changes on just the current entity.
changes	View a list of config areas with unapplied changes.
delete	Delete an item for an entity.
diff	Show additions, removals, changes and functional differences between the input and running configurations. See also `ogcli diff`.
discard	Discard changes on just the current entity.
edit	Making changes to configuration options without navigating through the hierarchy.
exit	Leave config mode without applying changes.
help / ?	Display the available options for the configuration section. Can be used in combination with a command or configuration option to access help documentation.

import/export	Copy a config file from a specific network location to the console server and run the file. The import/export commands operate in bash, ie. outside of config CLI. You must exit config to operate the import/export features.
show	Display information relevant to the configuration section, highlighting changes.
up/exit/ ..	Allows users to traverse the configuration hierarchy.

ADD

Description

The `add` command will add a new item for an entity. The add command requires a unique value to identify the record. This will be used for the entity's label field.

The **add** command can be used:

- Anywhere within the command structure to begin the process of progressively adding an element.
- As part of a single line command where an element is added and simple fields are set.

Parameters

`entity` - the entity to which the new item will be added.

`label` - a unique value to identify the record.

`field` - optional field to set for the item.

`value` - optional value corresponding to the field.

Syntax

```
add <entity> <optional-entity> <label> <optional-field> <optional-value>
```

Example

```
add user aconsoleuser description "I am a console user"
```

APPLY

Description

The `apply` command allows users to stage configuration changes by allowing proposed changes to be held in memory, separate from active configuration until they are applied.

This may be considered from a user perspective like this:

"When I am adding users and realize that groups are missing, I can take a pause and add the groups without having to discard my work so far."

or

"When I am in the process of creating a new firewall zone but there is required service missing, I can go off and add the service and come back without losing changes."

Users can choose to apply changes in the following manner:

- Isolated changes that are specific to sections of configuration.
- Across all configurations.

Parameters

When no parameters are provided, the command will apply the changes in the current item context. For example, if the current context is `user consoleuser`, any changes to the `consoleuser`user will be saved. If the `apply` command is used outside of an item context, this will result in an error.

`apply all` – When the '`all`' parameter is added, the command will apply all changes to all items that have been changed in this session.

Syntax

`apply [all]`

Examples

Apply changes to a single item

These commands change a user. Then the apply command is used while still in the "user myuser" item context so only changes to this user are applied:

```
config: user myuser
config(user myuser): password secret123 description "This is my user"
config(user myuser): apply
```

APPLY ALL CHANGES

These commands add a new group and then change a port setting. At the end, the apply all command saves both the group and port items.

```
config: add group mygroup
config: group mygroup
config(group mygroup): access_rights
config(group mygroup access_rights): add pmshell
config(group mygroup access_rights): up
config(group mygroup): ports
config(group mygroup ports): add port01
config(group mygroup ports): top
config: port port01
```

```
config(port port01): label "Port for my group"
config(port port01): top
config: apply all
```

APPLY CHANGES TO SPECIFIC SECTIONS OF CONFIGURATION

From within a specific section of hierarchy. For example:

```
config users johnsmith
apply
```

This will apply any changes made specifically within the user's configuration section.

Apply changes from a different section in the hierarchy:

For example, if changes have been made in

```
config users johnsmith
```

but the user has moved elsewhere in the hierarchy, the command:

```
apply users johnsmith
```

will apply any changes made specifically within the user's configuration section.

Alternatively, a user might choose to apply all changes in the user list using the following command:

```
apply users
```

Using `apply` across all configurations

```
apply
```

```
apply all
```

CHANGES

Description

The `changes` command allows users to view a list of config areas with unapplied changes.

This will be a list, ordered alphabetically. Users should be able to copy and paste items from the list and use it in conjunction with the show command to view details.

Parameters

none

Syntax

changes

Examples

The following example shows changes made to multiple users and a port:

```
config: edit user root description "New description"
config: add user newuser description "New User"
config: edit port port01 baudrate 115200
config: changes
Entity user item root (edit)
 description New description
Entity user item newuser (add)
 description New User
Entity port item port01 (edit)
 baudrate 115200
```

DELETE

Description

The `delete` command is used to delete an item or entity or remove a config section or sub-section. The command requires a unique value to identify the record. This will be used for the entity's label field.

Similar to the `add` command, `delete` makes the change in a temporary state and will affect configuration only once applied.

The `delete` command can be used on:

- Existing configuration
- Unapplied changes

When used on unapplied changes, this will behave in the same way as the `discard` command.

Parameters

`entity` - the entity from which to delete the item.

`Item-label` - the label identifying the item to delete.

Syntax

```
delete <entity> <optional-entity> <item-label>
```

Example

```
delete user aconsoleuser
config:
```

Removing an element

From the users context:

```
delete "username"
```

Single line command

```
delete user "username"
apply
```

Either of the above examples will result in exiting the context of an item being deleted.

Refer to the `apply` command for how this will behave.

DIFF

Description

The `config diff` feature provides the ability to compare the current running configuration of a device with a previously exported configuration template generated with `ogcli export`.

The `config diff` tool functions by performing an export of the current configuration of the system, and comparing it with a previous exported file, showing only the changes in a streamlined diff output containing the functional changes only.

Note: The `config diff` tool is performs the diff functionality in the same way as `ogcli diff`, and can be used interchangeably using export files in either format. See "The updated ogcli diff tool enables Opengear users to compare a proposed configuration with an existing configuration so that they may understand any prospective changes to the config. " on page 422

Diff tool behavior

- Diff shows additions, removals and changes clearly in the a streamlined format with only functional differences between the input and running configurations.
- If any section, list item or sub-property is out of order between the input configuration and the running configuration, it is not shown in the diff unless the values have actually changed.
- If the input configuration file is missing properties or sections of configuration, it shows the differences between running configuration and the default values for those properties.
- If any property or configuration section is missing from input configuration, and the running configuration is identical to the system defaults, it is omitted from the diff output.
- If diff function detects no differences between the current configuration and configuration template files match, a return code of 0 and no output is shown.

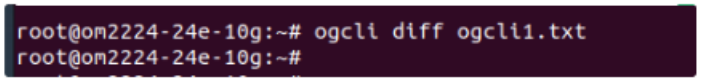
Syntax

```
diff <proposed_configuration_file>
```

Usage Examples

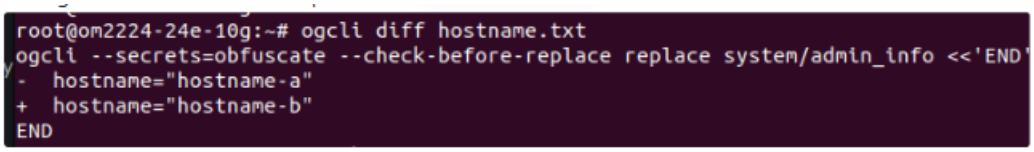
1. Active configuration matches input file:

No differences between input file and active configuration.



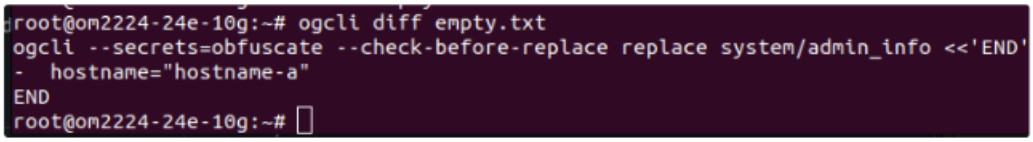
2. Configuration differs from template:

Differences displayed between input file and active configuration.



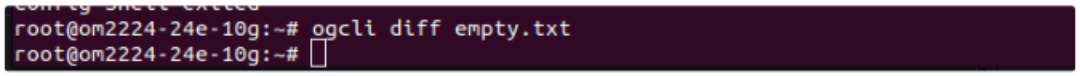
3. Configuration differs from template with defaults:

Differences between active configuration and default configuration because the input file was empty.



4. Configuration matches template with defaults:

No differences between active configuration and default configuration with empty input file.



Positional arguments

<input_file> Diff active configuration against <input_file>.

Options

`-h, --help` show this help message and exit.

If more detailed help is required, use `config diff --help`.

`--secrets` The --secrets flag can be used to control how sensitive fields are displayed in the diff output. By default, sensitive fields are obfuscated. If the proposed config file was exported with `--secrets=cleartext` or `--secrets=mask` then the same value must be used when running `ogcli diff`, for example, `ogcli --secrets=cleartext diff <input file>` If the input file contains a different `--secrets` parameter than that provided, an error will be returned.

DISCARD

Description

The discard command is used to remove unapplied changes.

This can be used to discard specific or configuration wide changes including:

- Updates to configuration items.
- Additions not applied.
- Items designated for deletion.

Parameters

`discard` - when used on its own discard the current item when in an item context, otherwise it will be an error.

`discard all` - when used with the 'all' command, then any changes staged in the current session will be dropped.

Syntax

```
discard [all]
```

Examples

The following commands create a user and then discard the user (it is never saved).

