NVIDIA UFM Enterprise User Manual v6.15.6

Table of contents

About This Document

NVIDIA ® UFM ® Enterprise is a powerful platform for managing InfiniBand scale-out computing environments. UFM enables data center operators to efficiently monitor and operate the entire fabric, boost application performance and maximize fabric resource utilization.

Software Download

To download the UFM software, please visit NVIDIA's Licensing Portal.

If you do not have a valid license, please fill out the NVIDIA Enterprise Account Registration form to get a UFM evaluation license.

Document Revision History

For the list of changes made to this document, refer to Document Revision History.

Release Notes

NVIDIA® UFM® is a powerful platform for managing InfiniBand scale-out computing environments. UFM enables data center operators to efficiently monitor and operate the entire fabric, boost application performance and maximize fabric resource utilization.

Key Features

UFM provides a central management console, including the following main features:

- Fabric dashboard including congestion detection and analysis
- Advanced real-time health and performance monitoring
- Fabric health reports
- Threshold-based alerts
- Fabric segmentation/isolation
- Quality of Service (QoS)
- Routing optimizations
- Central device management
- Task automation
- Logging
- High availability
- Daily report: Statistical information of the fabric during the last 24 hours
- Event management
- Switch auto-provisioning
- UFM-SDN Appliance in-service software upgrade

- Fabric validation tests
- Client certificate authentication
- IPv6 on management ports

> **Warning**
>
> Prior to installation, please verify that all prerequisites are met. Please refer to System Requirements.

> **Warning**
>
> The Logical Server Model Management feature is going to be deprecated in UFM v6.12.0.

Changes and New Features

This section lists the new and changed features in this software version.

> **Note**
>
> For an archive of changes and features from previous releases, please refer to Changes and New Features History.

No new features were introduced in this release.

Note

The items listed in the table below apply to all UFM license types.

Note

For bare metal installation of UFM, it is required to install MLNX_OFED 5.X (or newer) before the UFM installation.

Please make sure to use the UFM installation package that is compatible with your setup, as detailed in Bare Metal Deployment Requirements.

Unsupported Functionalities/Features

The following distributions are no longer supported in UFM:

- RH7.0-RH7.7 / CentOS7.0-CentOS7.7
- SLES12 / SLES 15
- EulerOS2.2 / EulerOS2.3
- Mellanox Care (MCare) Integration
- UFM on VM (UFM with remote fabric collector)
- Logical server auditing
- UFM high availability script - **/etc/init.d/ufmha** - is no longer supported
- The UFM Multi-site portal feature is no longer supported. The Multi-Subnet feature can be used instead
- The UFM Monitoring Mode is deprecated and is no longer supported as of UFM Enterprise version 6.14.0 (July release) and onwards

- Logical Elements tab - Removed as of UFM Enterprise v6.12.0
- Removed the following fabric validation tests: **CheckPortCounters & CheckEffectiveBER**

> **Note**
>
> In order to continue working with **/etc/init.d/ufmha** options, use the same options using the **/etc/init.d/ufmd** script.
>
> For example:
>
> Instead of using **/etc/init.d/ufmha model_restart,** please use **/etc/init.d/ufmd model_restart** (on the primary UFM server)
>
> Instead of using **/etc/init.d/ufmha sharp_restart,** please use **/etc/init.d/ufmd sharp_restart** (on the primary UFM server)
>
> The same goes for any other option that was supported on the **/etc/init.d/ufmha** script

Installation Notes

Supported Devices

Supported NVIDIA Externally Managed Switches

Type	Model	Latest Tested Firmware Version
NDR switches	• MQM9790	31.2010.6102
HDR switches	• MQM8790	27.2010.6102
EDR switches	• SB7790 • SB7890	15.2010.5108

Supported NVIDIA Internally Managed Switches

Type	Model	Latest Tested OS Version
NDR switches	• MQM9700	MLNX-OS 3.11.1014
HDR switches	• MQ8700 • MCS8500 • TQ8100-HS2F • TQ8200-HS2F	MLNX-OS 3.11.1014
EDR switches	• SB7700 • SB7780 • SB7800 • CS7500 • CS7510 • CS7520	MLNX-OS 3.10.5002

System Requirements

Bare Metal Deployment Requirements

Platform	Type and Version
OS and Kernel	64-bit OS: • RedHat 7.9: 3.10.0-1160.el7.x86_64 • RedHat 8.2: 4.18.0-193.el8.x86_64 • RedHat 8.4: 4.18.0-305.el8.x86_64 • RedHat 8.6: 4.18.0-372.9.1.el8.x86_64 • RedHat 9.0: 5.14.0-70.13.1.el9_0.x86_64 • CentOS 7.9: 3.10.0-1160.el7.x86_64 • Ubuntu 18.04: 4.15 • Ubuntu 20.04: 5.4.0 • Ubuntu 22.04: 5.15.0
CPU[a]	x86_64

Platform	Type and Version
HCAs	• NVIDIA ConnectX®-4 with Firmware 12.12.xxxx and above[c] • NVIDIA ConnectX®-4 Lx with Firmware 14.32.1010 • NVIDIA ConnectX®-5 with Firmware 16.19.1200 and above • NVIDIA ConnectX®-6 with Firmware 20.24.1000 and above • NVIDIA ConnectX®-7 with Firmware 28.33.1014 and above • NVIDIA Mezzanine Board with Four ConnectX-7 ASICs for Multi-GPU Connectivity (CEDAR) with Firmware 28.36.0394 and above • NVIDIA BlueField with Firmware 24.33.900 and above • NVIDIA BlueField-2 with Firmware 24.33.900 and above • NVIDIA BlueField-3 with Firmware 32.36.3058 and above
OFED[b]	• MLNX_OFED 5.X • MLNX_OFED23.x • MLNX_OFED24.x

Note

[a] CPU requirements refer to resources consumed by UFM. You can also dedicate a subset of cores on a multicore server. For example, 4 cores for UFM on a 16-core server.

[b] For supported HCAs in each MLNX_OFED version, please refer to MLNX_OFED Release Notes.

[c]UFM v6.15.0 is the last version to support NVIDIA ConnectX-4 adapter cards

Note

From RedHat 9* and onwards, packages with SHA1 signatures are no longer supported. The CONDA package binary is signed with SHA1 signatures and thus, CONDA will not be installed with RedHat 9*.

Two options are available to overcome this.

1. **Recommended Option:** Run the following command to install Conda (change gpgcheck from 1 to 0):

```
cat <<EOF > /etc/yum.repos.d/conda.repo
[conda]
name=Conda
baseurl=https://repo.anaconda.com/pkgs/misc/rpmrepo/conda
enabled=1
gpgcheck=0
gpgkey=https://repo.anaconda.com/pkgs/misc/gpgkeys/anaconda.asc
EOF

#install conda
yum install conda
```

2. **Alternative Option:** Run the following command to set the RedHat 9* system-wide cryptographic policy to use **legacy** (less-secured) policy:

```
update-crypto-policies --set LEGACY
```

   Install Conda as instructed by the UFM installation script.

   After Conda installation, the policy can be set back to default by running the following command:

```
update-crypto-policies --set DEFAULT
```

Note

For running SHARP Aggregation Manager within UFM, it is recommended to use MLNX_OFED-5.4.X version or newer.

Note

Installation of UFM on minimal OS distribution is not supported.

Note

UFM does not support systems in which NetworkManager service is enabled.

Before installing UFM on RedHat OS, make sure to disable the service.

Docker Installation Requirements

UFM Docker Container is supported on the standard docker environment (engine).

The following operating systems were tested with Docker Container:

Component	Type and Version
Supported OS	• RHEL7 • RHEL8 • Ubuntu18.04 • Ubuntu20.04 • Ubuntu22.04

UFM Server Resource Requirements per Cluster Size

Fabric Size	CPU Requirements*	Memory Requirements	Disk Space Requirements	
			Minimum	Recommended
Up to 1000 nodes	4-core server	4 GB	20 GB	50 GB
1000-5000 nodes	8-core server	16 GB	40 GB	120 GB
5000-10000 nodes	16-core server	32 GB	80 GB	160 GB
Above 10000 nodes	Contact NVIDIA Support			

UFM GUI Client Requirements

The platform and GUI requirements are detailed in the following tables:

Platform	Details
Browser	Edge, Internet Explorer, Firefox, Chrome, Opera, Safari
Memory	• Minimum: 8 GB • Recommended: 16 GB

MFT Package Version

Platform	Details
MFT	Integrated with MFT version mft-4.26.1-6

UFM SM Version

Platform	Type and Version
SM	UFM package includes SM version 5.17.1

> **Note**
>
> Assuming the SM is connected to the production cluster, it can handle any events (IB traps) coming from the fabric that is being built; such events should not affect the routing on the production cluster. If events occurred in the production cluster, the routing could be changed.
>
> However, NVIDIA recommends isolating fabric sections to allow faster bring-ups, **faster troubleshooting and misconfiguration avoidance** that can cause routing errors. Isolation provides clearer SM and CollectX logs, avoiding warnings/errors from masking real production issues.

UFM NVIDIA SHARP Software Version

Platform	Type and Version
NVIDIA® Scalable Hierarchical Aggregation and Reduction Protocol (SHARP)™	UFM package includes NVIDIA SHARP software version 3.5.2

Software Update from Prior Versions

The installer detects versions previously installed on the machine and prompts you to run a clean install of the new version or to upgrade while keeping user data and configuration unchanged.

The upgrade from previous versions maintains the existing database and configuration, allowing a seamless upgrade process.

Info

Upgrading UFM Enterprise software version is supported up to two previous GA software versions (GA -1 or -2).

For example, if you wish to upgrade to UFM Enterprise v6.11.0, it is possible to do so only from UFM Enterprise v6.9.0 or v6.10.0.

For detailed installation and upgrade instructions, refer to the *UFM Quick Start Guide*.

Note

Due to a possible conflict, SM and SHARP installed by the MLNX_OFED must be uninstalled. The installation procedure will detect and print all MLNX_OFED packages that must be removed.

Note

It is recommended to upgrade to the latest UFM version from the last 2 GA releases that came before it. Upgrading from older UFM versions may result in failures.

Bug Fixes in This Release

Ref #	Description
3851344	**Description**: Fixed the issue with multiple ports going down simultaneously.
	Keywords: Multiple, Ports, Down
	Discovered in Release: v6.15.1
3854288	**Description**: Fixed issue with long debug messages not being sent to remote syslog.
	Keywords: Debug, Message, Remote, Syslog
	Discovered in Release: v6.15.4
3910040	**Description**: Fixed issue where UFM events (Links is up / Link is Down) were not shown in the remote syslog
	Keywords: UFM Event, Remote, Syslog
	Discovered in Release: v6.15.1
3855246	**Description:** Fixed the issue where the new master node initialization would fail after a power cycle of the old master node.
	Keywords: Master, Fail, Power, Cycle
	Discovered in Release: v6.15.1

Known Issues in This Release

N/A

Info

For a list of known issues from previous releases, please refer to Known Issues History.

Changes and New Features History

Note

The items listed in the table below apply to all UFM license types.

Feature	Description
Rev 6.15.2	
UFM SM	New routing algorithm for asymmetric QFT topologies
Rev 6.15.1	
SHARP Reservation	Added support for Auto-cleanup of zombie SHARP reservations
Rev 6.15.0	
Defining Node Description	To prevent the formation of incorrect multi-NIC groups based on these default labels, this feature offers the option to establish a blacklist containing possible node descriptions that should be avoided when grouping Multi-NIC HCAs during host startup. For more information, refer to Defining Node Description Black-List.
Network Reports	Added the ability to view topology change events related to devices and links. For more information, refer to Events History, Device Status Events and Link Status Events.
User Authentication	Introduced a new user authentication login page. For more information, refer to Azure Authentication Login Page and Enabling Azure AD Authentication.
	Added support for a separate authentication server. For more information, refer to UFM Authentication Server and Enabling UFM Authentication Server.
Secondary Telemetry	Added the ability to expose SHARP telemetry in UFM Telemetry. For more information, refer to Exposing Switch Aggregation Nodes Telemetry.
	Added the ability to stop SHARP telemetry endpoint using CLI commands. For more information, refer to Stopping Telemetry Endpoint Using CLI Command.

Feature	Description
REST APIs	Enhanced the logging REST API by adding the ability to get event logs in JSON file format. For more information, refer to Get Events Logs in JSON Format.
	Added the ability to expose managed switch power consumption in Web UI. For more information, refer to Get Managed Switches Power Consumption.
	Added ability to filter the event logs by source. For more information, refer to Create Log History.
	Added the ability to generate enterprise network reports. For more information, refer to Events History, Device Status Events and Link Status Events.
	Introduced REST APIs for various authentication types. For more information, refer to Examples of REST APIs Using Various Authentication Types.
	Added the ability to update UFM Configuration REST API. For more information, refer to UFM Configuration REST API.
	Added the option to expose cable information. For more information, refer to Get Ports with Cable Information.
	Improved dynamic telemetry by adding the ability to instantiate a new instance and delete a running instance. For more information, refer to UFM Dynamic Telemetry Instances REST API.
	Added the option to set “down” ports as unhealthy. For more information, refer to Unhealthy Ports REST API.
	Added forge InfiniBand anti-spoofing support. For more information, refer to Forge InfiniBand Anti-Spoofing REST API.
	Added the ability to expose the "`site_name`" field in all supported REST APIs. For more information, refer to REST API Complementary Information.