> **Note:** The context changes to exit the 'myuser' item since it no longer exists.

```
config: add user myuser
```

```
config: user myuser
```

```
config(user myuser): discard
```

Discard changes

config(user):

The following commands discard changes to an existing item. The item isn't removed in this case since it has been applied previously. The description field will revert back to whatever it was before.

```
config: user root
config(user root): description "Root user"
config(user root): discard
```

The following commands discard changes to multiple entities, the group and port entities. Both will be reverted:

```
config: edit group admin description "New group description"
config: edit port port01 label "New label"
config: discard all
```

Discard all changes

```
discard *
```

This will result in a confirmation being displayed.

DISCARD GROUPS OF CHANGES

```
discard auth user "username"
```

- If "username" is an addition that has not been applied, it will result in the added user being discarded. In this case the user will be prompted to confirm before the command is implemented.
- If "username" is an existing user with unapplied configuration changes, this will result in any changes there being discarded. A confirmation will be required.

- If “username” is an existing user but with no changes, the user will be informed that there are no configuration changes to discard.

DISCARDING SPECIFIC CHANGES

```
port port01
discard
```

- If the entity has unapplied changes it will be discarded.
- If there are no unapplied changes an information message is displayed.

Confirmation

Discarding changes at a section, or configuration wide level will give a warning that multiple changes will be discarded.

EDIT

Description

The edit command is used when making changes to configuration options without navigating through the hierarchy.

Parameters

`entity` - the entity to be edited.

`item-label` - unique value that identifies the item.

`record field` - the field to set for the item.

`value`- the value corresponding to the field.

Syntax

edit <entity> <optional-entity> <item-label> <field>

<value>

Examples

Consider the following change to a port label:

```
config
port
port_01
label "Office-switch"
```

Alternatively, consider making the change from the root of configuration mode.

```
config
edit port port_01 label "Office-switch"
```

EXIT

Description

The `exit` command can be run at any level in the configuration structure and will allow you to leave config mode. If there are unapplied changes, you are informed and asked to confirm it you wish to proceed.

Parameters

There are no parameters applicable to the exit command.

Syntax

```
exit
```

Example

```
exit
```

HELP (OR ?)

Description

> **Note:**Config mode will accept either `help` or a question mark `?` input.

Can be used in the following ways:

- A standalone command to view available options for the configuration section.
- In combination with a command to access help documentation.
- In combination with a configuration option to access help documentation and examples.

Parameters

The `help` command shows help for the current context.

`command` - shows help for the command.

`field` - shows help for the field.

Syntax

```
help <command or field>
```

```
<command or field> ?
```

Examples

The following will print help for the “port port01” context:

```
config(port port01): help
```

or

```
config(port port01): ?
```

The following will print help for the baudrate field when in the “port port01” context:

```
config(port port01): help baudrate
```

or

```
config(port port01): baudrate ?
```

HELP COMMAND USED STANDALONE

When used by itself, `help` or `?` returns a list of available commands or configuration options.

HELP USED IN CONJUNCTION WITH A COMMAND

```
apply ?
```

When used in conjunction with a command, `help` displays available sub-options.

For example, when running the `apply` command from the root config level, the `help` command notifies you that changes will traverse the configuration structure, however, when running the `help` command from within a configuration section, changes will apply to configuration options contained within.

```
add user ?
```

Displays help content including syntax and config items (mandatory and optional).

HELP USED WITH A CONFIGURATION OPTION

In the context of this example, the user is running the command from within the port configuration section and is wanting to get information on the available options.

```
pinout ?
```

This will display a list of available options.

```
label ?
```

This will display expected format and a sample.

IMPORT/EXPORT

Description

> **Note:** The import / export and associated commands operate in bash, ie. outside of config CLI. You must exit config to operate the import/export features.

The Import / Export feature allows you to export the current configuration to a file and import or restore the configuration from that file. An import will add configuration to the current configuration and restore will replace the current configuration with the contents of the configuration file.

Import

Running the import command (within bash, not in config:) will allow you to import a configuration script from an external source file. You should point the console server to a config file on specific network location. The file will be copied to the console server and run. Depending on how it has been set up, the changes can be automatically applied after the config file is run.

Export

Running the export command (within bash, not in config:) will allow you to generate a configuration script based on the existing configuration on the console server.

This command can be run at any level in the hierarchy and used to export either:

- The configuration across the node
- Configuration specific to the users's location in the hierarchy.

```
export all current config
```

Will display all config on the console server before it has been applied for copying.

```
export all saved config
```

Will display all saved config on the console server for copying.

```
export current config
```

Will display the config from the users's current position in the navigation hierarchy.

Parameters

Import and export are run from outside of the config shell. The config command is invoked from bash with different parameters to cause it to import or export the configuration without entering the config shell.

filename – The name of the file to be imported from or exported to. If omitted then `stdin` or `stdout` will be used.

Syntax

```
config export <optional filename>
```

```
config import <optional filename>
```

Examples

```
config export /tmp/console_server.config
```

```
config import /tmp/console_server.config
```

Positional arguments

{export,import,restore,merge,replace,get}

Positional Argument	Description
export	Export the current configuration.
import	Import config from a file.
restore	Restore config from a file.
merge	Merge a provided list with existing config.
replace	Replace a list or item.
get	Display an entity's associated values.
Options	

`-h, --help`	Show this help message and exit.
`--show-config`	Display the entire configuration and exit.
`-d`	Increase debugging (up to 3 times).
`-j`	Export in json format.
`--entities`	Display entities and exit.

Exporting to a file

Note: The import/export and associated commands operate in bash, ie. outside of config CLI. You must exit config to operate the import/export features.

SHOW

Description

The `show` command displays information relevant to the configuration section, including the highlighting of changes. The context in which the command is run will determine what is displayed.

At `config root`, the `show` command will display system information.

Within a config section, for example from **config > auth > user**, this will display a flat list of available users.

Parameters

show	Used on its own, will display the fields of the current context. When used in the top context, it shows the list of all entities. When used in an entity context, it shows the list of items in that entity. When used in an item context, it shows the fields and values of the current item.
entity	The entity to display, or to show details of.
item	The item to display or show details of.
field	The field to show the value of.

Syntax

show <optional entity> <optional item> <optional field>

Context

Examples using context

The following examples show how the output of the show command changes in accordance with context as it may be used at the config, physif, net1 contexts:

show - at the config context:

```
config: show

Entities
========
access_right                          pdus/drivers
auth                                  physif
auto_response/beacon                  port
auto_response/reaction                port_session
auto_response/status                  ports/auto_discover/schedule
auto_response/status/beacon-module    ports/status_port
cellfw/info                           services/brute_force_protection
cellmodem                             services/lldp
cellmodem/sim                         services/ntp
conn                                  services/routing
failover/settings                     services/snmp_alert_manager
failover/status                       services/snmpd
firewall/policy                       services/ssh
firewall/predefined_service           services/syslog_server
firewall/service                      services/tftp
firewall/zone                         ssh/authorized_key
group                                 static_route
ip_passthrough                        system/admin_info
ip_passthrough/status                 system/banner
ipsec_tunnel                          system/cell_reliability_test
lighthouse_enrollment                 system/cellular_logging
local_password_policy                 system/cloud_connect
logs/portlog                          system/diskspace
logs/portlog_settings                 system/info
managementport                        system/model_name
monitor/brute_force_protection/ban    system/serial_number
monitor/lldp/chassis                  system/session_timeout
monitor/lldp/neighbor                 system/ssh_port
monitor/static_routes/status          system/system_authorized_key
monitoring/alerts/networking          system/time
monitoring/alerts/power               system/timezone
monitoring/alerts/system              system/version
pdu                                   user

config:
```

show - at the physif context:

```
config: physif
config(physif): show
Item names for entity physif
  net1
  net2

config(physif):
```

show - at the net1 context:

```
config(physif): net1
config(physif net1): show
Entity physif item net1
  description NET1 - 1G Copper/SFP
  enabled     true
  mtu         1500
  dns (object)
    nameservers (array)
    search_domains (array)
  ethernet_setting (object)
    link_speed auto

config(physif net1):
```

Examples using parameters

The following examples show the output of the show command when used with different parameters:

```
config: show physif
Item names for entity physif
  net1
  net2

config: show physif net1
Entity physif item net1
  description NET1 - 1G Copper/SFP
  enabled     true
  mtu         1500
  dns (object)
    nameservers (array)
    search_domains (array)
  ethernet_setting (object)
    link_speed auto

config:


config: show physif net1 description
NET1 - 1G Copper/SFP
config:
```

Config

You can view the content of all configuration in JSON format.

You can also view the config of a specific section of the hierarchy you are in.

```
show-config
```

Directed Usage

You will also be able to look into a config sections using the show command. For example:

```
show auth user
```

Will display a flat list of users.

```
show auth user “username”
```

Will display the configuration for the user specified.

UP / EXIT / ..

Description

These commands allow users to traverse the configuration hierarchy.

```
up
```

The position will move one level up in the hierarchy.

If used at the root configuration level, it should point trigger the exit command.

Parameters

No parameters.

Syntax

```
up
```

```
exit
```

Examples

If, as in this example, the context is a specific port, then the ports entity can be accessed by using the `up` command then moving into another port:

```
config: port port01
config(port port01): up
config(port): port02
config(port port02):
```

HOW CHANGES ARE APPLIED OR DISCARDED

When fields and entities are changed, the changes are not immediately applied to the system configuration but remain in a staged status. Items that are staged are indicated by an '*' (asterisk) when the 'show' command is used. In addition, the 'changes' command can be used to show what fields have been changed.

In the following example, the user 'john' has been changed to alter the description. The 'show ' command indicates the changed field with an '*'. The changes command lists the changed field.

```
config(user john): description "Admin"
config(user john): show
Entity user item john
 description Admin * enabled true
 no_password false password false
 password
 ssh_password_enabled true
 groups (array)
```

APPLYING OR DISCARDING CHANGES

Once fields and entities have been changed, they are not yet applied to the system configuration but are kept staged. Items that are staged are indicated with an ‘*' when the ‘show’ command is used. In addition, the 'changes’ command can be used to show what fields have been changed.

When any changes have been made to a single or multiple entities, the following commands become available. These commands are described in detail in the Config CLI Commands section:

Command	Description
changes	Show staged changes on all entities.
apply	Apply changes only on the current entity.
discard	Discard changes only on the current entity.
apply all	Apply changes on all entities.
discard all	Discard changes on all entities.

Example

In the following example, the user ‘john’ has been changed to alter the description. The ‘show’ command indicates the changed field with an asterisk '*'. The changes command lists the changed field.

```
config(user john): description "Scrum Master"
config(user john): show
Entity user item john
description Scrum Master *
enabled true
no_password false
password
ssh_password_enabled true
groups (array)
config(user john): changes
Entity user item john (edit)
description Scrum Master
config(user john):
```

MULTI-FIELD UPDATES

Description

Within config shell, it is possible to update multiple fields with one command line. This is restricted to 'flat' fields within the current context ie arrays and sub-objects cannot currently be updated all in one command line.

For example, the following port fields can all be changed in a single command: `baudrate, databits, escape_char, label, logging_level, mode, parity, pinout` and `stopbits`. Other complex fields such as `control_code` and `ip_alias` cannot be modified from the port item context in one commands (multiple commands are needed).

Example

The following command sets the `baudrate, escape_char` and `label` fields.

```
config(port port01): baudrate 115200 escape_char ! label "My Router"
```

The changes will be staged in config shell. Use the `apply` command to save the changes to config.

To further update the `control_codes` and `ip_aliases`, multiple commands are required as follows:

```
config(port port01): control_code
config(port port01 control_code): break b chooser c
config(port port01 control_code): up
config(port port01): ip_alias
config(port port01 ip_alias): add
config(port port01 ip_alias 1): interface net1 ipaddress 10.83.0.6/24
config(port port01 ip_alias 1): up
config(port port01 ip_alias): up
```

```
config(port port01): changes
Entity port item port01 (edit)
  control_code (object)
    break b
    chooser c
  ip_alias (array)
    1 (object)
      interface net1
      ipaddress 10.83.0.6/24
config(port port01):
```

If certain fields are hidden and only visible by first configuring other fields, these hidden fields need to be set in another line. For example, the `kernel_debug` field is only revealed by setting the field `mode` of a port to `localConsole`, so this is configured on the next line:

```
config: port port03
config(port port03): mode localConsole baudrate 115200 databits 7
label aaa
logging_level eventsOnly parity even
config(port port03): kernel_debug true
```

Error Messages

If there is an error while processing a multiple-fields command, the staged values in configuration will not be changed. If there were no staged changes on the item, then no staged changes will appear. If there were already staged changes, then those staged changes will not be affected.

In the following example, the user description was previously changed to “my user”

```
config(user consoleuser): show
Entity user item consoleuser
  description         my user *
  enabled              true
  no_password false
  password            ""
  ssh_password_enabled true
  groups (array)
    0 consoleuser
```

If a bad field name or value is supplied on the command line, then the existing staged value is retained. The bad field name is highlighted using a ^ marker.

```
config(user consoleuser): description "My console user" invalid true
                                                          ^
Invalid input detected at '^' marker.
config(user consoleuser):
```

If the field is missing a value, a different error message is displayed:

```
config(user consoleuser): description "My console user" enabled
Incomplete command.
config(user consoleuser): show
Entity user item consoleuser
  description         my user *
  enabled true
  no_password false
  password             ""
```

```
ssh_password_enabled true
groups (array)
  0 consoleuser
```

The bad value for the field is indicated by an error message hinting the expected type of the value:

```
config(user consoleuser): description "My console user" enabled bad
Value bad for field enabled cannot be parsed as a boolean.
config(user consoleuser): show
Entity user item consoleuser
  description my user *
  enabled              true
  no_password         false
  password             ""
  ssh_password_enabled true
  groups (array)
    0 consoleuser
```

Changes to previous functionality

With the new `show` command, some previous syntax has changed. Just typing a field name is now an error condition. Previously this would be equivalent to the show command.

```
config: user root
config(user root): description
Incomplete command.
config(user root):
```

ERROR MESSAGES

When an error is made in the command line an error message which identifies the error is returned. For example, if the first token of the command is mistyped, the unknown command message is displayed.

```
config: usear root
There is no command usear root.
Type 'help' to see the available commands.
config:
config: aaaaa
There is no command aaaaa.
Type 'help' to see the available commands.
config:
```

If only the first few tokens of the command can be parsed, an error message with a ^ marker is displayed showing which part of the command cannot be parsed. If a context navigation is mistyped on the command line, then the context remains unchanged. It does not partially navigate through multiple contexts. In the following example, the context remains at the top context because `roopt` is not a valid item context in the user entity context.

```
config: user roopt
             ^
invalid input detected at '^' marker.
config:
```

STRING VALUES IN CONFIG COMMANDS

Description

The syntax for the use of string values has changed. It was previously possible to enter values containing spaces without using quotes. Multiple fields can now be assigned in one command line, quotes are required to keep field values together.

Example

The following example shows setting multiple fields where the field value for the description has spaces. The first attempt doesn't work because the second part of the description is interpreted as a field name. The second attempt is the correct syntax:

> **Note:**In the example the syntax error in the first line in highlighted in **bold** for clarity; Ihe correct syntax is highlighted in bold in line four.

```
config(user consoleuser): description My console user enabled true
There is no command description My console user enabled true.
Type 'help' to see the available commands.
config(user consoleuser): description "My console user" enabled true
config(user consoleuser): changes
Entity user item consoleuser (edit)
  description My console user
  enabled true
config(user consoleuser):
```

If the value itself must contain quotes, there is a triple quote form for entering string values:

```
config(user consoleuser): description """My "console" user""" enabled true
config(user consoleuser): changes
Entity user item consoleuser (edit)
  description My "console" user
  enabled true
```

The triple quoted string is used for entering multi-line strings:

```
config(system/banner): banner """
This is a banner that has
multiple lines.
"""
config(system/banner):
```

Error Messages

If the multi-line command string cannot be tokenised, an error message will be displayed in the following form:

```
config(system/banner): banner """
aaa
""""
Invalid input. Tokens must be separated by whitespace.
Check your input and try again.
config(system/banner):
```

CONFIG CLI USE CASE EXAMPLES

ADDING A USER

The following is a fully worked example showing the adding of a new user.

Note: In the following examples, some commentary has been added, the commentary is denoted with a '//' prefix. Where sessions continue onto the next page, this is shown with the comment "// session continues here:"

```
# config
Welcome to the Opengear interactive config shell. Type ? or help for help.
// Move to the user entity

config: user
config(user): help add
Add a new item for entity user.

The add command requires a unique value to identify the record.
This will be used for the username field.

Description for the item:
  Retrieve and update information for a specific user.

// Create the new user

config(user): add matt
config(user matt): show
Entity user item matt
```

```
  description

// Session continues here:
  enabled              true
  no_password          false
password                    (required)
ssh_password_enabled true
username             matt
groups (array)

// Fill out some fields

config(user matt): password topsecretpassword
config(user matt): description scrum master
config(user matt): show
Entity user item matt
  description          scrum master *
  enabled              true
password               topsecretpassword *
ssh_password_enabled   true
username               matt
groups (array)

// Edit the groups

config(user matt): groups
config(user matt groups): show
Entity user item matt field groups
config(user matt groups): add // Tab completion to show available values
```

```
admin myuser netgrp
config(user matt groups): add admin
config(user matt groups): up // Exit the groups list
// Session continues here:
// Show and apply

config(user matt): show
Entity user item matt
  description          scrum master *
  enabled              true
  password             topsecretpassword *
  ssh_password_enabled true
  username             matt
  groups  (array)
    0 admin *
config(user matt): apply
Creating entity user item matt.
config(user matt):
```

CONFIGURING A PORT

```
config: port
config(port): help
You are here: entity port

Description for the entity:
   Configuring and viewing ports information
```

```
Names (type <name> or help <name>)
==================================
USB-A USB-E USB-front-lower port03 port07 port11 port15 port19 port23
USB-B USB-F USB-front-upper port04 port08 port12 port16 port20 port24
USB-C USB-G port01          port05 port09 port13 port17 port21
USB-D USB-H port02 port06   port10 port14 port18 port22

Commands (type help <command>)
==============================
exit help show up

config(port): port01
config(port port01): baudrate // tab completion
110 1200 150 19200 230400 300 4800 57600 75
115200 134 1800 200 2400 38400 50 600 9600
config(port port01): baudrate 57600
config(port port01): label Router
config(port port01): control_code
config(port port01 control_code): break a
config(port port01 control_code): up
config(port port01): show
// Session continues here:
Entity port item port01
  baudrate        57600 *
  databits        8
  escape_char     ~
  label Router    *
  logging_level   disabled
  mode            consoleServer
```

```
  parity          none
  pinout          X2
  stopbits        1
  control_code  (object)
    break a *
    chooser
    pmhelp
    portlog
    power
    quit
  ip_alias (array)
config(port port01): apply
Updating entity port item port01.
config(port port01):
```

CONFIGURE A SINGLE SESSION ON A PORT

The feature is enabled by typing `single_session true`, then apply the change.

```
config(port port01):      single_session true
config(port port01):      apply
Updating entity port      item port01.
config(port port01):      show
Entity port item              port01
    baudrate       9600
...
single_session                true
...
ip_alias (array)
```

CREATE OR CONFIGURE A LOOPBACK INTERFACE

Loopbacks are not physical interfaces and as such cannot be attached to a firewall zone; firewall zone or policy rules must be created for whatever interface you are connecting over. Service translations can be created through the `firewall/service_translation` endpoint to change the source address of outbound packets to the loopback address.