Feature	Description
Plugins	Added support for the gNMI-Telemetry plugin that employs the gNMI protocol to stream data from UFM telemetry. In addition, added support for secure mode based on client authentication. For more information, refer to the GNMI-Telemetry Plugin.
	Added support for ALM configuration for controlling isolation/de-isolation. For more information, refer to ALM Configurations.
	REST over RDMA Plugin: Moved to Ubuntu 22-based docker container, OFED 5.8-3.0.7.0, ucx_py 0.35.0 and Python 3.10.
Supported Transceivers	Added support for FR4 transceivers
Rev 6.14.2	
Cable and Transceivers Burning	UFM supports second-source cable transceivers burn.
Module REST API	Added HW revision field in GET module REST API response.
Telemetry	Added support for the MRCS register read in UFM Telemetry.
UFM Reports	UFM Daily report will be disabled by default after upgrade or clean installation.
Rev 6.14.0	
UFM Upgrade	Added support for in-service upgrade procedure for UFM HA. Refer to the following sections: • Upgrading UFM on Bare Metal - High Availability Upgrade • Upgrading UFM Container in High Availability Mode
User Authorization	Added support for user-defined roles based on REST APIs subsets. Refer to Rest Roles Access Control.
User Authentication	Added support for user authentication based on Azure Active Directory. Refer to Azure AD Authentication.
Plugins Management	Added support for loading UFM plugin to both master and standby nodes in case of UFM HA deployment. Refer to Plugin Management.
Unhealthy Ports Policy Management	Added support for unhealthy ports policy management via UFM Web UI. Refer to Health Policy Management.

Feature	Description
REST over RDMA Plugin	Added support for remote ibdiagnet authentication. Refer to rest-rdma Plugin.
SHARP Reservation	Added support for synchronous SHARP reservation REST API (in addition to the existing asynchronous REST API). Refer to the NVIDIA SHARP REST API.
Secondary Telemetry	Added support for secondary telemetry running by default upon UFM startup, fetching NVIDIA Amber counters. Refer to Secondary Telemetry.
	Added support for down ports telemetry. Refer to Secondary Telemetry.
PCI Analysis	Added support for PCI analysis as part of UFM Fabric Analysis Report (added new events for degraded hosts PCI devices). Refer to Appendix - Supported Port Counters and Events.
UFM System Dump	Added human readable time to the dmsg de-message output as part of UFM system dump.
Factory Reset	Added support for UFM Factory Reset. Refer to Appendix - UFM Factory Reset.
Rev 6.13.0	
Network Fast Recovery	Added the ability to automatically isolate a malfunctioning switch port as detected by the switch. Refer to Enabling Network Fast Recovery
Multi-Subnet UFM	Added support for multiple UFM instances, wherein multiple instances are aggregated, managed and controlled by a centralized UFM instance. Refer to Multi-Subnet UFM.
Switch ASIC Failure Detection	Added support for a new indication (UFM event) that identifies a failure of a specific switch ASIC. Refer to Configuring Partial Switch ASIC Failure Events.
UFM High-Availability Enhancements	Added support for configuring high-availability with dual-link connections to improve the high-availability robustness.
Automatic Switch Grouping	Added support for enabling automatic grouping of 1U switches by UFM, as per a pre-defined user-configured mapping. Refer to Appendix - Switch Grouping.
SHARP Trees APIs	Incorporated support for a new UFM REST API that presents the current active SHARP trees. Refer to NVIDIA SHARP Resource Allocation REST API.

Feature	Description
SHARP Reservation APIs	Added support for SHARP Reservation API enhancements. Refer to NVIDIA SHARP Resource Allocation REST API.
Operating System Update support	Implemented functionality to support the installation and upgrade of a standalone UFM after the upgrade of operating system packages (e.g., using yum update/apt upgrade). Furthermore, upgrading operating system packages will not impact a standalone UFM installation.
Email Time-Zone Settings	Added the ability to configure time-zone settings for UFM email notifications, ensuring that sent events or daily reports align with the configured time zone. Refer to Email.
Switch Connectivity Failure Indication	Incorporated support for a new UFM event indication that identifies failed communication with a specified managed switch. Appendix - Supported Port Counters and Events
Dynamic Telemetry	Added APIs that enable the creation and management of UFM Telemetry instances, allowing users to select desired counters and ports as per their requirements. Refer to UFM Dynamic Telemetry Instances REST API.
TFS (Telemetry Fluent Streaming) Plugin	Added support for UFM telemetry data streaming from multiple endpoints to Fluent Bit. Refer to Telemetry to Fluent Streaming (TFS) Plugin REST API.
	Added support for enabling white/black counters lists within the TFS Plugin. Refer to Telemetry to Fluent Streaming (TFS) Plugin REST API.
DTS (DPU Telemetry) Plugin	Added support for displaying DPUs data within the UFM Web UI. Refer to DTS Plugin.
Cyber-AI Plugin	Added support for displaying Cyber-AI software within the UFM Web UI. Refer to UFM Cyber-AI Plugin.
Packet Mirroring Collector (PMC) Plugin	Added the Packet Mirroring Collector (PMC) plugin that allows users to catch and collect mirrored pFRN and congestion notifications from switches for enhanced real-time network visibility. Refer to Packet Mirroring Collector (PMC) Plugin.
SNMP Traps Listener Plugin	Added the capability to enable registration and monitoring of SNMP traps from managed switches, in addition to updating UFM with the relevant trap information. Refer to SNMP Plugin.

Feature	Description
Bright Cluster Integration Plugin	Added support for integration of data from Bright Cluster Manager (BCM) into UFM, providing a more comprehensive network perspective. Refer to UFM Bright Cluster Integration Plugin.
UFM System Dump	UFM System Dump collection enhancement. Refer to UFM System Dump Tab.
Expanding Non-Blocking Fabric (NDT Plugin extension)	Added a feature that facilitates seamless expansion of the IB fabric, ensuring uninterrupted functionality and optimal performance throughout the fabric. Refer to NDT Format – Merger.
PDR (Packet Drop Rate) Plugin	Added a new functionality that enables automatic detection and isolation of port failures through monitoring of PDR (Packet Drop Rate), BER (Bit Error Rate), and high cable temperatures. Refer to PDR Deterministic Plugin.
Rev 6.12.0	
Managed Switches - Sysinfo Mechanism	Added the ability to save switches inventory data into JSON format files and present the latest fetched switches data upon UFM start-up. The saved switches data is available UFM upon system dump. Refer to Appendix - Managed Switches Configuration Info Persistency
REST over RDMA Plugin	Introduced security improvements (allowed read-only options in remote ibdiagnet) and added support for Telemetry API. Refer to rest-rdma Plugin.
Events and Notifications	Added support for indicating potential switch ASIC failure by detecting a defined percentage of unhealthy switch ports. Refer to Additional Configuration (Optional)
SHARP AM Multi-Port	Added support for detecting IB fabric interface failure and automatic failover to an alternative active port in SHARP Aggregation Manager (AM). Refer to Multi-port SM
UFM System Dump	Added support for downloading the generated UFM system dump. Refer to UFM System Dump Tab
UFM REST API	Added support for adding or removing hosts to Partition key (PKey) assignments (when adding/removing hosts, all the related host GUIDs are assigned to/removed from the PKey). Refer to Add Host REST API
	UFM System Dump Improvements including Creating New System Dump API

Feature	Description
UFM SLURM Integration	Enhanced UFM SLURM integration; allow flexible configuration of PKey and SHARP resources usage. Refer to Appendix - UFM SLURM Integration
UFM HA	Improved UFM HA configuration by setting UFM HA nodes using IP addresses only (removed the need of using hostnames and sync interface names). Refer to Configuring UFM Docker in HA Mode and Installing UFM Server Software for High Availability
Managed Switch Operations	Added support for persistent enablement/disablement of managed switches ports. Refer to Ports Window
UFM SDK	Created a script to get TopX data by category. Refer to UFM Aggregation TopX README.md file
Proxy Authentication	Added option to delegate authentication to a proxy. Refer to Delegate Authentication to a Proxy
UFM Initial Settings	Removed the requirement to set the IPoIB address to the main IB interface used by UFM/SM (gv.cfg → fabric_interface)
Port auto-isolation	Symbol BER **warning** does not trigger port auto-isolation, only symbol BER **error**
MFT Package	Integrated with MFT version 4.23.0-104
Rev 6.11.0	
UFM Discovery and Device Management	• InBand autosicovery of switchs' IP addresses using `ibdiagnet` • Discovering the device's PSID and FW version using `ibdiagnet` by default instead of using an SM vendor plugin
CPU Affinity	Enabling the user to control CPU affinity of UFM's major processes
gRPC API	Added support for streaming UFM REST API data over gRPC as part of new UFM plugin. Refer to GRPC-Streamer Plugin
Telemetry	• Added support for flexible counters infrastructure (ability to change counter sets that are sampled by the UFM) • Updated the set of available counters for Telemetry (removed General counters from default view: Row BER, Effective BER and Device Temperature. Now available through the secondary telemetry instance). Refer to Secondary Telemetry

Feature	Description
EFS UFM Plugin	Added support for streaming UFM events data to FluentD destination as part of a new UFM plugin. Refer to UFM Telemetry FluentD Streaming (TFS) Plugin
General UI Enhancements	• Displayed columns of all tables are persistent per user, with the option to restore defaults. Refer to Displayed Columns • Improved look and feel in Network Map. Refer to Network Map• Added Reveal Uptime to the general tab in the devices information tabs. Refer to Device General Tab
High Availability Deployment	• Added support for joining a new UFM device into the HA pair without stopping the UFM HA (in case of a secondary UFM node permanent failure). For more information, refer to Installing UFM Server Software for High Availability • Changed UFM HA package installation command parameters. For more information, refer to Installing UFM Server Software for High Availability
REST APIs	Added support for PKey filtering for default session data. Refer to Get Default Monitoring Session Data by PKey Filtering.
	Added support for filtering session data by groups. Refer to Monitoring Sessions REST API.
	Added support for resting all unhealthy ports at once. Refer to Mark All Unhealthy Ports as Healthy at Once
	Added support for presenting system uptime in UFM REST API. Refer to Systems REST API.
Deployment Installation	UFM installation is now based on Conda-4.12 (or newer) for python3.9 environment and third party packages deployments.
NVIDIA SHARP Software	Updated NVIDIA SHARP software version to v3.1.1.
UFM Logical Elements	UFM Logical Elements (Environments, Logical Servers, Networks) views are deprecated and will no longer be available starting from UFM v6.12.0 (January 2023 release)
Rev 6.10.0	
System health enhancements	Add support for the periodic fabric health report, and reflected the ports' results in UFM's dashboard
UFM Plugins Management	Add support for plugin management via UFM web UI

Feature	Description
UFM Extended Status	• Add support for showing UFM's current processes status (via shell script) • Added REST API for exposing UFM readiness
Failover to Other Ports	Add support for SM and UFM Telemetry failover to other ports on the local machine
UFM Appliance Upgrade	Added a set of REST APIs for supporting the UFM Appliance upgrade
Configuration Audit	Add support for tracking changes made in major UFM configuration files (UFM, SM, SHARP, Telemetry)
UFM Plugins	Add support for new SDK plugins
Telemetry	Add support for statistics processing based on UFM telemetry csv format
UFM High Availability Installation	UFM high availability installation has changed and it is now based on an independent high availability package which should be deployed in addition to the UFM Enterprise standalone package. for further details about the new UFM high availability installation, please refer to - Installing UFM Server Software for High Availability

Bug Fixes History

Ref. #	Description
Rev 6.15.1	
3670183	**Description:** Monitoring endpoint not returning counters for an active interface
	Keywords: Monitoring, Active Interface, Counters
	Discovered in release: v6.15.0
3670182	**Description:** Inconsistent port format type returned from the UFM
	Keywords: Inconsistent**,** Port, Format Type
	Discovered in release: v6.14.1

Ref. #	Description
3666944	**Description:** Port auto isolation failed to activate when a port consistently exhibited a high Symbol BER (1e-7)
	Keywords: Port Auto Isolation, Symbol BER
	Discovered in release: v6.13.1
3665316	**Description:** The UFM REST API endpoint `/ufmRest/resources/ports` provide inaccurate port state information
	Keywords: Ports REST API, Port State
	Discovered in release: v6.14.1
3604194	**Description:** UFM Fabric Validation "`CheckPortCounters`" failure
	Keywords: Fabric Validation, `CheckPortCounters`
	Discovered in release: v6.13.2
Rev 6.15.0	
3665001	**Description:** UFM Web UI does not display Network Map (stuck with "please wait" message)
	Keywords: Web UI, Network Map
	Discovered in release: v6.14.1
3644553	**Description**: When querying the ports, adding a `cable_info=true` as an argument will give cable information per port
	Keywords: Ports, Query, `cable_info=true`
	Discovered in release: v6.14.0
3604212	**Description:** Broken links REST API
	Keywords: REST API, Broken link
	Discovered in release: v6.13.2
3604183	**Description:** UFM error UFM NOT performed OpenSM polling for fabric changes more than 230742 seconds
	Keywords: OpenSM, UFM Error
	Discovered in release: v6.13.2-5

Ref. #	Description
3604021	**Description:** UFM Enterprise installation under Ubuntu 22.04 fails on `configure_ha_nodes.sh`
	Keywords: Ubuntu 22.04, Installation, `configure_ha_nodes.sh`
	Discovered in release: v6.14.1-5
3587849	**Description:** OpenSM restarted when backup UFM lost power
	Keywords: OpenSM, Restart
	Discovered in release: v6.9
3577427	**Description:** UFM REST API returns wrong switch type for NDR unmanaged switch
	Keywords: Unmanaged Switch, NDR, REST API
	Discovered in release: v6.13.1
3575882	**Description:** UFM event is not generated for a switch down
	Keywords: UFM Event, Switch Down
	Discovered in release: v6.13.1
3628421	**Description:** UFM Web UI timezone issue when selecting Local Time
	Keywords: Timezone, Web UI, Local Time
	Discovered in release: v6.14.1-5
3566193	**Description:** Request for docker UFM HA support on Debian OS 10.13
	Keywords: Docker, HA support, Debian
	Discovered in release: v6.14.1-5
3565820	**Description:** UFM container CLI bugs
	Keywords: CLI, Container
	Discovered in release: v6.13.2-5
Rev 6.14.0	
3590777	**Description**: After upgrading UFM new telemetry data is not being collected and presented in UI Telemetry tab.
	Keywords: Telemetry, Coredump
	Discovered in release: 6.14.0