To create a loopback, navigate to the `physifs` endpoint and set the media to `loopback`:

CREATE A LOOPBACK IN CONFIG SHELL

```
config: physif
config(physif): add loop
config(physif loop): media loopback
config(physif loop): enabled true
config(physif loop): apply
Creating entity physif item loop.
```

CREATE A LOOPBACK IN OGCLI

```
ogcli create physif << 'END'
device="loop"
enabled=true
media="loopback"
END
```

ADD AN ADDRESS TO A LOOPBACK INTERFACE

To add an address to a loopback interface, navigate to the `conns` endpoint and attach an ipv4 or ipv6 static address to the loopback (dhcp and ipv6_automatic are invalid for loopbacks):

ADD AN ADDRESS IN CONFIG SHELL

```
config: conn
config(conn): add new
config(conn new): mode static
config(conn new): physif loop
config(conn new): ipv4_static_settings
config(conn new ipv4_static_settings): address 10.0.0.1
config(conn new ipv4_static_settings): netmask 255.255.255.0
config(conn new ipv4_static_settings): apply
Creating entity conn item new.
```

ADD AN ADDRESS IN OGCLI

```
ogcli create conn << 'END'
mode="static"
physif="loop"
ipv4_static_settings.address="10.0.0.1"
ipv4_static_settings.netmask="255.255.255.255"
END
```

In the above example the `physif` is set to `loop`. Do not set the `broadcast_address` and `gateway_address` for loopback interfaces.

CREATE SOURCE NAT RULES

Note: When referring to service translation rules, we refer to translating the source ip of traffic to a desired source ip address. To change the source address of outbound packets for a particular service, a `service_translation` rule must be added, see the following example:

The following rule contains a list of outbound services along with the changed source address for the service packets. Navigating to the `firewall/service_translation` endpoint, you can add a new translation rule by using the `add` command. **Note:** Only services which use tcp or udp protocols are valid.

```
config(firewall/service_translation 10.0.0.1): show
Entity firewall/service_translation item 10.0.0.1
  address 10.0.0.1
  services (array)
     0 ssh
     1 https
```

If a service translation rule contains an address that does not exist on the box, a warning message is shown when creating the rule; however, it will not prevent these rules being created. See the following:

```
config(firewall/service_translation): add 10.0.0.2
  WARNING: The IP entered does not exist as a known IPv4 or IPv6
address.
  If this is expected, you can safely ignore this message.
```

If required, source NAT may be used for all tcp and udp traffic leaving the box by adding the service `all-tcp-udp` to the service list:

```
config(firewall/service_translation 10.0.0.1): show
Entity firewall/service_translation item 10.0.0.1
address 10.0.0.1
services (array)
0 all-tcp-udp
```

Note: There **must** be either a static or dynamic route to the loopback address from which you are connecting to the device.

Note: Source NAT is not used for packets on the cell interface wwan0. A VPN can be set up over the cell interface if the loopback address is used over cell; dynamic routing will need to be configured over the VPN to share the route to the loopback address.

REST API

The `firewall/service_translation` endpoint is used to create nftables rules which configure source NATs for outgoing service traffic. This replaces the outgoing IP address of a service packet with the address given in the `service_translation`. This is done for all services within the service translation rule.

```
"service_translation" : {
   "address": "A.B.C.D"
   "services": []
}
```

The address can be ipv4 or ipv6 (no netmask required), and does not need to exist on the box (a warning is presented if the address does not exist).

The list of services is a list of strings of service names. The outbound services must already be defined on the box, either as a predefined `firewalld` service or as a custom user service.

LOGGING AND DEBUGGING

You can ping the loopback address like any other interface. You will need a static or dynamic route to the loopback in order to reach it.

- Use the command `ip a` to display logging information.
- Conman logs information about creating or deleting loopback interfaces, and connections attached to loopback interfaces, in `/var/log/message`.
- When creating loopback interfaces, the generated files should be directed to `/etc/config/conman.conf`.
- Use the command `tcpdump` on interfaces connected to the device to see source NAT traffic.
- Source NAT rules can be found under `/etc/nftables/og-service-snat/og-service-snat.conf`, or use the command `nft list ruleset` to check for rules under the service SNAT tables.

CONFIGURE NET1 STATIC IPV4

```
conn default-conn-1 ipv4_static_settings
  address 192.168.2.54
  gateway 192.168.2.1
```

```
top
```

CONFIGURE NET2 STATIC IPV4

```
add conn net2-static-1 mode static physif net2
conn net2-static-1 ipv4_static_settings
  address 192.168.3.58
  gateway 192.168.3.1
  netmask 255.255.255.0
top
```

CONFIGURE NET3 STATIC IPV4 FOR OM2224-24E UNITS

```
add conn net3-static-1 mode static physif net3
conn net3-static-1 ipv4_static_settings
  address 192.168.4.58
  gateway 192.168.4.1
  netmask 255.255.255.0
top
```

CONFIGURE WIREGUARD THROUGH CONFIG SHELL

WireGuard is configured through Config Shell (or REST API). The minimum configuration of WireGuard is shown in the following:

1. Provide a name for the interface (wg0 in the example below).
2. Set enabled.

3. Set the private_key of your WireGuard interface.
4. Add an address (at least one) for your WireGuard interface (10.0.0.1/24 in this case).
5. Add a peer with the following parameters: endpoint_address, endpoint_port, public_key.
6. Add an allowed_ip for your peer. At least one - this is the WireGuard address(es) (as it can also accept an address range) of the other interface to which you are connected.

For example:

```
config: wireguard
config(wireguard): add wg0
config(wireguard wg0): private_key
AGiZvFHY+r/dD0rHSKU5ZCrHNdLM0W/h29VxobxWgFo=
config(wireguard wg0): enabled true
config(wireguard wg0): addresses
config(wireguard wg0 addresses): add 10.0.0.1/24
config(wireguard wg0 addresses): up
config(wireguard wg0): peers
config(wireguard wg0 peers): add
config(wireguard wg0 peers 0): public_key
o+quB4sbUAG2hEGSPpMNTnO0YSaQTP7dD+Q4IVjiCW8=
config(wireguard wg0 peers 0): allowed_ips
config(wireguard wg0 peers 0 allowed_ips): add 10.0.0.2/32
config(wireguard wg0 peers 0 allowed_ips): up
config(wireguard wg0 peers 0): endpoint_address 192.168.1.2
config(wireguard wg0 peers 0): endpoint_port 51820
config(wireguard wg0 peers 0): up
config(wireguard wg0 peers): top
```

ROOT USER PASSWORD - CLEARTEXT

```
edit user root password newpassword
```

ROOT USER PASSWORD = PASSWORD VIA SHA256

openssl passwd -5 password

Note:* this operation is not available in config shell

DEFINE PASSWORD COMPLEXITY RULES

```
edit local_password_policy
  password_complexity_enabled true
  password_expiry_interval_enabled true
  edit local_password_policy
  password_disallow_username true
  password_must_contain_number true
  password_must_contain_special true
  password_must_contain_upper_case true
```

HOSTNAME

```
edit system/admin_info hostname "OM2216-1-lab"
```

CONTACT INFO

```
edit system/admin_info
  contact "fred.bloggs@opengear.com"
  hostname "om2216-1.lab"
  location "Happy Valley Lab"
```

TIME ZONE AND NTP

```
edit system/timezone timezone "America/New_York"


edit services/ntp enabled true
services/ntp servers
  add
  value "74.207.242.234"
top
```

CREATE ADMIN USER

```
add user admin
  description "admin"
  enabled true
  no_password false
  password "password"
  user admin groups
  add "admin"
top
```

CREATE BREAKGLASS USER (BELONGS TO NETGRP)

```
add user breakglass
  description "breakglass" enabled true
  no_password false
  password "password"
  user breakglass groups
  add "netgrp"
top
```

ENABLE NETGRP - SET TO CONSOLEUSER

```
edit group netgrp enabled true
group netgrp ports
  add port01
  add port02
  add port03
  add port04
top
group netgrp access_rights
  add web_ui
  add pmshell
  delete admin
top
```

CHANGE SSH DELIMIITER TO : DEFAULT IS +

```
edit services/ssh ssh_url_delimiter ":"
```

CHANGE PORT LABELS

```
edit port port01 label "cisco1"
edit port port02 label "cisco2"
edit port port03 label "cisco3"
edit port port04 label "cisco4"
```

ENABLE TACACS - SET MODE TO REMOTELOCAL

```
edit auth mode "tacacs"
edit auth tacacsMethod "pap" tacacs
Password "tac_tests"
policy "remotelocal"
tacacsService "raccess"
auth tacacsAuthenticationServers
  add
  hostname "192.168.2.220"
  port 49
top
```

ENABLE LLDP ON NET1 & NET2

```
edit services/lldp enabled true
services/lldp physifs
  add "net1"
  add "net2"
top
```

ENABLE TFTP

```
edit services/tftp enabled true
```

ENABLE BOOT MESSAGES

Displays on local console port.

```
edit managementport ttyS0 kerneldebug true
```

DEFINE SESSION TIMEOUTS

```
edit system/session_timeout cli_timeout 100 serial_port_timeout 100 webui_timeout 100
```

Note: The inactivity timer starts only after you exit config shell, ie. it begins the count when you have left config and are at the bash command prompt.