Ref. #	Description
Rev 6.13.2	
3228893	Description: ufm-prolog.sh failure: hostnames are not found in the fabric after reboot
	Keywords: Hostnames; `ufm-prolog.sh`, reboot
	Discovered in Release: 6.10.0
3495692	Description: UFM Enterprise v6.13.1 server hangs intermittently, blocking UFM REST server, and UFM GUI
	Keywords: UFM REST, UFM GUI
	Discovered in Release: 6.13.1
N/A	Description: Reverted setGuidsForPkey APIs for supporting SHARP reservation (in case it is enabled)
	Keywords: setGuidsForPkey, SHARP Reservation
	Discovered in Release: 6.13.1
Rev 6.13.1	
3459431	Description: UFM System Dump cannot be extracted from UFM 3.0 Enterprise Appliance host when running in high-availability mode.
	Keywords: System Dump, High-Availability
	Discovered in Release: 6.12.0
3461658	Description: The network fast recovery configuration (`/opt/ufm/files/conf/opensm/fast_recovery.conf`) is missing when UFM is deployed in Docker Container mode.
	Keywords: Network Fast Recovery; Docket Container; Missing Configuration
	Discovered in Release: 6.12.0
3461058	Description: When using the Dynamic Telemetry API to create a new telemetry instance, the log rotation mechanism will not be applied for the newly generated logs of the UFM Telemetry instance
	Keywords: Dynamic, Telemetry, Log-rotate
	Discovered in Release: 6.13.0
Rev 6.13.0	

Ref. #	Description
3410826	Description: Rectified inability to modify user password
	Keywords: User Password, Update, Fail
	Discovered in Release: 6.12.1
3383916	Description: Fixed Client CTRL+C server disruption
	Keywords: Client CTRL+C, Server functionality
	Discovered in Release: Rest Over RDMA Image 1.0.0-21
3375414	Description: Fixed improper functionality of UFM UI Dashboard
	Keywords: UI Dashboard
	Discovered in Release: 6.11.0
3342713	Description: Fixed UFM Health configuration for periodic restarts of the telemetry
	Keywords: UFM Health, Telemetry, Periodic restarts
	Discovered in Release: 6.11.1
3361160	Description: Fixed UFM long upgrade time due to a large historical Telemetry database file
	Keywords: Long Upgrade Time, Historical Telemetry, Database File
	Discovered in Release: 6.11.0
3268270	Description: Show managed switches inventory data (Sysinfo) immediately after UFM initialization
	Keywords: Managed Switches, Inventory, Sysinfo
	Discovered in Release: 6.11.0
3338613	Description: Fixed UFM log rotation for supported Ubuntu OSs
	Keywords: Log rotation, Ubuntu
	Discovered in Release: 6.11.0
3338600	Description: Fixed UFM UI lockdown by adding protection to the failed path on backend side
	Keywords: UFM UI, lockdown
	Discovered in Release: 6.11.0

Ref. #	Description
3276163	Description: Fixed remote syslog configuration in UFM Web UI to be persistent
	Keywords: Remote Syslog, Web UI
	Discovered in Release: 6.11.0
3234082	Description: UFM WebUI unresponsive after failover issue
	Keywords: UFM, WebUI, failover
	Discovered in Release: 6.10.0
3199572	Description: Incorrect Tier reporting in the UFM events
	Keywords: Tier, Incorrect Report
	Discovered in Release: 6.10.0
3107006	Description: Using GET All Modules REST API (GET /ufmRest/resources/modules), returns N/A in device_name.
	Keywords: Modules, N/A, device_name
	Discovered in Release: 6.9
3076817	Description: Upgrading to the latest UFM version (UFMAPL_4.8.0.6_UFM_6.9.0.7), the UFM WEB UI shows log and error messages with "invalid date."
	Keywords: WEB UI, "invalid date"
	Discovered in Release: 6.9
3060127	Description: UFM WEB UI - Ports REST API returns tier parameters as N/A in response
	Keywords: WEB UI, tier, N/A
	Discovered in Release: 6.9
3052660	Description: UFM monitoring mode is not working
	Keywords: Monitoring, mode
	Discovered in Release: 6.9
3031121	Description: Network map showing a link between QM8790 and Manta Ray leaf having BW of >20,000 Gb/s
	Keywords: Network Map, BW, 20,000
	Discovered in release: 6.8.0

Ref. #	Description
3003366	Description: UFM Starting and Stopping On Its Own Since Merge
	Keywords: Start, Stop
	Discovered in release: 6.7.0
2968236	Description: Fabric health Old Alerts and events do not clear
	Keywords: Fabric Health, Alerts, clear
	Discovered in release: 6.8.0
2957984	Description: BER Not Being Read or Reported
	Keywords: BER, Not, Reported
	Discovered in release: 6.8.0
3032227	Description: UFM UFMAPL_4.7.0.3_UFM_6.8.0.6 lists one of my skyways as "host" instead of "gateway"
	Keywords: skyway, gateway, host
	Discovered in release: 6.8.0
2966472	Description: UFM Fabric health BER_CHECK warnings
	Keywords: Fabric Health, BER, check
	Discovered in release: 6.8.0
2801258	Description: UFM failed to serve incoming REST API requests
	Keywords: REST API, hang, unresponsive
	Discovered in release: 6.7.0
2782069	Description: UFM APL 4.6 BER not reported (None) in event logs
	Keywords: BER, events, log
	Discovered in release: 6.7.0
2744757	Description: UFM health test: CheckSMConnectivityOnStandby should consider multiple GUIDs on a port
	Keywords: UFM Health, SM connectivity, multiple guids
	Discovered in release: 6.7.0

Ref. #	Description
2830281	Description: UFM (container) is not starting after server reboot
	Keywords: UFM Container, reboot
	Discovered in release: 6.7.0
2804807	Description: UFM WEB GUI becomes Unresponsive and Event/REST API log stops printing
	Keywords: Web UI, unresponsive
	Discovered in release: 6.7.0
2699393	Description: IPMI console login connects to CentOS (UM docker OS) instead of Ubuntu (host OS) after UFM docker installation.
	Keywords: IPMI; CentOS; Login
	Discovered in release: 6.6.1
2638032	Description: Wrong module (line/spine) label appears in effective BER event.
	Keywords: Module; Effective; BER; Event
	Discovered in release: 6.4.1
2618603	Description: UFM failover is not working when bond0 is configured with IPoIB.
	Keywords: Failover, Bond; IPoIB
	Discovered in release: 6.6.1
2615514	Description: UFM software no longer supports license type "UFM APPLIANCE".
	Keywords: License; UFM Appliance
	Discovered in release: 6.5.2
2589617	Description: UFM stopped to discover topology on SuperPOD environment.
	Keywords: Stopped; discover
	Discovered in release: 6.5.2
2335141	Description: Memory leak discovered in ModelMain.py process.
	Keywords: Memory leak
	Discovered in Release: 6.5.1
	Fixed in Release: 6.5.2

Ref. #	Description
2300082	Description: CMP python error
	Keywords: Python, error
	Discovered in Release: 6.5.1
	Fixed in Release: 6.5.2
2373665	Description: UFM license check of UFM permanent license generates invalid license status at the UFM Health Report.
	Keywords: Permanent license; UFM health report
	Discovered in Release: 6.5.1
	Fixed in Release: 6.5.2
2125784	Description: Some commands appear for users with monitor privileges which are not functional. It is recommended not to use this user role.
	Keywords: Monitor, permissions, user
	Discovered in Release: 4.2.0
	Fixed in Release: 6.5.1
-	Description: Performance degradation caused by OpenSM changing the default rate limit of management PKey (0x7fff) to 2.5 GB/s instead of 10GB/s.
	Keywords: OpenSM, Degradation, rate limit
	Discovered in version: 4.2.0
	Fixed in Release: 6.5.1
-	Description: Each HCA is discovered and represented as a separate host. A host with multiple HCAs will be represented as multiple host instances.
	Keywords: Fabric Topology
	Fixed in Release: 6.5.1
1967348	Description: Email sender address cannot contain more than one period (“.”) in the domain name.
	Keywords: Email, sender, period
	Discovered in Release: 6.3
	Fixed in Release: 6.4

Ref. #	Description
2069425	Description: SMTP server username cannot have more than 20 characters.
	Keywords: Email
	Discovered in Release: 6.3
	Fixed in Release: 6.4
1914379	Description: MellanoxCare service can now communicate with UFM (valid only when http communication is configured between MCare and UFM).
	Keywords: MellanoxCare, http, https
	Discovered in Release: 6.2
	Fixed in Release: 6.3
1783048	Description: Opening UFM web UI in monitoring mode is now supported.
	Keywords: Web UI, monitoring mode
	Discovered in Release: 6.2
	Fixed in Release: 6.3
1691882	Description: UFM Agent now is now part of the UFM web UI.
	Keywords: UFM Agent
	Discovered in Release: 6.1
	Fixed in Release: 6.3
1793244	Description: UFM/module temperature thresholds notifications.
	Keywords: Temperature thresholds
	Discovered in Release: 6.1
	Fixed in Release: 6.3
1678669	Description: Fixed an issue where UFM HA prerequisite script was checking for wrong Virtual IP port argument.
	Keywords: UFM HA, prerequisite, Virtual IP, port
	Discovered in Release: 6.1
	Fixed in Release: 6.2

Ref. #	Description
1706226	Description**:** Fixed an issue where MLNX_OS credentials were missing at the device "access_credentials" menu (the issue was detected on old Java based GUI). At the new UFM Web UI – MLNX_OS credentials are represented by HTTP credentials.
	Keywords: MLNX_OS, credentials
	Discovered in Release: 6.1
	Fixed in Release: 6.2
1486595	Description: Fixed an issue where CentOS 7.5 was not recognized as RHEL 7 flavor upon installation.
	Keywords: Installation, CentOS, RHEL
	Discovered in: 6.0
	Fixed in: 6.1
1358248	Description: Fixed the issue where ibdiagnet's unresponsiveness when using the get_physical_info flag caused UFM to hang.
	Keywords: ibdiagnet
	Discovered in: 5.10
	Fixed in: 6.0
1294010	Description: Fixed the issue where partition configuration was lost after upgrading to UFM version 5.9.6 and restarting the server.
	Keywords: partitions.conf, PKey, configuration
	Discovered in: 5.9.6
	Fixed in: 5.10
1276539	Description: Updated report execution command in order to avoid the following false warning of wrong link speed during topology comparison.
	Keywords: Topology compare report
	Discovered in: 5.9.6
	Fixed in: 5.10

Ref. #	Description
1131286	Description: Fixed a memory leak of UFM's main process when running multiple reports periodically.
	Keywords: Memory leak, reports
	Discovered in: 5.9
	Fixed in: 5.9.6
1064349	Description: Fixed an issue where UFM reported false alarm about OpenSM irresponsiveness (sminfo command returned with failure).
	Keywords: OpenSM, sminfo
	Discovered in: 5.8
	Fixed in: 5.9.6
987236	Description: Fixed a web UI security issue by changing the SSL certificate RSA keys' size to 2048 bit (instead of 1024).
	Keywords: Web UI, security, certificate, apache
	Discovered in: 5.8
	Fixed in: 5.9
965302	Description: Fixed UFM HA installation with non-standard file mode creation mask (umask 000).
	Keywords: HA, umask
	Discovered in: 5.8
	Fixed in: 5.9

Known Issues History

Ref #	Issue
3560659	**Description**: Modifying the `mtu_limit` parameter for [MngNetwork] in gv.cfg does not accurately reflect changes upon restarting UFM.
	Keywords: `mtu_limit`, MngNetwork, gv.cfg, UFM restart
	Workaround: UFM needs to be restarted twice in order for the changes to take effect.
	Discovered in Release: v6.15.0

Ref #	Issue
3729822	**Description**: The Logs API temporarily returns an empty response when SM log file contains messages from both previous year (2023) and current year (2024).
	Keywords: Logs API, Empty response, Logs file
	Workaround: N/A (issue will be automatically resolved after the problematic SM log file, which include messages from 2023 and 2024 years, will be rotated)
	Discovered in Release: v6.15.0
3675071	**Description**: UFM stops gracefully after the b2b primary cable is physically disconnected
	Keywords: UFM HA, B2B, Primary Cable Disconnection
	Workaround: N/A
	Discovered in Release: 6.14.1
N/A	**Description:** Execution of UFM Fabric Health Report (via UFM Web UI / REST AP will trigger ibdiagnet to use SLRG register which might cause some of the Switch and HCA's firmware to stuck and cause the HCA's ports to stay at "Init" state.
	Keywords:
	Discovered in Release: 6.14.0
3538640	**Description:** Fixed ALM plugin log rotate function.
	Keywords: ALM, Plugin, Log rotate
	Discovered in Release: 6.13.0
3532191	**Description**: Fixed UFM hanging (database is locked) after corrective restart of UFM health.
	Keywords: Hanging, Database, Locked
	Discovered in Release: 6.13.0
3555583	**Description:** Resolved REST API links' inability to return hostname for computer nodes.
	Keywords: REST API, Links, Hostname, Computer Nodes
	Discovered in Release: 6.12.1
3549795	**Description:** Fixed ufm_ha_cluster status to show DRBD sync status.
	Keywords: ufm_ha_cluster, DRBD, Sync Status
	Discovered in Release: 6.13.0

Ref #	Issue
3549793	**Description:** Fixed UFM HA installation failure.
	Keywords: HA, Installation
	Discovered in Release: 6.13.0
3547517	**Description:** Fixed UFM logs REST API returning empty result when SM logs exi on the disk.
	Keywords: Logs, SM logs, Empty
	Discovered in Release: 6.11.0
3546178	**Description:** Fixed SHARP jobs failure when SHARP reservation feature is enabled.
	Keywords: SHARP, Jobs, Reservation
	Discovered in Release: 6.13.0
3541477	**Description:** Fixed UFM module temperature alerting on wrong thresholds.
	Keywords: Module Temperature, Alert Threshold
	Discovered in Release: 6.13.0
3191419	**Description:** Fixed UFM default session API returning port counter values as NULL.
	Keywords: Null, Port Counter, Value, API
	Discovered in Release: 6.9.0
3560659	**Description**: Fixed proper update in [MngNetwork] mtu_limit in gv.cfg when restarting UFM.
	Keywords: mtu_limit, gv.cfg, Update, UFM restart
	Discovered in Release: 6.13.1
3534374	**Description:** Fixed configure_ha_nodes.sh failure when deploying UFM6.13.x HA on Ubuntu22.04.
	Keywords: configure_ha_nodes.sh, HA, Ubuntu22.04
	Discovered in Release: 6.13.0
3496853	**Description:** Fixed daily report not being sent properly.
	Keywords: Daily Report, Failure
	Discovered in Release: 6.13.0

Ref #	Issue
3469639	**Description**: Fixed REST RDMA server failure every couple of days, causing inability to retrieve ibdiagnet data.
	Keywords: REST RDMA, ibdiagnet
	Discovered in Release: 6.12.0
3455767	**Description:** Fixed incorrect combination of multiple devices in monitoring.
	Keywords: Monitoring, Incorrect combination
	Discovered in Release: 6.12.0
3511410	**Description**: Collect system dump for DGX host does not work due to missing sshpass utility.
	Workaround: Install sshpass utility on the DGX .
	Keywords: System Dump, DGX, sshpass utility
3432385	**Description**: UFM does not support HDR switch configured with hybrid split mode, where some of the ports are split and some are not.
	Workaround: UFM can properly operate when **all** or **none** of the HDR switch ports are configured as split.
	Keywords: HDR Switch, Ports, Hybrid Split Mode
3472330	**Description**: On bare-metal high availability (HA), when initiating a UFM system dump from either the master or standby node, the collection process will not include the HA dumps (pacemaker and DRBD).
	Workaround: To extract the HA system dump from bare-metal, run the following command from the master/standby nodes: `/usr/bin/vsysinfo -S all -e -f /etc/ufm/ufm-ha-sysdump.conf -O /tmp/HA_sysdump` The extracted HA system dump are stored in `/tmp/HA_sysdump.gz.tar`
	Keywords: UFM System Dump, HA, Bare-Metal

Ref #	Issue
3461658	**Description**: After the upgrade from UFM Enterprise v6.13.0 GA to UFM Enterprise v6.13.1 FUR, the network fast recovery path in `opensm.conf` is not automatically updated and remains with a null value (`fast_recovery_conf_file (null))`
	Workaround: If you wish to enable the network fast recovery feature in UFM, make sure to set the appropriate path for the current fast recovery configuration file (`/opt/ufm/files/conf/opensm/fast_recovery.conf`) in the opensm.conf file located at `/opt/ufm/files/conf/opensm`, before starting UFM.
	Keywords: Network fast recovery, Missing, Configuration
N/A	**Description**: Enabling a port for a managed switch fails in case that port is not disabled in a persistent way (this may occur in ports that were disabled on previous versions of UFM - prior to UFM v6.12.0)
	Workaround: Set "persistent_port_operation=false” in `gv.cfg` to use non-persistent (legacy) disabling or enabling of the port. UFM restart is required.
	Keywords: Disable, Enable, Port, Persistent
3346321	**Description**: Failover to another port (multi-port SM) will not work as expected case UFM was deployed as a docker container
	Workaround: Failover to another port (multi-port SM) works properly on UFM Bare-metal deployments
	Keywords: Failover to another port, Multi-port SM
3348587	**Description**: Replacement of defected nodes in the HA cluster does not work when PCS version is 0.9.x
	Workaround: N/A
	Keywords: Defected Node, HA Cluster, pcs version

Ref #	Issue
3336769	**Description**: UFM-HA: In case the back-to-back interface is disabled or disconnected, the HA cluster will enter a split-brain state, and the "ufm_ha_cluster status" command will stop functioning properly.
	Workaround: To resolve the issue: 1. Connect or enable the back-to-back interface 2. Run `pcs cluster start --all` 3. Follow instructions in Split-Brain Recovery in HA Installation.
	Keywords: HA, Back-to-back Interface
3361160	**Description**: Upgrading UFM Enterprise from versions 6.8.0, 6.9.0 and 6.10.0 results in cleanup of UFM historical telemetry database (due to schema change This means that the new telemetry data will be stored based on the new schema.
	Workaround: To preserve the historical telemetry database data while upgradir from UFM version 6.8.0, 6.9.0 and 6.10.0, perform the upgrade in two phases. First, upgrade to UFM v6.11.0, and then upgrade to the latest UFM version (UFM v6.12.0 or newer). It is important to note that the upgrade process may take longer depending on the size of the historical telemetry database.
	Keywords: UFM Historical Telemetry Database, Cleanup, Upgrade
3346321	**Description:** In some cases, when multiport SM is configured in UFM, a failover to the secondary node might be triggered instead of failover to the local available port
	Workaround: N/A
	Keywords: Multiport SM, Failover, Secondary port
3240664	**Description**: This software release does not support upgrading the UFM Enterprise version from the latest GA version (v6.11.0). UFM upgrade is supported in UFM Enterprise v6.9.0 and v6.10.0.
	Workaround: N/A
	Keywords: UFM Upgrade

Ref #	Issue
3242332	**Description**: Upgrading MLNX_OFED uninstalls UFM
	Workaround: Upgrade UFM to a newer version (v6.11.0 or newer), then upgrade MLNX_OFED
	Keywords: MLNX_OFED, Uninstall, UFM
3237353	**Description:** Upgrading from UFM v6.10 removes MLNX_OFED crucial packages
	Workaround: Reinstall MLNX_OFED/UFM
	Keywords: MLNX_OFED, Upgrade, Packages
N/A	Description: Running UFM software with external UFM-SM is no longer supported
	Workaround: N/A
	Keywords: External UFM-SM
3144732	**Description**: By default, a managed Ubuntu 22 host will not be able to send system dump (sysdump) to a remote host as it does not include the sshpass utility.
	Workaround: In order to allow the UFM to generate system dump from a managed Ubuntu 22 host, install the sshpass utility prior to system dump generation.
	Keywords: Ubuntu 22, sysdump, sshpass
3129490	**Description**: HA uninstall procedure might get stuck on Ubuntu 20.04 due to multipath daemon running on the host.
	Workaround: Stop the multipath daemon before running the HA uninstall script on Ubuntu 20.04.
	Keywords: HA uninstall, multipath daemon, Ubuntu 20.04
3147196	**Description**: Running the upgrade procedure on bare metal Ubuntu 18.04 in HA mode might fail.
	Workaround: For instructions on how to apply the upgrade for bare metal Ubuntu 18.04, refer to High Availability Upgrade for Ubuntu 18.04 .
	Keywords: Upgrade, Ubuntu 18.04, Docker Container, failure

Ref #	Issue
3145058	**Description**: Running upgrade procedure on UFM Docker Container in HA mode might fail.
	Workaround: For instructions on how to apply the upgrade for UFM Docker Container in HA, refer to Upgrade Container Procedure.
	Keywords: Upgrade, Docker Container, failure
3061449	**Description**: Upon upgrade of UFM all telemetry configurations will be overridden with the new telemetry configuration of the new UFM version.
	Workaround: If the telemetry configuration is set manually, the user should set up the configuration after upgrading the UFM for the changes to take effect. Telemetry manual configuration should be set on the following telemetry configuration file right after UFM upgrade: `/opt/ufm/conf/telemetry_defaults/launch_ibdiagnet_config.ini`
	Keywords: Telemetry, configuration, upgrade, override.
3053455	**Description:** UFM “Set Node Description” action for unmanaged switches is not supported for Ubuntu18 deployments
	Workaround: N/A
	Keywords: Set Node Description, Ubuntu18
3053455	**Description:** UFM Installations are not supported on RHEL8.X or CentOS8.X
	Workaround: N/A
	Keywords: Install, RHEL8, CentOS8
3052660	**Description:** UFM monitoring mode is not working
	Workaround: In order to make UFM work in monitoring mode, please edit telemetry configuration file: `/opt/ufm/conf/telemetry_defaults/launch_ibdiagnet_config.ini` Search for `arg_12` and set empty value: `arg_12=` Restarting the UFM will run the UFM in monitoring mode. Before starting the UFM make sure to set: `monitoring_mode = yes` in gv.cfg
	Keywords: Monitoring, mode
3054340	**Description:** Setting non-existing log directory will fail UFM to start
	Workaround: Make sure to set a valid (existing) log directory when setting this parameter (gv.cfgàlog_dir)
	Keywords: Log, Dir, fail, start

Ref #	Issue
-	**Description:** Restoring HA standby node and configuring UFM HA with externa UFM-Subnet Managers are not supported on Ubuntu bare-metal deployments
	Workaround: N/A
	Keywords: HA standby node, bare-metal
2887364	**Description:** After upgrading to UFM6.8, in case UFM failed over to the secondary node, trying to get cable information for selected port will fail.
	Workaround: On the secondary UFM node, copy the following files to /usr/bin/ folder: • /usr/flint • /usr/flint_ext • /usr/mlxcables • /usr/mlxcables_ext • /usr/mlxlink • /usr/mlxlink_ext trying to get cable information on the secondary UFM node should work now.
	Keywords: upgrade, failover, cable information
2784560	**Description:** Intentional stop for master container and start it again or reboot c master server will damage the HA failover option
	Workaround: manually restart UFM cluster
	Keywords: UFM Container; Reboot, Failover
2872513	**Description:** after rebooting master container, Failover will be triggered twice (once to the standby and then back again to the master container)
	Workaround: N/A
	Keywords: UFM Container, reboot, failover
2863388	**Description:** Fail to get cables info for NDR Split Port.
	Workaround: N/A
	Keywords: Cable, NDR, Split
N/A	**Description:** In case of using SM mkey per port, several UFM operations might fail (get cable info, get system dump, switch FW upgrade)
	Workaround: N/A
	Keywords: SM, mkey per port

Ref #	Issue
2702950	**Description:** Internet connection is required to download and install SQLite on the old container during software the upgrade process.
	Workaround: N/A
	Keywords: Container; upgrade
2694977	**Description:** Adding a large number of devices (~1000) to a group or a logical server, on large scale setup takes ~2 minutes.
	Workaround: N/A
	Keywords: Add device; group; logical server; large scale
2710613	**Description:** Periodic topology compare will not report removed nodes if the las topology change included only removed nodes.
	Workaround: N/A
	Keywords: Topology comparison
2698055	**Description:** UFM, configured to work with telemetry for collecting historical data, is limited to work only with the configured HCA port. If this port is part of bond interface and a failure occurs on the port, collection of telemetry data via this port stops.
	Workaround: Reconfigure telemetry with the new active port and restart it within UFM.
	Keywords: Telemetry; history; bond; failure
2705974	**Description:** If new ports are added after UFM startup, the default session RES API (GET /ufmRest/monitoring/session/0/data) will not include port statistics fo the newly added ports.
	Workaround: Reset the main UFM. • For UFM standalone – `/etc/init.d/ufmd model_restart` • For UFM HA – `/etc/init.d/ufmha model_restart`
	Keywords: Default session; REST API; missing ports
2714738	**Description:** Intentional stop for master container and start it again or reboot c master server will damage the HA failover option
	Workaround: manually Restart UFM cluster
	Keywords: UFM Container; Reboot, Failover

Ref #	Issue
2872513	Description: after rebooting master container, Failover will be triggered twice (once to the standby and then back again to the master container)
	Workaround: N/A
	Keywords: UFM Container, reboot, failover
2863388	**Description:** Fail to get cables info for NDR Splitted Port.
	Workaround: N/A
	Keywords: Cable, NDR, Split
N/A	**Description:** In case of using SM mkey per port, several UFM operations might fail (get cable info, get system dump, switch FW upgrade)
	Workaround: N/A
	Keywords: SM, mkey per port,
_	**Description:** The UFM which is configured to work with telemetry for collecting historical data, is limited to work only with the configured HCA port - if this port is part of the bond interface and failure occurs, all telemetry data via this port will be stopped.
	Workaround: If a historical telemetry port is apart of the bond and a failure occurs, user should reconfigure the telemetry with a new active port and restart it within UFM.
	Keywords: telemetry, history, bond, failure
	Discovered in release: 6.7
2459320	**Description:** Docker upgrade to UFM6.6.1 from UFM6.6.0 is not supported.
	Workaround: N/A
	Keywords: Docker; upgrade
	Discovered in release: 6.6.1
-	**Description:** SHARP Aggregation Manager over UCX is not supported.
	Workaround: N/A
	Keywords: UCX; SHARP AM
	Discovered in release: 6.6.1