DEFINE MOTD

Enter banner text within quotations.

```
edit system/banner banner """
```

ENABLE SIMM 1 ENABLE AND ADD APN

```
edit physif wwan0 enabled true
physif wwan0 cellular_setting
  apn hologram
top
```

ENABLE SIMM 1 COMPLETE END POINTS

```
edit physif wwan0 enabled true
physif wwan0 cellular_setting
  active_sim 1
  apn hologram
  iptype IPv4v6
  sim_failback_disconnect_mode ping
  sim_failback_policy never
  sim_failover_disconnect_mode ping
  sim_failover_policy never
top
physif wwan0 cellular_setting sims 0
  fail_probe_address 8.8.8.8
  fail_probe_count 3
  fail_probe_interval 600
  fail_probe_threshold 1
  failback_delay 60
  iptype "IPv4v6"
  slot 1
top
physif wwan0 cellular_setting sims 1
  fail_probe_address 8.8.8.8
  fail_probe_count 3
  fail_probe_interval 600
  fail_probe_threshold 1
  failback_delay 60
```

```
  iptype IPv4v6
  slot 2
top
```

ENABLE FAILOVER

```
edit failover/settings enabled true probe_address 192.168.2.1 probe_physif net1
```

ADD A SYSLOG SERVER

```
services/syslog_server
  add server1
  address 192.168.34.113
  protocol TCP
  port 610
  description "my syslog server"
top
```

Add Five Syslog Servers

Note: Due to page width limitations, in the following example, some command lines break over two lines.

```
add services/syslog_server server0 address 192.168.34.112 min_severity notice
port 514 port_logging_enabled true protocol UDP
add services/syslog_server server1 address 192.168.34.113 min_severity notice
port 514 port_logging_enabled true protocol UDP
add services/syslog_server server2 address 192.168.34.114 min_severity notice
```

```
port 514 port_logging_enabled true protocol UDP
add services/syslog_server server3 address 192.168.34.116 min_severity info
port 514 port_logging_enabled true protocol UDP
add services/syslog_server server4 address 192.168.128.1 description
"lighthouse-remote-syslog" min_severity info port 514 port_logging_enabled
true protocol UDP
```

SET PORT LOGGING REMOTE SYSLOG SETTINGS

```
edit logs/portlog_settings facility daemon severity infoEnable system
monitor snmp traps
```

ENABLE SYSTEM MONITOR SNMP TRAPS

```
monitoring/alerts/power power_supply_voltage_alert
  millivolt_lower 11000
  millivolt_upper 13000
  snmp
    enabled true
  up
top
monitoring/alerts/networking cell_signal_strength_alert
  enabled true
  threshold_lower 33
  threshold_upper 66
top
monitoring/alerts/system
  authentication_alert
    enabled true
  up
  config_change_alert
    enabled true
  up
  temperature_alert
    enabled true
    threshold_lower 35
    threshold_upper 67
  up
top
```

ENABLE SNMP V2 SERVICE FOR POLLING

```
edit services/snmpd enable_legacy_versions true
enable_secure_snmp false enabled true port 161 protocol UDP
edit services/snmpd rocommunity
"TkcxJAAAABBfDsigaxdDf7whb3sxKQKnjtCuuy/0COC6rE3lUu9ghg=="
```

ENABLE 2 SNMP TRAPS AND TRAP SERVERS

Note: Due to page width limitations, in the following example, some command lines break over two lines.

```
add services/snmp_alert_manager "snmp trap server 1" address 10.1.1.199 port
162 protocol UDP version v2c
services/snmp_alert_manager "snmp trap server 1"
  community "TkcxJAAAABBfDsigaxdDf7whb3sxKQKnjtCuuy/0COC6rE3lUu9ghg==" msg_type TRAP
  top
apply all


services/snmp_alert_manager 10.1.1.199:162/UDP
  name "snmp trap server 1" privacy_password secret auth_password secret
  top
apply all
```

CREATE A STATIC ROUTE

Note: Due to page width limitations, in the following example, some command lines break over two lines.

```
add static_route "static route test" destination_address 10.0.0.0
destination_netmask 8 interface net2
```

EDIT LAN (NET2) FIREWALL ZONE

(allow only source address traffic)

```
firewall/zone lan custom_rules
  add
    description "source_net4-1"
    rule_content "rule family=ipv4 source address=192.168.3.0/24 accept"
  up
  add
    description "source_net4-2"
    rule_content "rule family=ipv4 source address=10.202.198.0/27 accept"
  up
top
```

EDIT WAN (NET1) FIREWALL ZONE

(allow only source address traffic)

```
firewall/zone wan custom_rules
  add
    description "source_net4-1"
    rule_content "rule family=ipv4 source address=192.168.2.0/24 accept"
  up
  add
    description "source_net4-2"
    rule_content "rule family=ipv4 source address=192.168.4.0/24 accept"
  up
top
```

CUSTOM_RULE EXAMPLE FOR PORT AND PROTOCOL

```
add firewall/service myports label "My Serial Ports"
firewall/service myports
  add
    port 3001
    protocol tcp
  up
  apply
top
firewall/zone wan address_filters
  add
    source_address 10.10.2.0/19
    services
      add myports
    up
  up
top
```

ENROLL INTO LIGHTHOUSE

```
add lighthouse_enrollment lh1 address 2.21.99.188 bundle om2216-1 token password
```

OPENGEAR CLI GUIDE

The **ogcli** command line tool is used for getting and setting configuration, and for retrieving device state and information. The purpose of ogcli is perform a single operation and exit. Operations are performed on a single entity, a list of entities, or all entities. Entities in ogcli are collections of related information items that represent device state, information or configuration.

For a list of operations supported by ogcli, see the "ogcli Operations" section.

> **Note:**ogcli is not an interactive shell, it runs a single command and exits.

GETTING STARTED WITH OGCLI

The best way to get started with ogcli is to use the help command. Refer to the table below to access help topics within ogcli.

For detailed information about ogcli and how it works, view the ogcli help topic by running this command:

```
ogcli help ogcli
```

ACCESS OGCLI HELP AND USAGE INFORMATION

Help Command	Displays...
ogcli help	Basic ogcli help and usage information.

Help Command	Displays...
ogcli help help	Detailed information about the help command.
ogcli help operations	The full list of operations and a brief description of each.
ogcli help entities	The full list of entities and a brief description of each.
ogcli help syntax	How to get information into and out of ogcli.
ogcli help ogcli	More detailed information about the ogcli tool.
ogcli help usage	Common ogcli usage examples.
ogcli help secrets	Detailed information about controlling the display of secrets in ogcli.
ogcli help <operation>	A description and example usage of a specific ogcli operation.
ogcli help <entity>	A description of a specific entity and the operations it supports.
ogcli help <entity> <operation>	An example of how to perform a specific operation on a specific entity.

BASIC SYNTAX

The ogcli tool is always called with an operation, with most operations also taking one or more arguments specifying an entity for the operation to act on.

```
ogcli <operation> [argument] [argument]
```

OGCLI OPERATIONS

Operation	Description
create	Create an item.
export	Export the system configuration.
diff	Show additions, removals, changes and functional differences between the input and running configurations. See also `config diff`.
get	Retrieve a list or single item.
help	Display ogcli help.
import	Import system configuration, merging with current system configuration.

Operation	Description
merge	Merge a provided list with existing config.
replace	Replace a list or single item.
restore	Import system configuration, replacing the current system configuration.
update	Update an item, supports partial edits.

SUPPLYING DATA TO OGCLI

For operations that modify an entity (e.g. 'update') the new information can be passed as inline positional arguments, but this quickly becomes cumbersome when setting a large number of fields. Information can instead be supplied through stdin by piping the contents of a file, or with Here Document (heredoc) style. The heredoc style is the most flexible format and is used extensively in ogcli examples.