Ref #	Issue
2288038	**Description:** When the user try to collect system dump for UFM Appliance host the job will be completed with an error with the following summary: "Running as a none root user Please switch to root user (super user) and run again."
	Workaround: N/A
	Keywords: System dump, UFM Appliance host
	Discovered in release: 6.5.2
2100564	**Description:** For modular dual-management switch systems, switch information is not presented correctly if the primary management module fails and the secondary takes over.
	Workaround: To avoid corrupted switch information, it is recommended to manually set the virtual IP address (box IP address) for the switch as the managed switch IP address (manual IP address) within UFM.
	Keywords: Modular switch, dual-management, virtual IP, box IP
	Discovered in release: 6.4.1
2135272	**Description:** UFM does not support hosts equipped with multiple HCAs of different types (e.g. a host with ConnectX®-3 and ConnectX-4/5/6) if multi-NIC grouping is enabled (i.e. multinic_host_enabled = true).
	Workaround: All managed hosts must contain HCAs of the same type (either using ConnectX-3 HCAs or use ConnectX-4/5/6 HCAs).
	Keywords: Multiple HCAs
	Discovered in release: 6.4.1
2063266	**Description:** Firmware upgrade for managed hosts with multiple HCAs is not supported. That is, it is not possible to perform FW upgrade for a specific host HCA.
	Workaround: Running software (MLNX_OFED) upgrade on that host will automatically upgrade all the HCAs on this host with the firmware bundled as part of this software package.
	Keywords: FW upgrade, multiple HCAs
	Discovered in release: 6.4.1

Ref #	Issue
-	**Description:** Management PKey configuration (e.g. MTU, SL) can be performed only using PKey management interface (via GUI or REST API).
	Workaround: N/A
	Keywords: PKey, Management PKey, REST API
	Discovered in release: 6.4
2092885	**Description:** UFM Agent is not supported for SLES15 and RHEL8/CentOS8.
	Workaround: N/A
	Keywords: UFM Agent
	Discovered in release: 6.4
-	**Description:** CentOS 8.0 does not support IPv6.
	Workaround: N/A
	Keywords: IPv6
	Discovered in release: 6.4
1895385	**Description:** QoS parameters **(**mtu, sl and rate_limit) change does not take effect unless OpenSM is restarted.
	Workaround: N/A
	Keywords: QoS, PKey, OpenSM
	Discovered in release: 6.3
-	**Description:** Logical Server Auditing feature is supported on RedHat 7.x operating systems only.
	Workaround: N/A
	Keywords: Logical Server, auditing, OS
	Discovered in release: 5.9
-	**Description:** Configuration from lossy to lossless requires device reset.
	Workaround: Reboot all relevant devices after changing behavior from lossy to lossless.
	Keywords: Lossy configuration

Overview

Scale-Out Your Fabric with Unified Fabric Manager

NVIDIA's Unified Fabric Manager (UFM®) is a powerful platform for managing scale-out computing environments. UFM enables data center operators to efficiently monitor and operate the entire fabric, boost application performance and maximize fabric resource utilization.

While other tools are device-oriented and involve manual processes, UFM's automated and application-centric approach bridges the gap between servers, applications and fabric elements, thus enabling administrators to manage and optimize from the smallest to the largest and most performance-demanding clusters.

UFM Benefits

Benefit	Description
Central Console for Fabric Management	UFM provides all fabric management functions in one central console. The ability to monitor, troubleshoot, configure and optimize all fabric aspects is available via one interface. UFM's central dashboard provides a one-view fabric-wide status view.
In-Depth Fabric Visibility and Control	UFM includes an advanced granular monitoring engine that provides real-time access to switch and host data, enabling cluster-wide monitoring of fabric health and performance, real-time identification of fabric-related errors and failures, quick problem resolution via granular threshold-based alerts and a fabric utilization dashboard.
Advanced Traffic Analysis	Fabric congestion is difficult to detect when using traditional management tools, resulting in unnoticed congestion and fabric under-utilization. UFM's unique traffic map quickly identifies traffic trends, traffic bottlenecks, and congestion events spreading over the fabric, which enables the administrator to identify and resolve problems promptly and accurately.

Benefit	Description
Enables Multiple Isolated Application Environments on a Shared Fabric	Consolidating multiple clusters into a single environment with multi-tenant data centers and heterogeneous application landscapes requires specific policies for the different parts of the fabric. UFM enables segmentation of the fabric into isolated partitions, increasing traffic security and application performance.
Service-Oriented Automatic Resource Provisioning	UFM uses a logical fabric model to manage the fabric as a set of business-related entities, such as time critical applications or services. The logical fabric model enables fabric monitoring and performance optimization on the application level rather than just at the individual port or device level. Managing the fabric using the logical fabric model provides improved visibility into fabric performance and potential bottlenecks, improved performance due to application-centric optimizations, quicker troubleshooting and higher fabric utilization.
Quick Resolution of Fabric Problems	UFM provides comprehensive information from switches and hosts, showing errors and traffic issues such as congestion. The information is presented in a concise manner over a unified dashboard and configurable monitoring sessions. The monitored data can be correlated per job and customer, and threshold-based alarms can be set.
Seamless Failover Handling	Failovers are handled seamlessly and are transparent to both the user and the applications running on the fabric, significantly lowering downtime. The seamless failover makes UFM in conjunction with other Mellanox products, a robust, production-ready solution for the most demanding data center environments.
Open Architecture	UFM provides an advanced Web Service interface and CLI that integrate with external management tools. The combination enables data center administrators to consolidate management dashboards while flawlessly sharing information among the various management applications, synchronizing overall resource scheduling, and simplifying provisioning and administration.

Main Functionality Modules

Fabric Dashboard

UFM's central dashboard provides a one-view fabric-wide status view. The dashboard shows fabric utilization status, performance metrics, fabric-wide events, and fabric health

alerts.

The dashboard enables you to efficiently monitor the fabric from a single screen and serves as a starting point for event or metric exploration.

Fabric Segmentation (PKey Management)

In the PKey Management view you can define and configure the segmentation of the fabric by associating ports to specific defined PKeys. You can add, remove, or update the association of ports to the related PKeys and update the qos_parameters for pkey (mtu, rate, service_level).

Fabric Discovery and Physical View

UFM discovers the devices on the fabric and populates the views with the discovered entities. In the physical view of the fabric, you can view the physical fabric topology, model the data center floor, and manage all the physical-oriented events.

Central Device Management

UFM provides the ability to centrally access switches and hosts, and perform maintenance tasks such as firmware and software upgrade, shutdown and restart.

Monitoring

UFM includes an advanced granular monitoring engine that provides real time access to switch and server data. Fabric and device health, traffic information and fabric utilization are collected, aggregated and turned into meaningful information.

Configuration

In-depth fabric configuration can be performed from the Settings view, such as routing algorithm selection and access credentials.

The Event Policy Table, one of the major components of the Configuration view, enables you to define threshold-based alerts on a variety of counters and fabric events. The fabric administrator or recipient of the alerts can quickly identify potential errors and failures, and actively act to solve them.

Fabric Health

The fabric health tab contains valuable functions for fabric bring-up and on-going fabric operations. It includes one-click fabric health status reporting, UFM Server reporting, database and logs' snapshots and more.

Logging

The Logging view enables you to view detailed logs and alarms that are filtered and sorted by category, providing visibility into traffic and device events as well as into UFM server activity history.

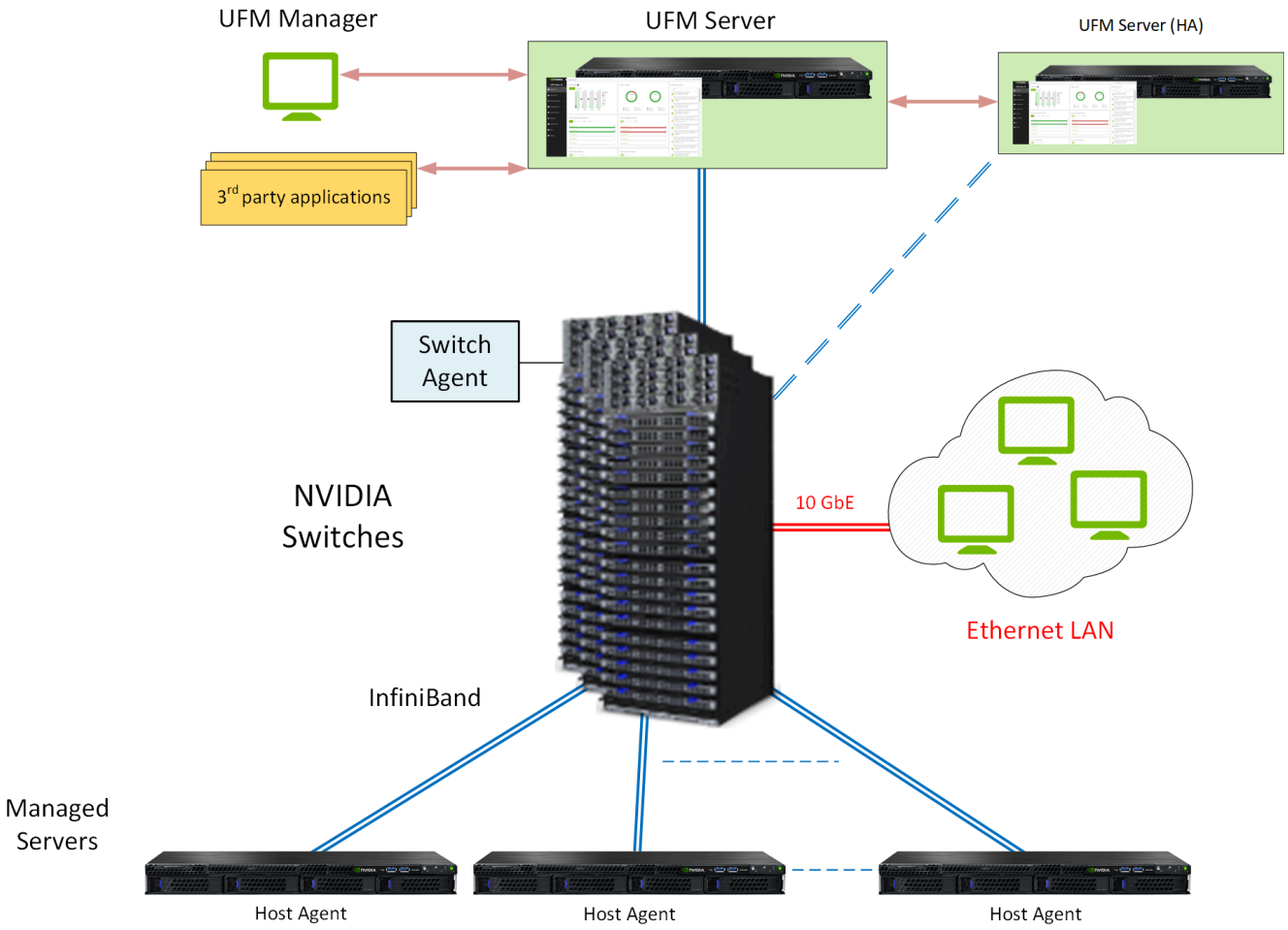
High Availability

In the event of a failover, when the primary (active) UFM server goes down or is disconnected from the fabric, UFM's High Availability (HA) capability allows for a secondary (standby) UFM server to immediately and seamlessly take over fabric management tasks. Failovers are handled seamlessly and are transparent to both the user and the applications running in the fabric. UFM's High Availability capability, when combined with Mellanox's High Availability switching solutions allows for non-disruptive operation of complex and demanding data center environments.

InfiniBand Fabric Managed by UFM

NVIDIA®UFM is a host-based solution that provides all the management functionalities required for managing fabrics.

Fabric Topology with UFM



UFM Server is a server on which UFM is installed and has complete visibility over the fabric to manage routing on all devices.

UFM HA Server is a UFM installed server on a secondary server for High Availability deployment.

Managed Switching Devices are fabric switches, gateways, and routers managed by UFM.

Managed Servers are the compute nodes in the fabric on which the various applications are running, and UFM manages all servers connected to the fabric.

UFM Host Agent is an optional component that can be installed on the Managed Servers. UFM Host Agent provides local host data and host device management functionality.

The UFM Host Agent provides the following functionality:

- Discovery of IP address, CPU, and memory parameters on host
- Collection of CPU/Memory/Disk performance statistics on host

- Upgrading HCA Firmware and OFED remotely
- Creating an IP interface on top of the InfiniBand partition

UFM Switch Agent is an embedded component in NVIDIA switches that allows IP address discovery on the switch and allows UFM to communicate with the switch. For more information, please refer to Device Management Feature Support.

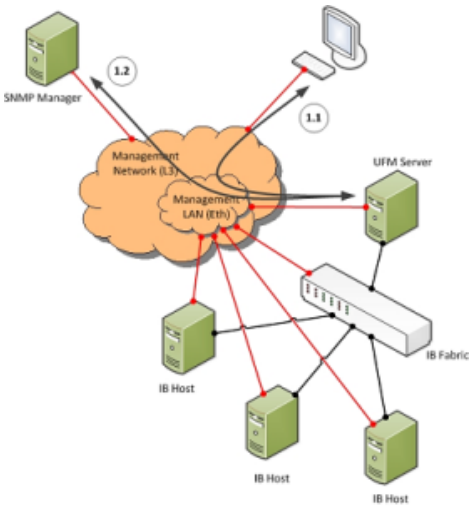
UFM Communication Requirements

This chapter describes how the UFM server communicates with InfiniBand fabric components.

UFM Server Communication with Clients

The UFM Server communicates with clients over IP. The UFM Server can belong to a separate IP network, which can also be behind the firewall.

UFM Server Communication with Clients



UFM Server Communication with UFM Web UI Client

Communication between the UFM Server and the UFM web UI client is HTTP(s) based. The only requirement is that TCP port 80 (443) must not be blocked.

UFM Server Communication with SNMP Trap Managers

The UFM Server can send SNMP traps to configured SNMP Trap Manager(s). By default, the traps are sent to the standard UDP port 162. However, the user can configure the destination port. If the specified port is blocked, UFM Server traps will not reach their destination.

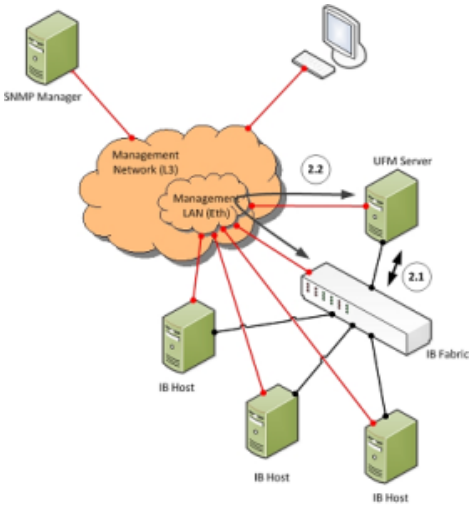
Summary of UFM Server Communication with Clients

Affected Service	Network	Address / Service / Port	Direction
Web UI Client	Out-of-band management*	HTTP / 80 HTTPS / 443	Bi-directional
SNMP Trap Notification	Out-of-band management*	UDP / 162 (configurable)	UFM Server to SNMP Manager

*If the client machine is connected to the IB fabric, IPoIB can also be used.

UFM Server Communication with InfiniBand Switches

UFM Server Communication with InfiniBand Switches



UFM Server InfiniBand Communication with Switch

The UFM Server must be connected directly to the InfiniBand fabric (via an InfiniBand switch). The UFM Server sends the standard InfiniBand Management Datagrams (MAD) to

the switch and receives InfiniBand traps in response.

UFM Server Communication with Switch Management Software (Optional)

The UFM Server auto-negotiates with the switch management software on Mellanox Grid Director switches. The communication is bound to the switch Ethernet management port.

The UFM Server sends a multicast notification to MCast address 224.0.23.172, port 6306 (configurable). The switch management replies to UFM (via port 6306) with a unicast message that contains the switch GUID and IP address. After auto-negotiation, the UFM server uses Switch JSON API (HTTPS based) to retrieve inventory data and to apply switch actions (software upgrade and reboot) on the managed switch.

The following Device Management tasks are dependent on successful communication as described above:

- Switch IP discovery
- FRU Discovery (PSU, FAN, status, temperature)
- Software and firmware upgrades

The UFM Server manages IB Switch Devices over **HTTPS** (default port **443** – configurable) and / or SSH (default port 22 – configurable).

UFM Server Communication with Externally Managed Switches (Optional)

UFM server uses Ibdiagnet tool to discover chassis information (PSU, FAN, status, temperature) of the externally managed switches.

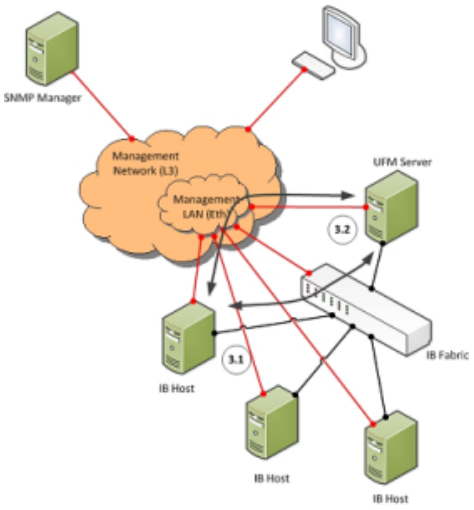
By monitoring chassis information data, UFM can trigger selected events when module failure occurs or a specific sensor value is above threshold.

Summary of UFM Server Communication with InfiniBand Switches

Affected Service	Network	Address / Service / Port	Direction
InfiniBand Management / Monitoring	InfiniBand	Management Datagrams	Bi-directional
Switch IP Address Discovery (auto-negotiation with switch management software)	Out-of-band management	Multicast 224.0.23.172, TCP / 6306 (configurable)	Multicast: UFM Server to switch TCP: Bi-directional
Switch Chassis Management / Monitoring	Out-of-band management	TCP / UDP / 6306 (configurable) SNMP / 161 (configurable) SSH / 22 (configurable)	Bi-directional

UFM Server Communication with InfiniBand Hosts

UFM Server Communication with InfiniBand Hosts



UFM Server InfiniBand Communication with HCAs

The UFM Server must be connected directly to the InfiniBand fabric. The UFM Server sends the standard InfiniBand Management Datagrams (MADs) to the Host Card Adapters (HCAs) and receives InfiniBand traps.

UFM Server Communication with Host Management (Optional)

The UFM Server auto-negotiates with the UFM Agent on a Host. The UFM Host Agent can be bound to the management Ethernet port or to an IPoIB interface (configurable). The UFM Server sends a multicast notification to MCast address 224.0.23.172, port 6306 (configurable). The UFM Agent replies to UFM (port 6306) with a unicast message that contains the host GUID and IP address. After auto-negotiation, the UFM Server and UFM Agent use XML-based messaging.

The following Device Management tasks are dependent on successful communication as described above:

- Host IP discovery
- Host resource discovery and monitoring: CPU, memory, disk
- Software and firmware upgrades

Note

UFM 3.6 supports in-band HCA FW upgrade. This requires enabling FW version and PSID discovery over vendor-specific MADs. for more information, see the UFM User Manual.

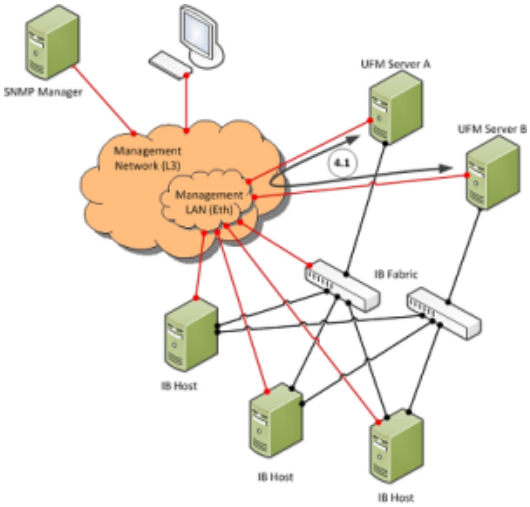
The UFM Server connects to the hosts over SSH (default port 22 - configurable) with root credentials, which are located in the UFM Server database.

Summary of UFM Server Communication with InfiniBand Hosts

Affected Service	Network	Address / Service / Port	Direction
InfiniBand Management / Monitoring	InfiniBand	Management Datagrams	Bi-directional
Host IP Address Discovery (auto-negotiation with UFM Host Agent)	Out-of-band management or IPoIB	Multicast 224.0.23.172, TCP / 6306 (configurable)	Multicast: UFM Server to UFM Agent TCP: Bi-directional
Host OS Management / Monitoring	Out-of-band management or IPoIB	TCP / UDP / 6306 (configurable) SSH / 22 (configurable)	Bi-directional

UFM Server High Availability (HA) Active—Standby Communication

UFM Server HA Active—Standby Communication



UFM Server HA Active—Standby Communication

UFM Active — Standby communication enables two services: heartbeat and DRBD.

- *heartbeat* is used for auto-negotiation and keep-alive messaging between active and standby servers. *heartbeat* uses port 694 (udp).

- DRBD is used for low-level data (disk) synchronization between active and standby servers. DRBD uses port 8888 (tcp).

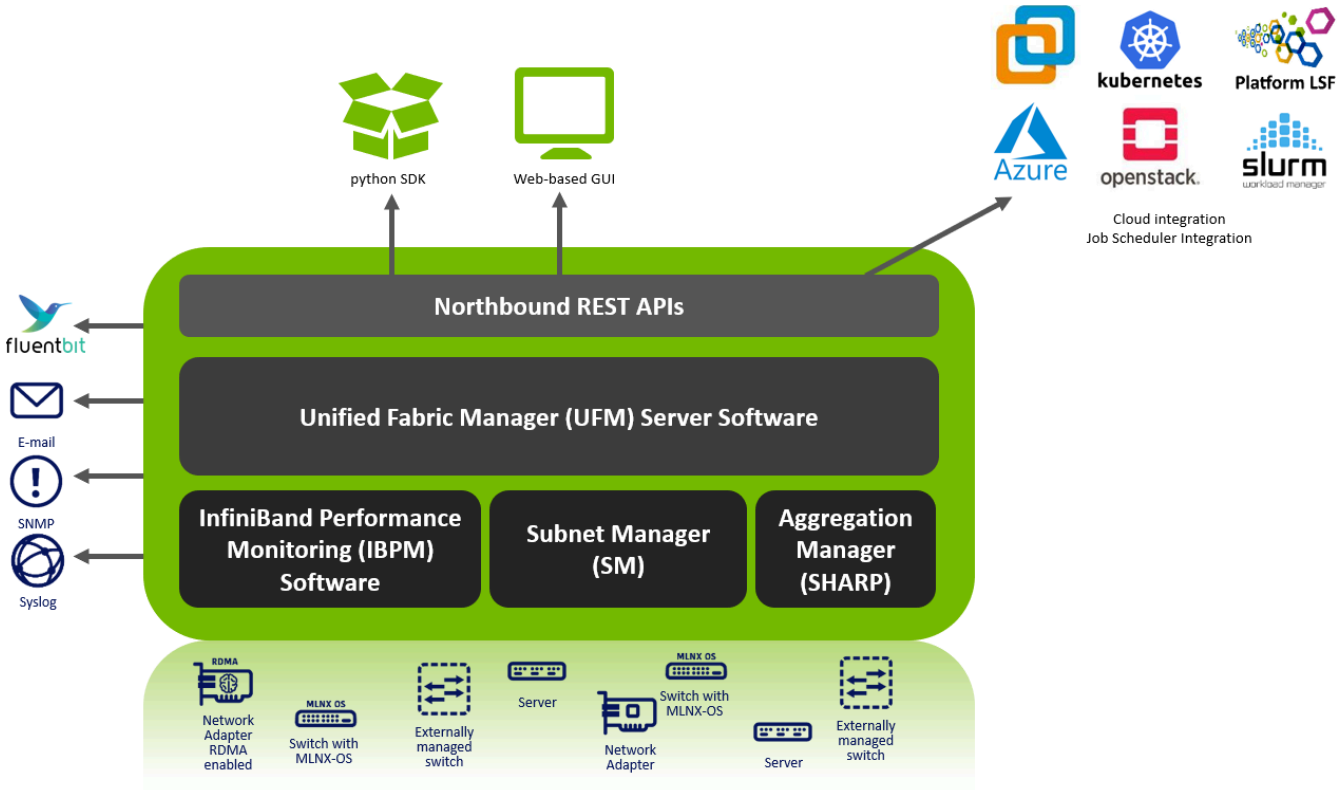
Affected Service	Network	Address / Service / Port	Direction
UFM HA heartbeat	Out-of-band management*	UDP / 694	Bi-directional
UFM HA DRBD	Out-of-band management*	TCP / 8888	Bi-directional

*An IPoIB network can be used for HA, but this is not recommended, since any InfiniBand failure might cause split brain and lack of synchronization between the active and standby servers.

UFM Software Architecture

The following figure shows the UFM high-level software architecture with the main software components and protocols. Only the main logical functional blocks are displayed and do not necessarily correspond to system processes and threads.

UFM High-Level Software Architecture



Graphical User Interface

UFM User Interface is a web application based on JavaScript and Angular JS, which is supported by any Web Browser. The Web application uses a standard REST API provided by the UFM server.

Client Tier API

Third-party clients are managed by the REST API.

Client Tier SDK Tools

Support for UFM's API and a set of tools that enhance UFM functionality and interoperability with third-party applications are provided as part of UFM.

UFM Server

UFM server is a central data repository and management server that manages all physical and logical data. UFM-SDN Appliance receives all data from the Device and Network tiers and invokes Device and Network tier components for management and configuration tasks. UFM-SDN Appliance uses a database for data persistency. The UFM-SDN Appliance is built on the Python twisted framework.

Subnet Manager

Subnet Manager (SM) is the InfiniBand "Routing Engine", a key component used for fabric bring-up and routing management. UFM uses the Open Fabric community OpenSM Subnet Manager. UFM uses a plug-in API for runtime management and fabric data export.

NVIDIA Scalable Hierarchical Aggregation and Reduction Protocol (SHARP)™ Aggregation Manager

NVIDIA Scalable Hierarchical Aggregation and Reduction Protocol (SHARP) is a technology that improves the performance of mathematical and machine learning applications by offloading collective operations from the CPU to the switch network.

Aggregation Manager (AM) is a key component of NVIDIA SHARP software, used for NVIDIA SHARP resources management.

For further information about NVIDIA SHARP AM, refer to Appendix - NVIDIA SHARP Integration .

Performance Manager

The UFM Performance Manager component collects performance data from the managed fabric devices and sends the data to the UFM-SDN Appliance for fabric-wide analysis and display of the data.

Device Manager

The Device Manager implements the set of common device management tasks on various devices with varying management interfaces. The Device Manager uses SSH protocol and operates native device CLI (command-line interface) commands.

UFM Switch Agent

UFM Switch Agent is an integrated part of NVIDIA switch software. The agent supports system parameter discovery and device management functionality on switches.

Communication Protocols

UFM uses the following communication protocols:

- Web UI communicates with the UFM server utilizing **Web Services** carried on **REST API**.
- The UFM server communicates with the switch Agent located on managed switches by proprietary **TCP/UDP**-based discovery and monitoring protocol and **SSH**.
- Monitoring data is sent by the switch Agent to UFM server Listener by a proprietary **TCP**-based protocol.