HERE DOCUMENT

A here document (heredoc) is a form of input redirection that allows entering multiple lines of input to a command. The syntax of writing heredoc takes the following form:

```
ogcli [command] << 'DELIMITER'
 HEREDOC
DELIMITER
```

- The first line starts with the ogcli command, followed by the special redirection operator << and a delimiting identifier. Any word can be used as the delimiter, commonly 'EOF' or 'END'.
- The `HEREDOC` block can contain multiple lines of strings, variables, commands or any other type of input. Each line can specify one field to update.
- The last line ends with the delimiting identifier used above, indicating the end of input.

```
ogcli update user <username> << 'END'
 description="operator"
 enabled=false
END
```

INLINE ARGUMENTS

Field data can be entered inline with the ogcli command as arguments, with each field separated by a space.

```
ogcli update user <username> enabled=false description=\"operator\"
```

PIPES AND STANDARD INPUT

The data can also be entered via `stdin` by piping the data to the ogcli command.

```
echo 'enabled=true description="operator"' | ogcli update user
<username>
```

Alternatively, you can provide a file via input redirection with <.

```
echo 'enabled=true description="operator"' > partial_record
```

```
ogcli update user <username> < partial_record
```

QUOTING STRING VALUES

All string fields require the argument to be specified with double quotes ". The shell can consume double quotes, so care must be taken when specifying strings to ensure the quotes are passed to ogcli as input.

1. Double quotes in heredoc do not need to be escaped.

```
ogcli update physif <device-identifier> << 'END'
 description="test network"
END
```

2. Double quotes within single quotes do not need to be escaped.

```
ogcli update physif user <username> 'description="test user"'
```

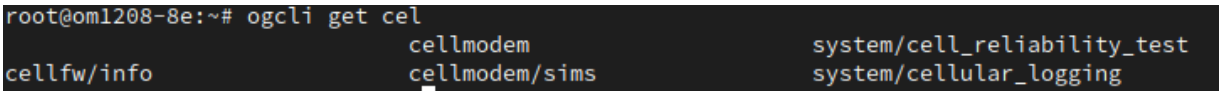
3. Double quotes not within single quotes need to be escaped.

```
ogcli update physif user <username> description=\"test user\"
```

TAB COMPLETION

ogcli includes tab completion to assist with typing commands. When entering the start of a command, press the **<tab>** key to complete the phrase to the nearest match.

If there are multiple matches, all options will be displayed for your reference.



DISPLAYING SECRETS IN OGCLI

Fields containing sensitive information are called **secrets**, which are handled specially by **ogcli** to obfuscate their values when they are displayed or exported.

Passwords and private keys are examples of secret fields.

The obfuscation process provides protection against "casual observation" only and offers no cryptographic security. The **obfusc** tool can be used to obtain the clear text version of any obfuscated secret generated by any Console Manager.

For more information, view the secrets help topic by running:

```
ogcli help secrets
```

The default behavior is for secrets to be passed to ogcli in clear text, and exported or displayed in obfuscated form.

For example, setting the password:

```
ogcli update services/snmpd auth_password=\"my secret\"
```

Retrieving the password (note, the output is abridged):

```
# ogcli get services/snmpd
auth_
password="TkcxJAAAABBSB3xoFWhPA6B7sDrzq3HwaTOAO/jsURqFa0qa7hc3TA=="
```

This behavior can be overridden to display sensitive fields in clear text, obfuscated form, or masked form using the **--secrets** option. The clear text and obfuscated forms are also accepted when supplying a sensitive field.

```
# ogcli --secrets=cleartext get snmpd
auth_password="my_secret"
```

```
# ogcli --secrets=obfuscate get snmpd
auth_password="my secret"
```

```
# ogcli --secrets=mask get snmpd
auth_password="********"
```

If an export is performed with the **--secrets=mask** option it is impossible to subsequently import the configuration, because the secrets have been removed.

COMMON CONFIGURATION EXAMPLES

These examples contain a variety of notations and usage patterns to help illustrate the flexibility of ogcli. The examples can be copied and pasted into the CLI.

REPLACE MESSAGE OF THE DAY (MOTD) DISPLAYED AT LOGIN

```
ogcli replace banner banner=\"updated message\"
```

RETRIEVE USER RECORD

```
ogcli get user <username>
```

UPDATE ITEM WITH FIELD WHERE VALUE IS A STRING

```
ogcli update user <username> description=\"operator\"
```

UPDATE ITEM WITH FIELD WHERE VALUE IS NOT A STRING

For example, a numeric or boolean value

```
ogcli update user <username> enabled=true
```

EXPORT SYSTEM CONFIGURATION

```
ogcli export <file_path>
```

IMPORT SYSTEM CONFIGURATION

```
ogcli import <file_path>
```

RESTORE SYSTEM CONFIGURATION

```
ogcli restore <file_path>
```

COMPARE CURRENT CONFIGURATION WITH A PROPOSED CONFIGURATION

The updated `ogcli diff` tool enables Opengear users to compare a proposed configuration with an existing configuration so that they may understand any prospective changes to the config.

The diff function performs a comparison of active configuration and an input configuration file, which must be in the format an export file produced by either a `config export <template-file>` or an `ogcli export <template-file>` operation. Any manual changes to this export file must include config or ogcli commands in a multi-line format using the 'END' heredoc marker as produced by an export. One line config or ogcli commands will not be accepted.

USING THE DIFF TOOL

The diff tool can be used by any user with administrator permission via the command line.

```
ogcli diff <input file>
```

or using config:

```
config diff <input file>
```

Note: `config diff` and `ogcli diff`, and can be used interchangeably using export files in either format.

If there are no differences between the active configuration and the input configuration file, the diff tool will not print any output, and the operation will have an exit code of 0.

```
root@om2248:~# ogcli export config_file
root@om2248:~# ogcli diff config_file
root@om2248:~# echo $?
0
```

The diff function will show any additions, removals and changes clearly in a streamlined format with only functional differences between the input and running configurations. Any additions that will be made to the active configuration are marked with a (+). For example, the `new_user` user does not exist in the active configuration, but is present in the input file supplied. If the input file was imported, this user would be added.

```
ogcli --secrets=obfuscate merge users <<'END'
+ users[1].enabled=true
+ users[1].groups[0]="admin"
+ users[1].no_password=false
+ users[1].ssh_password_enabled=true
+ users[1].username="new_user"
END
```

If the `new_user` user exists in the active configuration, but does not exist in the input file, this user will be removed if the input file was imported or restored. Removals will be marked with a (-) symbol.

```
ogcli --secrets=obfuscate merge users <<'END'
- users[1].enabled=true
- users[1].groups[0]="admin"
```

```
- users[1].no_password=false
- users[1].ssh_password_enabled=true
- users[1].username="new_user"
END
```

Changes in configuration between an item which exists in both the active configuration and the input file will also be displayed. The existing configuration will be marked with a (-) and the incoming change as a (+). In the example below, the `new_user` user belongs to the netgrp group on the device. However, if the input file is imported, it will belong only to the admin group.

```
ogcli --secrets=obfuscate merge users <<'END'
- users[1].groups[0]="netgrp"
- users[1].groups[1]="admin"
+ users[1].groups[0]="admin"
END
```

If any differences are found, the operation will have an exit code of 1. If there are any errors, the diff tool will have an exit code of 2.

Comparison to Default Values

If the input configuration file is missing properties or sections of configuration, the diff function will instead consider the differences between active configuration and the default values for those properties. Missing sections or properties from the input file will only be displayed in the diff tool output if the active configuration is different from the default system values. If any property or configuration section is missing from input configuration, and the running configuration is identical to the system defaults, it is omitted from the diff output.

How Secrets are Handled

The `--secrets` flag can be used to control how sensitive fields are displayed in the diff output. By default, sensitive fields are obfuscated. If the proposed config file was exported with `--secrets=cleartext` or `--secrets=mask` then the same value must be used when running `ogcli diff`.

```
root@om2248:~# ogcli --secrets=cleartext export config_file
root@om2248:~# ogcli --secrets=cleartext diff config_file
```

If the input file contains a different `--secrets` parameter than is passed to `ogcli diff` or `config diff`, an error will be returned:

```
root@om2248:~# config --secrets=cleartext export config_file
oot@om2248:~# config --secrets=cleartext diff config_file
root@om2248:~# config --secrets=mask diff config_file
The secrets flag provided doesn't match the flag in the proposed
config for physifs.
This error can be ignored with the --ignore-secrets-mismatch flag.
Type ogcli diff --help for more information.
```

This behaviour is the same for config:

```
root@om2248:~# config --secrets=cleartext export config_file
oot@om2248:~# config --secrets=cleartext diff config_file
root@om2248:~# config --secrets=mask diff config_file
The secrets flag provided doesn't match the flag in the proposed
config for physifs.
This error can be ignored with the --ignore-secrets-mismatch flag.
Type ogcli diff --help for more information.
```

The `--ignore-secrets-mismatch` flag can be used to ignore a difference in the `--secrets` parameter:

```
root@om2248:~# ogcli diff --ignore-secrets-mismatch config_file
```

Diff Tool Help

Basic help for `ogcli diff` can be accessed with `ogcli diff -h`. Similarly, help for `config diff` can be accessed with `config diff -h`. Detailed help for both diff tools can be accessed by `ogcli help diff`.

Limitations
JSON template files are no longer supported with config diff.
`ogcli diff` only supports input configurations generated by the same product SKU and software version.
Comments can be included between ogcli or config commands in the export file, but not within the commands or an error will be thrown. Comments must start with #. These will be ignored by the diff tool.