Overview of Data Model

UFM enables the fabric administrator to manage the fabric based on discovery data collected from the fabric. This data is mapped into model elements (objects) available to the end user via UFM REST API and UFM Web UI.

UFM Model Basics

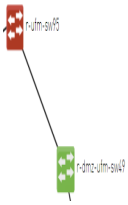
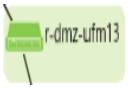
The fabric managed by UFM consists of a set of physical and logical objects, including their connections. The Object Model has a hierarchical object-oriented tree structure with

objects as the tree elements. Each object defines an abstraction for physical or logical fabric elements.

Physical Model

The Physical Model represents the physical resources and connectivity topology of the Network. UFM enables discovery, monitoring and configuration of the managed physical objects.

Physical Objects

Icon	Name	Description
N/A	Port Object	Represents the external physical port on switch or on Host Channel Adapter (HCA). A port is identified by its number. UFM provides InfiniBand standard management and monitoring capabilities on the port level.
N/A	Module Object	Represents the Field Removable Unit, Line card, and Network card on switch or HCA on host. For NVIDIA Switches, Line and Network Cards are modeled as modules.
	Link Object	Represents the physical connection between two active ports.
N/A	Cable Object	Represents the physical cable or the transceiver connected to one of the link edges.
	Computer Object	Represents the computer (host) connected to the Fabric. The UFM Agent installed on the host provides extended monitoring and management capabilities. Hosts without agents are limited to InfiniBand standard management and monitoring capabilities.
	Switch Object	Represents the switch chassis in the Fabric. A Switch object is created for every NVIDIA Switch. Switches of other vendors are represented as InfiniBand Switches and limited by InfiniBand standard management and monitoring capabilities.
	Rack Object	Represents the arbitrary group of switches or computers. When linked devices are shown as a group, the link is shown between the group and the peer object.

UFM Installation and Initial Configuration

UFM® software includes Server and Agent components. UFM Server software should be installed on a central management node. For optimal performance, and to minimize interference with other applications, it is recommended to use a dedicated server for UFM. The UFM Agent is an optional component and should be installed on fabric nodes. The UFM Agent should not be installed on the Management server.

The following sections provide step-by-step instructions for installing and activating the license file, installing the UFM server software, and installing the UFM Agent.

Prerequisites for UFM Server Software Installation

Please refer to Installation Notes for information on system prerequisites.

UFM Installation Steps

To install the UFM software:

- UFM Installation Steps
- Activating Software License
- UFM Configuration
- Running UFM Server Software
- Upgrading UFM Software
- Uninstalling UFM
- UFM Configuration Backup and Restore
- UFM Factory Reset

UFM Installation Steps

- Downloading UFM Software and License File
- Installing UFM Server Software

Downloading UFM Software and License File

Before you obtain a license for the UFM® software, prepare a list of servers with the MAC address of each server on which you plan to install the UFM software. These MAC addresses are requested during the licensing procedure.

Obtaining License

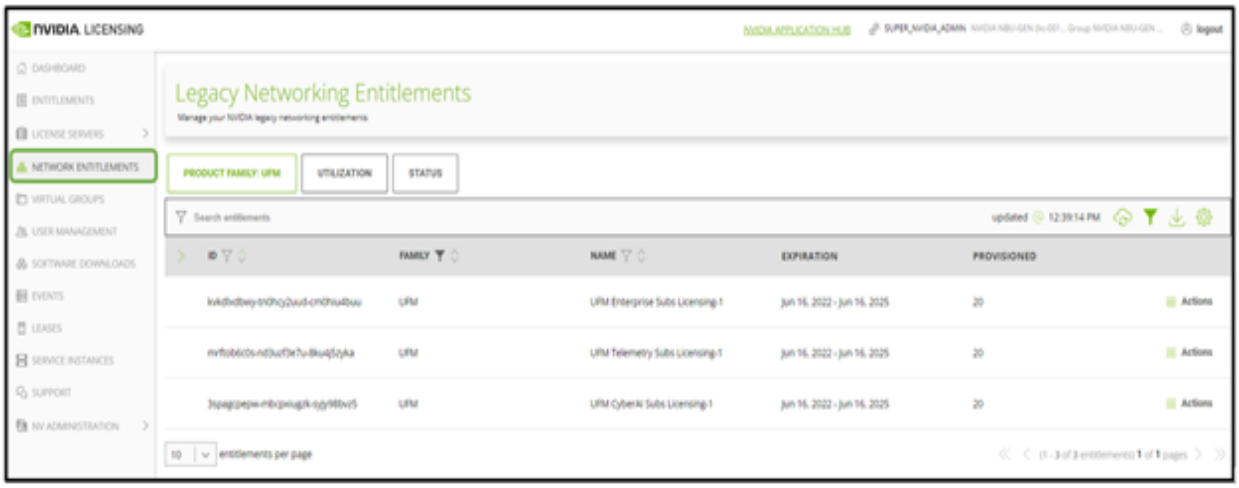
UFM is licensed per managed device according to the UFM license agreement.

When you purchase UFM, you will receive an email with instructions on obtaining your product license. A valid UFM license is a prerequisite for the installation and operation of UFM.

UFM licenses are per managed node and are aggregative. If you install an additional license, the system adds the previous node number and the new node number and manages the sum of the nodes. For example, if you install a license for 10 managed nodes and an additional license for 15 nodes, UFM will be licensed for up to 25 managed nodes.

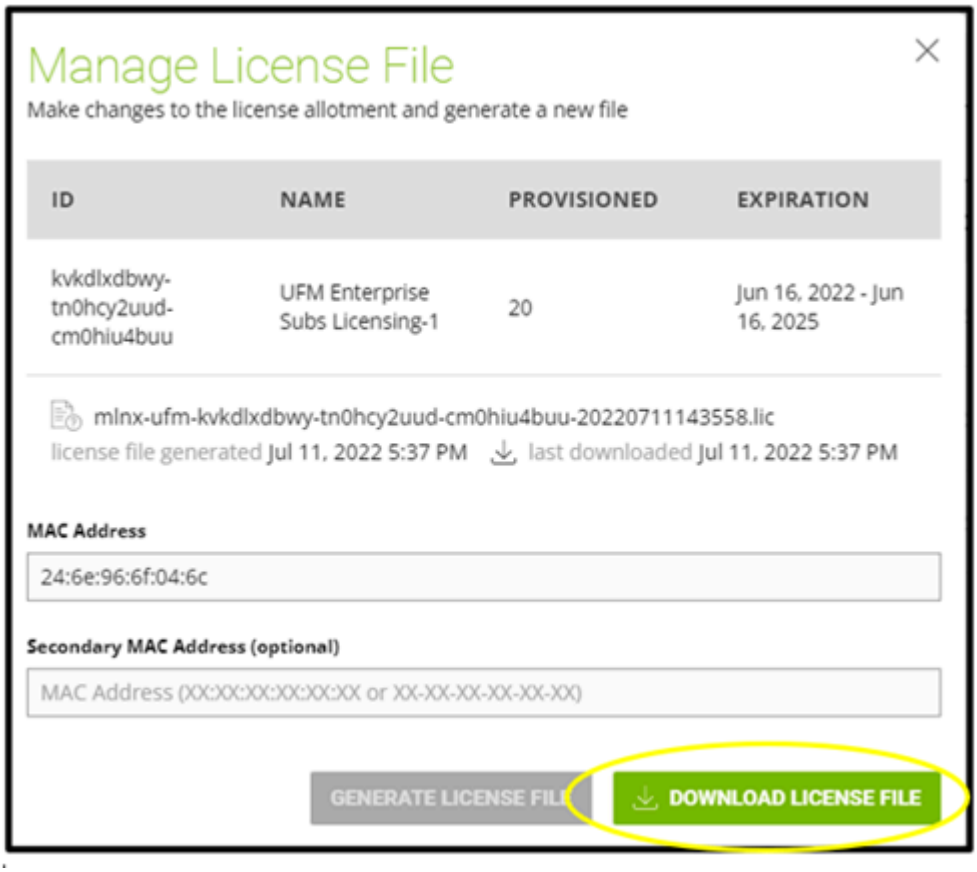
To obtain the license:

1. Go to NVIDIA's Licensing and Download Portal and log in as specified in the licensing email you received.
    - If you did not receive your NVIDIA Licensing and Download Portal login information, contact your product reseller.
2. If you purchased UFM directly from NVIDIA and you did not receive the login information, contact **enterprisesupport@nvidia.com.** Click on the Network Entitlements tab. You'll see a list with the serial licenses of all your software products and software product license information and status.



3. Select the license you want to activate and click on the “Actions” button.

4. In the MAC Address field, enter the MAC address of the delegated license-registered host. If applicable, in the HA MAC Address field, enter your High Availability (HA) server MAC address. If you have more than one NIC installed on a UFM Server, use any of the MAC addresses.



5. Click on Generate License File to create the license key file for the software.

6. Click on Download License File and save it on your local computer.

If you replace your NIC or UFM server, repeat the process of generating the license to set new MAC addresses. You can only regenerate a license two times. To regenerate the license after that, contact NVIDIA Sales Administration at **enterprisesupport@nvidia.com**.

Downloading UFM Software

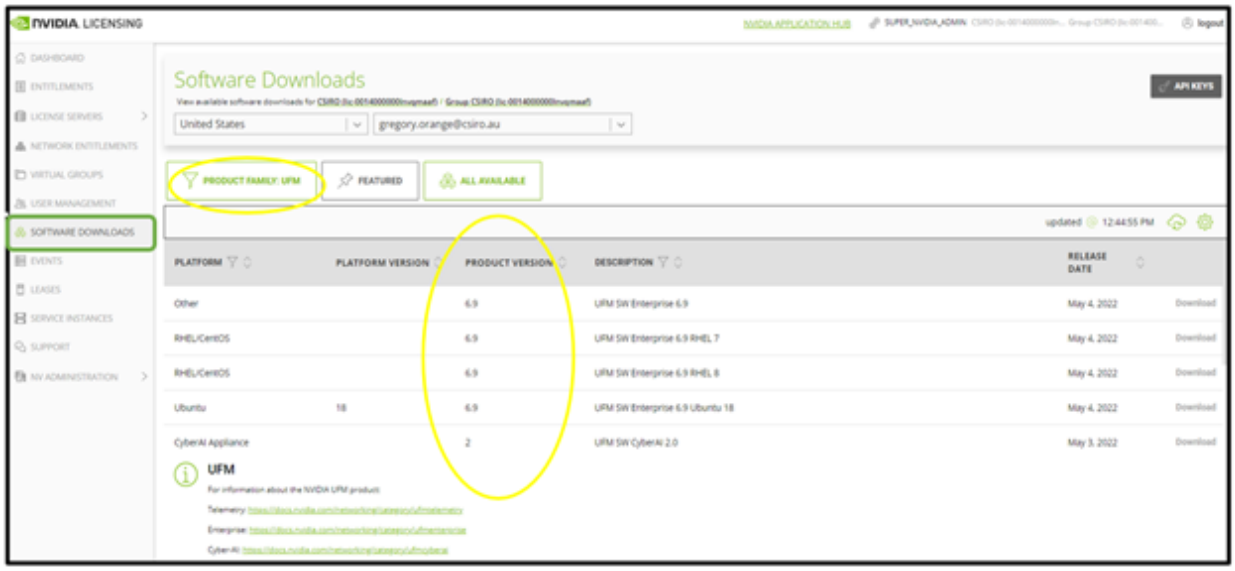
> **Note**
>
> Due to internal packaging incompatibility, this release has two different packages for each of the supported distributions:
>
> - One for UFM deployments over MLNX_OFED 5.X (or newer)
>
> Please make sure to use the UFM installation package compatible to your setup.

This software download process applies to software updates and first-time installation.

If you own the UFM Media Kit and this is your first-time installation, skip this section.

To download the UFM software:

1. Click on Software Downloads, filter the product family to UFM, and select the relevant version of the software. Click on Download.



2. Save the file on your local drive.

3. Click Close.

Installing UFM Server Software

The default UFM® installation directory is */opt/ufm*.

For instructions on installing the UFM server software, please refer to following instructions per desired installation mode.

- Installing UFM Server on Bare Metal Server
    - Installing UFM on Bare Metal Server- Standalone Mode
    - Installing UFM on Bare Metal Server - High Availability Mode
- Installing UFM Docker Container Mode
    - Installing UFM on Docker Container - Standalone Mode
    - Installing UFM on Docker Container - High Availability Mode

The following processes might be interrupted during the installation process:

- httpd (Apache2 in Ubuntu)

- dhcpd

> **Note**
>
> To install UFM over static IPv4 configuration (instead of DHCP) please refer to Configuring UFM Over Static IPv4 Address before installation.

After installation:

1. Activate the software license
2. Perform initial configuration

> **Note**
>
> Before you run UFM, ensure that all ports used by the UFM server for internal and external communication are open and available. For the list of ports, see Appendix – Used Ports.

Prerequisites for UFM Server Software Installation

Verify that a supported version of Linux is installed on your machine. For details, see UFM System Requirements.

The following table lists the packages that must be installed on your machine (according to the system OS) before you install the UFM server software.