See also "diff" on page 356

ENABLE LOCAL CONSOLE BOOT MESSAGES

```
ogcli get managementports
```

```
ogcli update managementport mgmtPorts-1 kerneldebug=true
```

CREATE NEW USER

```
ogcli create user << 'END'
 description="superuser"
 enabled=true
 groups[0]="admin"
 password=”test123"
 username="superuser123"
END
```

CHANGE ROOT PASSWORD

```
ogcli update user root password=\"oursecret\"
```

CREATE NEW ADMINISTRATIVE USER

```
ogcli create user << 'END'
 username="adal"
 description="Ada Lovelace"
 enabled=true
 no_password=false
 groups[0]="groups-1"
 password="oursecret"
END
```

MANUALLY SET DATE AND TIME

```
ogcli update system/timezone timezone=\"America/New_York\"
```

```
ogcli update system/time time=\"15:30 Mar 27, 2020\"
```

ENABLE NTP SERVICE

```
ogcli update services/ntp << 'END'
 enabled=true
 servers[0].value="0.au.pool.ntp.org"
END
```

UPDATE SYSTEM HOSTNAME

```
ogcli update hostname hostname=\"system-hostname\"
```

ADJUST SESSION TIMEOUTS

```
ogcli update system/cli_session_timeout timeout=180
```

```
ogcli update system/webui_session_timeout timeout=180
```

SETUP REMOTE AUTHENTICATION WITH TACACS+

```
ogcli update auth << 'END'
 mode="tacacs"
 tacacsAuthenticationServers[0].hostname="192.168.250.21"
 tacacsMethod="pap"
 tacacsPassword="tackey"
END
```

SETUP REMOTE AUTHENTICATION WITH RADIUS

```
ogcli update auth << 'END'
 mode="radius"
 radiusAuthenticationServers[0].hostname="192.168.250.21"
 radiusAccountingServers[0].hostname="192.168.250.21"
 radiusPassword="radkey"
END
```

CREATE USER GROUP WITH LIMITED ACCESS TO SERIALPORTS

```
ogcli create group << 'END'
 description="Console Operators"
 groupname="operators"
 role="ConsoleUser"
 mode="scoped"
 ports[0]="ports-10"
 ports[1]="ports-11"
 ports[2]="ports-12"
END
```

VIEW AND CONFIGURE NETWORK CONNECTIONS

```
ogcli get conns
```

```
ogcli get conn system_net_conns-1
```

```
ogcli update conn system_net_conns-1 ipv4_static_
settings.address=\"192.168.0.3\"
```

```
ogcli create conn << 'END'
 description="2nd IPv4 Static Address Example"
 mode="static"
 ipv4_static_settings.address="192.168.33.33"
 ipv4_static_settings.netmask="255.255.255.0"
 ipv4_static_settings.gateway="192.168.33.254"
 physif="net1"
END
```

CONFIGURE SERIAL PORTS

```
ogcli get ports
```

```
ogcli get ports | grep label
```

```
ogcli get port ports-1
```

```
ogcli update port "port05" << 'END'
 mode="consoleServer"
 label="Router"
 pinout="X2"
 baudrate="9600"
 databits="8"
 parity="none"
 stopbits="1"
 escape_char="~"
 ip_alias[0].ipaddress="192.168.33.35/24"
```

```
 ip_alias[0].interface="net1"
 logging_level="eventsOnly"
END
```

CONFIG SHELL GUIDE

The Config Shell feature provides an interactive and familiar environment similar to older Opengear appliances. The result is an user-experience that feels like an Interactive CLI.

Advantages of the Config Shell are:

- Items can be created or updated without being applied immediately
- Items that are not applied are indicated by an asterisk (*) beside them when viewing information.
- Tab complete is supported for many commands.
- Built-in help (see "Global Context Commands" on page 436).
- Has a structured, tabular view when displaying lists of data.

START AND END A CONFIG SHELL SESSION

Start the config shell by typing `config` at a bash prompt. The bash prompt is presented to root and admin users when they log in via SSH or on the maintenance console.

You can exit the Config Shell by any of the following:

- Type `exit` to end the session.
- Send an EOF (**Control+D**).
- Send an INT (**Control+C**).

Note: The session is prevented from exiting if there are un-committed changes, this condition is indicated by a message. However, you can force an exit by immediately executing an exit command again, any un-committed changes will be discarded.

NAVIGATE IN THE CONFIG SHELL

The Config Shell operates in a hierarchy of entities. Due to the variety of entities, there are several ways for you to get to a place where you can make changes.

Starting at the root, enter the entity names to descend down through lower entities. Every entity name is an operation that descends into that entity. Similarly, type the names of entities higher in the hierarchy to ascend towards the root.

IDENTIFIERS:

Singleton entity	Require only the entity name to be uniquely identified.
List/item entity	The first level is the entity name, the second level is the item identifier (the identifier is the same identifier used by ogcli).
Multiple identifiers	A **single entity** (ssh/authorized_keys) requires an extra identifier. In this case, the hierarchy is: **ssh/authorized_keys > userid > [key_id]**.
Nested fields	The Config Shell treats nested fields as additional hierarchy levels. This applies both to arrays and maps. For arrays of complex values, each value shall also be a hierarchy level.

FIELDS, ENTITIES AND CONTEXTS

The config shell allows you to configure a number of fields which define settings.

The fields are grouped in entities that describe a small set of functionality. For example, there is a ‘user’ entity which is used to access user settings. Entities can contain sub-entities as well as simple fields.

CONTEXT WITHIN CONFIG SHELL

Once in the shell, a number of commands are available depending on the current context. The context is the current entity that is the focus of the config shell. When the shell is first started, the context is a special parent context from which sub-entities can be seen.

Once a context is selected by typing the name of the entity, it is shown in the prompt between brackets. For example, in the following snippet, the ‘user’ context is accessed and then the ‘john’ sub-entity is accessed causing the context to become ‘user john’. The ‘show’ command is used to list the entities and fields that descend from the current context.

```
config: user
config(user): show
Item names for entity user
 john matt myuser netgrp root
config(user): john
config(user john):
Entity user item john
 description
 enabled true
 no_password false
```

```
  password
  ssh_password_enabled true
  groups (array)
config(user john):
```

GLOBAL CONTEXT COMMANDS

The following commands are available on any context:

help (or '?')	Show help which is context sensitive. It will list some special details about the current context, the list of sub entities (or fields) and a list of available commands.
help <entity>	Show the help for the specific entity.
help <field>	Show the help for the specific field.
show	List the available entities and fields.
<entity>	Typing the name of an entity changes the context to focus on the named entity.
exit	Exit the command shell.

ENTITY CONTEXT COMMANDS

The following commands available on any entity context:

<field>	Show the value of a field.
<field> <value>	Change the field to the specific value.

delete	Delete the current entity. This is available when the context entity is an item in a list.
add	Append a sub entity or field to the current entity. This is only available when the context entity is a list.

APPLY OR DISCARD FIELD CHANGES

When fields and entities are changed, they are not yet applied to the system configuration but are kept staged. Items that are staged are indicated with an '*' when the `show` command is used. In addition, the `changes` command can be used to show what fields have been changed.

In the following example, the user 'john' has been changed to alter the description. The `show` command indicates the changed field with an '*'. The `changes` command lists the changed field.

```
config(user john): description "Admin"
config(user john): show
Entity user item john
 description Admin *
 enabled true
 no_password false
 password
 ssh_password_enabled true
 groups (array)
config(user john): changes
Entity user item john (edit)
 description Admin
config(user john):
```

OPERATIONS

Once a change has been made, the following commands are available:

changes	show staged changes on all entities
apply	apply changes only on the current entity
discard	discard changes only on the current entity
apply all	apply changes on all entities
discard all	discard changes on all entities

SUPPORTED ENTITIES

The following entities are supported in phase 1 of this feature and are available in release 22.06.0:

auth	Configure remote authentication, authorization, accounting (AAA) servers.
group	Retrieve or update user group information.
ip_passthrough	Passthrough entities are for retrieving / changing IP Passthrough settings.
ip_passthrough/status	The IP Passthrough status entity provides information about what part of the IP

	Passthrough connection process the device is currently at and information about the connected downstream device.
local_password_policy	Configure the password policy for local users. This includes expiry and complexity settings.
logs/portlog_settings	Check and update port log settings.
managementport	Used for working with local management console information.
port	Configure and view ports information.
ports/auto_discover/schedule	Manage Port Auto-Discovery scheduling.
system/admin_info	Retrieve or change the Console Manager appliance system's information (hostname, contact and location).
system/banner	Retrieve or change the Console Manager appliance system's banner text.
system/cloud_connect	Retrieve or change the Console Manager appliance system's cloud connect configuration.
system/model_name	Retrieve the Console Manager appliance's Model Name.
system/serial_number	Retrieve the Console Manager appliance's Serial Number.

system/session_timeout	Retrieve or change the Console Manager appliance session timeouts.
system/ssh_port	The SSH port used in Direct SSH links.
system/time	Retrieve and update the Console Manager's time.
system/timezone	Retrieve and update the system's timezone.
system/version	Retrieve the Console Manager's most recent firmware and REST API version.
user	Retrieve and update user information.

EXAMPLE CLI COMMANDS

ADDING A USER

In this example below, some commentary is added. Commentary added later is denoted with a '//' prefix.

```
# config
Welcome to the Opengear interactive config shell. Type ? or help for
help.
// Move to the user entity
```

```
config: user
config(user): help add
Add a new item for entity user.
```

```
The add command requires a unique value to identify the record.
This will be used for the username field.

Description for the item:
 Retrieve and update information for a specific user.

// Create the new user

config(user): add matt
config(user matt): show
Entity user item matt
 description
 enabled true
 no_password false
 password (required)
 ssh_password_enabled true
 username matt
 groups (array)
```

```
// Fill out some fields

config(user matt): password secretpassword
config(user matt): description Admin
config(user matt): show
Entity user item matt
 description Admin *
 enabled true
 password secretpassword *
```

```
 ssh_password_enabled true
 username matt
 groups (array)
```

```
// Edit the groups
config(user matt): groups
config(user matt groups): show
Entity user item matt field groups
config(user matt groups): add // Tab completion to show available
values
admin myuser netgrp
config(user matt groups): add admin
config(user matt groups): up // Exit the groups list
```

```
// Show and apply
config(user matt): show
Entity user item matt
 description Admin *
 enabled true
 password secretpassword *
 ssh_password_enabled true
 username matt
 groups (array)
 0 admin *
config(user matt): apply
Creating entity user item matt.
config(user matt):
```

CONFIGURING A PORT

```
config: port
config(port): help
You are here: entity port
Description for the entity:
  Configuring and viewing ports information

Names (type <name> or help <name>)
==================================
USB-A  USB-E  USB-front-lower  port03  port07  port11  port15  port19  port23
USB-B  USB-F  USB-front-upper  port04  port08  port12  port16  port20  port24
USB-C  USB-G  port01           port05  port09  port13  port17  port21
USB-D  USB-H  port02           port06  port10  port14  port18  port22

Commands (type help <command>)
==============================
exit  help  show  up
config(port): port01
config(port port01): baudrate // tab completion
110     1200    150     19200   230400  300     4800    57600   75
115200  134     1800    200     2400    38400   50      600     9600
config(port port01): baudrate 57600
config(port port01): label Router
config(port port01): control_code
config(port port01 control_code): break a
config(port port01 control_code): up
config(port port01): show
Entity port item port01
  baudrate      57600         *
  databits      8
  escape_char   ~
  label         Router        *
  logging_level disabled
  mode          consoleServer
  parity        none
  pinout        X2
  stopbits      1
  control_code (object)
    break   a *
    chooser
    pmhelp
    portlog
    power
    quit
  ip_alias (array)
config(port port01): apply
Updating entity port item port01.
config(port port01):
```

ADVANCED PORTMANAGER PM SHELL GUIDE

The Portmanager program allows you to access any serial port on the console server using `pmshell` commands.

- Routes network connection to serial ports.
- Checks permissions.
- Monitors and logs all the data flowing to/from the ports.
- Allows you to run power commands if the serial port is associated with a PDU outlet.

RUNNING `PMSHELL`

`pmshell` provides an environment that allows you to access and interact with serial ports via a number of command sequences. It lets you navigate between ports using the chooser command (`~m`). For example, you can use pmshell to connect to port 8 via the portmanager via the following command line sequence.

```
# pmshell -l port08
```

PMSHELL COMMANDS

When running `pmshell` there are a number of command sequences that you can use that begin with the ~ key.

Note: If you are connected to `pmshell` via SSH, you must add an additional ~ escape sequence.

Options	Name	Result
~c		The Single Session feature can be enabled or disabled by editing the single_session field in a given port. When a user port level administration access is logged in via pmshell, the port configuration menu can be accessed via any port by pressing the escape character (~ by default) followed by c (~c).
~b	break	Generates a BREAK on the serial port (if you're doing this over ssh, you'll need to type "~~b").
~h	portlog	Generates a history on the serial port. Displays the traffic logs for the port - must have port logging enabled.
~.	quit	Quits pmshell.
~p	power	Opens the power menu for the port. The port

Options	Name	Result
		must be configured for a PDU.
~u		Opens the list of user sessions, select by number to disconnect.
~m	chooser	Connects to the port menu - go back to the serial port selection menu.
~?	pmhelp	Displays help message.

CUSTOM CONTROL CODES FOR SERIAL PORTS

Custom control codes can be defined for ease of use per port or can be applied to all ports. For example, users could define a different Power Menu control code for every port, while having a single control code for View History that applies to all ports.

Custom control codes can be used by any user with access to the serial port. In order to run the shortcuts, the user presses the CTRL key + the keycode.

> **Note:**Only Admin users can specify short-cut control codes.

CONFIGURE CUSTOM CONTROL CODES

Admin users can configure control codes for any of the `pmshell` commands through the REST API, ogcli and the new interactive config shell.

Control code limitations are as follows:

- Cannot set multiple control codes for a port to use the same keycode
- The available key codes are a-z, excluding ‘i’ and ‘m’ as these can be triggered by commonly used keys TAB and BACKSPACE.

To disable a certain control code for an individual port, set the port’s control code to an empty string.

CONFIGURE CONTROL CODES FOR A SPECIFIED PORT (CLI EXAMPLES)

Control Codes Action	CLI Examples

Set control codes for a given port. In this example, the user sets multiple control codes for port 2	`ogcli update port port02 << 'END'`
	` control_code.break="b"`
	` control_code.chooser="c"`
	` control_code.pmhelp="h"`
	` control_code.portlog="l"`
	` control_code.power="p"`
	` control_code.quit="q"`
	`END`

Clear all control codes for a given port, in this example, port 2	`ogcli update port port02 << 'END'`
	` control_code.break=""`
	` control_code.chooser=""`
	` control_code.pmhelp=""`
	` control_code.portlog=""`
	` control_code.power=""`
	` control_code.quit=""`
	`END`

CONFIGURE A CONTROL CODE VALUE FOR ALL PORTS

To set a particular control code to one value across all serial ports, admin users can use the script `set-serial-control-codes` from the CLI as follows:

```
set-serial-control-codes CONTROL_CODE KEY
```

where:

- CONTROL_CODE - Must be one of the following values: `break, chooser, pmhelp, portlog, power` or `quit`.
- KEY - Must be a single lower case letter a-z excluding ‘i' and ‘m’ or an empty string designated by '' which is used to clear the control code.

CONTROL CODES FOR ALL PORTS VIA CLI (EXAMPLES)

Control Codes Action	CLI Examples
Set chooser control code to CTRL-a on all ports	`set-serial-control-codes chooser a`
Clear chooser control code on all ports	`set-serial-control-codes chooser ''`

DOCKER

Docker is a tool designed to make it easier to create, deploy, and run applications by distributing them in containers. Developers can use containers to package up an application with all of the parts it needs, like libraries and dependencies, and then ship it out as one package. Docker is running by default on the Console Manager. You can access commands by typing docker in the Local Terminal or SSH.

For more information on Docker, enter `docker --help`.

CRON

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

Usage:

crontab [options] file

crontab [options]

crontab -n [hostname]

OPTIONS:

-u <user> define user

-e edit user's crontab

-l list user's crontab

-r delete user's crontab

-i prompt before deleting

-n <host> set host in cluster to run users' crontabs

-c get host in cluster to run users' crontabs

-x <mask> enable debugging

To perform start/stop/restart on crond service:

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

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

To verify the current crond status:

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

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

```
crontab -l
```

To edit or create a custom crontab file:

```
crontab -e
```

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

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

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

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

Save and close the file.

INITIAL PROVISIONING VIA USB KEY

Also known as “ZTP over USB”, this feature allows provisioning an unconfigured (factory erased) unit from a USB storage device like a thumb drive.

The USB device must contain a filesystem recognized by the CM (currently FAT32 or ext4) with a file named `manifest.og` in the root directory. This file specifies which provisioning steps will be done. An article with a partial description of the file format is available here:Automated enrollment using USB.

The USB device can be inserted any time (before or after power is applied to the unit) and as long as the unit is unconfigured, the ZTP over USB process will be triggered. Here “unconfigured” has the same meaning as for ZTP: no changes made to the ogconfig data store.

> **Note:**Setting the root password on first log in counts as a config change.

The following manifest.og keys are implemented. This provides image installation, Lighthouse enrollment, and arbitrary script execution:

manifest.og contains <key>=<value> pairs. Recognized keys are:

image : Firmware image file name on the USB device's filesystem that will be flashed after boot once the image is validated

script : Configuration script to run

address : Primary Lighthouse address to enroll with

api_port : Optional port to use for the primary address when requesting enrollment

password : LH global or bundle enrollment password

bundle : Name of LH enrollment bundle

EULA AND GPL

The current Opengear End-User License Agreement and the GPL can be found at http://opengear.com/eula.

UI BUTTON DEFINITIONS

The table below provides a definition of the button icons used in the UI.

Button Icon	Definition
	Edit buttons
	Add item (eg. SNMP Manager)
	VLAN interface or create VLAN interface.
	Bonded interfaces or create new bond
	Bridged interfaces or create new bridge
	Standard network interface
	Cellular interface

	Interface with bridge
	Interface with bond
	Bin widget. **Delete** selected object.