RedHat 7	RedHat 8	RedHat 9	Ubuntu 18.04	Ubuntu 20.04	Ubuntu 22.04
acl	acl	acl	acl	acl	acl
apr-util-openssl	apr-util-openssl	apr-util-openssl	apache2	apache2	apache2
bc	bc	bc	bc	bc	bc

RedHat 7	RedHat 8	RedHat 9	Ubuntu 18.04	Ubuntu 20.04	Ubuntu 22.04
cairo	gnutls	gnutls	chrpath	chrpath	chrpath
gnutls	httpd	httpd	cron	cron	cron
httpd	iptables	iptables-nft	gawk	gawk	gawk
iptables	jansson	jansson	lftp	lftp	lftp
lftp	lftp	lftp	libcurl4	libcurl4	libcurl4
libxml2	libnsl	libnsl	logrotate	logrotate	logrotate
libxslt	libxml2	libxml2	python3	python3	python3
mod_session	libxslt	libxslt	qperf	qperf	qperf
mod_ssl	mod_session	mod_session	rsync	rsync	rsync
net-snmp	mod_ssl	mod_ssl	snmpd	snmpd	snmpd
net-snmp-libs	net-snmp	net-snmp	sqlite3	sqlite3	sqlite3
net-snmp-utils	net-snmp-libs	net-snmp-libs	sshpass	sshpass	sshpass
net-tools	net-snmp-utils	net-snmp-utils	ssl-cert	ssl-cert	ssl-cert
php	net-tools	net-tools	sudo	sudo	sudo
psmisc	php	php	telnet	telnet	telnet
python3	psmisc	psmisc	zip	zip	zip
python3-libs	python36	python3			
qperf	qperf	qperf			
rsync	rsync	rsync			
sqlite	sqlite	sqlite			
sshpass	sshpass	sshpass			
sudo	sudo	sudo			
telnet	telnet	telnet			
zip	zip	zip			

Note

On some Ubuntu OSs, Docker is installed via SNAP, which might lead to errors when trying to use UFM Plugins.

To solve this issue, perform the following:

1. Remove Docker installed via SNAP, run:

```
snap remove --purge docker
```

2. Update the local package index, run :

```
apt update
```

3. Install native Docker, run:

```
apt install-y docker.io
```

In addition, ensure the following before you begin installation:

- The computer hostname is not defined as 127.0.0.1 and localhost is defined as 127.0.0.1.
- The hostname must NOT appear on the loopback address line. An example of the loopback address is: 127.0.0.1 localhost.localdomain localhost.
- Disable the firewall service (`/etc/init.d/iptables stop`), or ensure that the required ports are open (see the prerequisite script, refer to Used Ports).
- Disable SELinux.

- If more than one fabric is managed by different UFM instances, set up different management network spaces for each fabric (not the same LAN).
- Uninstall any previously installed Subnet Manager from the UFM server machine.
- MLNX_OFED 5.x version is installed prior to installing UFM.
- As of UFM v.6.12.0, it is **NOT mandatory** to configure the IPoIB fabric interface with an IP address.

  In cases where the IP is configured, it is **mandatory** that the IP is permanently configured and that it starts automatically upon server reboot (the IPoIB fabric interface should be active even if the network is down).

  > **Note**
  >
  > The user can set a persistent IP address using Netplan (mainly for Ubuntu systems) or modifying the interface network script (RedHat systems).

- The default MLNX_OFED installation includes opensm. Remove the MLNX_OFED opensm before UFM installation like the following examples:

  RedHat:

  ```
  rpm -e opensm-3.3.9.MLNX_20111006_e52d5fc-0.1
  ```

  Ubuntu:

  ```
  apt purge opensm
  ```

  By default, ib0 and eth0 are configured as primary access points for the UFM management. If different management and/or InfiniBand interfaces (including bond interfaces) are used as the primary access points, you should modify the

configuration file by running the script */opt/ufm/scripts/change_fabric_config.sh* as described in the section Configuring General Settings in gv.cfg.

Change the UFM Agent interface to the Ethernet and/or IPoIB interfaces used for communication with UFM Agent:

```
ufma_interfaces = ib0,eth0
```

Additional Prerequisites for UFM High Availability Installation

- Reliable and high-capacity out-of-band IP connectivity between the UFM Primary and Secondary servers (1 Gb Ethernet is recommended). This connectivity is used for DRBD synchronization.
- Format two identical servers with dedicated disk partitions for UFM replication. Since the UFM configuration file is replicated to the standby server, both master and standby servers must have the same interfaces.
- Allocate exactly the same size partition on both servers (master and slave) for the replicated data. See UFM Server Requirements for the recommended partition size.

  Partitions should not be mounted and must be zeroed (the file system should not be installed on the partitions). For disk partitioning, see the Linux user manual (man fdisk).
- We recommend establishing a passwordless SSH (via /root/.ssh/authorized_keys file) between the two servers before the installation.
- In fabrics consisting of multiple tiers of switches, it is recommended that the management ports (ib0) of the primary and secondary UFM server be connected to different fabric switches on the same tier (the outermost edge in CLOS 5 designs).

  This is because by default, UFM manages the IB fabric via ib0, port 1 of the HCA. Failure or disconnect of ib0, the IB management port, causes a failure condition in UFM resulting in HA failover.

  When the management ports (ib0) of the primary and secondary UFM server are connected to the same switch, a failure of this switch will result in a disconnect of both UFMs from the fabric, and therefore UFM will not be able to manage the fabric.

Note

Subnet Manager is running over the native InfiniBand layer, therefore bonding the IpoIB interfaces will not provide high availability. For additional information, please refer to section UFM Failover to Another Port.

The UFM installation includes the InfiniBand Performance Management module (IBPM). This module is responsible for reporting performance information back to UFM and upper layer applications. When available, this process is offloaded to the non-management port (default ib1) of the UFM server. Failure or disconnect of the non-management port (ib1) on the primary UFM server will not cause UFM to failover. By default, the UFM Health Monitoring process is configured to try to restart the IBPM. For more information, see UFM Health Configuration in the UFM User Manual.

Installing UFM Server on Bare Metal Server

Installing UFM server on Bare Metal server can be done with the following modes:

- Installing UFM on Bare Metal Server- Standalone Mode
- Installing UFM on Bare Metal Server - High Availability Mode

Installing UFM on Bare Metal Server - High Availability Mode

Before installing UFM server software in High-Availability mode, ensure that the Additional Prerequisites for UFM High Availability Installation are met.

The UFM High-Availability configuration requires dual-link connectivity based on two separate interfaces between the two UFM HA nodes. This configuration comprises of a primary link that is exclusively reserved for DRBD operations and a secondary link

designated for backup purposes. Crucially, it is imperative that communication between the servers is established in a bidirectional manner across both interfaces and validated through user-initiated testing, such as a 'ping' command or other suitable alternatives before HA configuration can be implemented. In cases where only one link is available among the two UFM HA nodes/servers, manually configure UFM with a single link. Refer to Configure HA without SSH Trust (Single Link Configuration).

> **Note**
>
> UFM HA package requires a dedicated partition with the same name for DRBD on both servers. This guide uses `/dev/sda5` as an example.

1. On both servers, Install UFM Enterprise in Stand Alone (SA) mode.

   > **Note**
   >
   > Do not start UFM service.

2. Install the latest pcs and drbd-utils drivers on both servers.

   For Ubuntu:

   ```
   apt install pcs pacemaker drbd-utils
   ```

   For CentOS/Red Hat:

   ```
   yum install pcs pacemaker drbd84-utils kmod-drbd84
   ```

   OR

```
yum install pcs pacemaker drbd90-utils kmod-drbd90
```

3. Download UFM-HA latest package from using this command:

```
wget https://www.mellanox.com/downloads/UFM/ufm_ha_5.3.1-2.tgz
```

For Sha256:

```
wget https://download.nvidia.com/ufm/ufm_ha/5.6.0/ufm_ha_5.3.1-2.sha256
```

> **Note**
>
> For more information on the UFM-HA package and all installation and configuration options, please refer to UFM High-Availability User Guide.

4. Extract the downloaded UFM-HA package on both servers under `/tmp/.`

5. Go to the directory you extracted /tmp/ufm_ha_XXX and run the installation script. For example, if your DRBD partition is `/dev/sda5` run:

```
./install.sh -l /opt/ufm/files/ -d /dev/sda5 -p enterprise
```

6. Configure the HA cluster. There are the three methods:

- Configure HA with SSH Trust (Dual Link Configuration) - Requires passwordless SSH connection between the servers.

- Configure HA without SSH Trust (Dual Link Configuration) - Does not require passwordless SSH connection between the servers, but asks you to run configuration commands on both servers.
- Configure HA without SSH Trust (Single Link Configuration) - Can be used in cases where only one link is available among the two UFM HA nodes/servers.

Configure HA with SSH Trust (Dual Link Configuration)

1.

    1. On the **master server only**, configure the HA nodes. To do so, from /tmp, run the `configure_ha_nodes.sh` command as shown in the below example

```
configure_ha_nodes.sh \
--cluster-password 12345678 \
--master-primary-ip 10.10.10.1 \
--standby-primary-ip 10.10.10.2 \
--master-secondary-ip 192.168.10.1 \
--standby-secondary -ip 192.168.10.2 \
--no-vip
```

> **Note**
>
> The script `configure_ha_nodes.sh` is is located under /usr/local/bin/, therefore, by default, you do not need to use the full path to run it.

> **Note**
>
> The `--cluster-password` must be at least 8 characters long.

Note

To set up a Virtual IP for UFM and gain access to UFM through this IP, regardless of which server is running UFM, you may employ the `--no-vip` OR `--virtual-ip` command and provide an IP address as an argument. This can be achieved by navigating to https://<Virtual-IP>/ufm on your web browser.

Note

When using back-to-back ports with local IP addresses for HA sync interfaces, ensure that you add your IP addresses and hostnames to the `/etc/hosts` file. This is needed to allow the HA configuration to resolve hostnames correctly based on the IP addresses you are using.

Note

`configure_ha_nodes.sh` requires SSH connection to the standby server. If SSH trust is not configured, then you are prompted to enter the SSH password of the standby server during configuration runtime

2. Depending on the size of your partition, wait for the configuration process to complete and DRBD sync to finish.

Configure HA without SSH Trust (Dual Link Configuration)

If you cannot establish an SSH trust between your HA servers, you can use **ufm_ha_cluster directly to configure HA. To configure HA, follow the below instructions:**

Note

Please change the variables in the commands below based on your setup.

1.

1. [On Standby Server] Run the following command to configure Standby Server:

```
ufm_ha_cluster config -r standby \
--local-primary-ip 10.10.50.1 \
--peer-primary-ip 10.10.50.2 \
--local-secondary-ip 192.168.10.1 \
--peer-secondary-ip 192.168.10.2 \
--hacluster-pwd 123456789 \
--no-vip
```

2. [On Master Server] Run the following command to configure Master Server:

```
ufm_ha_cluster config -r master --local-primary-ip
10.10.50.1 \
--peer-primary-ip 10.10.50.2 \
--local-secondary-ip 192.168.10.1 \
--peer-secondary-ip 192.168.10.2 \
--hacluster-pwd 123456789 \
--no-vip
```

You must wait until after configuration for DRBD sync to finish, depending on the size of your partition. To check the DRBD sync status, run:

```
ufm_ha_cluster status
```

Configure HA without SSH Trust (Single Link Configuration)

> **Warning**
>
> This is not the recommended configuration and, in case of network failure, it might cause HA cluster split brain.

If you cannot establish an SSH trust between your HA servers, you can use **ufm_ha_cluster directly to configure HA. To configure HA, follow the below instructions:**

> **Note**
>
> Please change the variables in the commands below based on your setup.

1.
    1. [On Standby Server] Run the following command to configure Standby Server:

```
ufm_ha_cluster config \
-r standby \
-e 10.212.145.5 \
-l 10.212.145.6 \
--enable-single-link
```

    2. [On Master Server] Run the following command to configure Master Server:

```
ufm_ha_cluster config -r master \
-e 10.212.145.6 \
-l 10.212.145.5 \
-i 10.212.145.50 \
--enable-single-link
```

You must wait until after configuration for DRBD sync to finish, depending on the size of your partition. To check the DRBD sync status, run:

```
ufm_ha_cluster status
```

Starting HA Cluster

- To start UFM HA cluster:

```
ufm_ha_cluster start
```

- To check UFM HA cluster status:

```
ufm_ha_cluster status
```

Stopping UFM HA cluster:

```
ufm_ha_cluster stop
```

Note

For complete details on high availability, refer to NVIDIA UFM High-Availability User Guide.

Installing UFM on Bare Metal Server-Standalone Mode

To install the UFM server software as a standalone for InfiniBand:

1. Create a temporary directory (for example */tmp/ufm*).
2. Open the UFM software zip file that you downloaded. The zip file contains the following installation files:
    - RedHat 7/CentOS 7/OEL 7: ufm-X.X-XXX.el7.x86_64.tgz
    - RedHat 8/Centos 8: ufm-X.X-XXX.el8.x86_64.tgz
    - Ubuntu 18.04: ufm-X.X-XXX.Ubuntu18.x86_64.tgz
    - Ubuntu 20.04: ufm-X.X-XXX.Ubuntu20.x86_64.tgz
    - Ubuntu 22.04: ufm-X.X-XXX.Ubuntu22.x86_64.tgz
3. Extract the installation file for your system's OS to the temporary directory that you created.
4. From within the temporary directory, run the following command as root:

```
./install.sh
```

Note

Running with the option "-o ib" is no longer required. For automatic installation, use the -q flag.

For “quiet” installation -q flag can be added (automatically answer yes for each question the installer asks).

Note

Export MULTISUBNET_CONSUMER=1 environment variable before running the installation script to install the UFM server in Multisubnet Consumer mode.

The UFM software is installed. You can now remove the temporary directory.

Installing UFM Docker Container Mode

General Prerequisites

- MLNX_OFED must be installed on the server that will run UFM Docker
- For UFM to work, you must have an InfiniBand port configured with an IP address and in "up" state.

Note

For InfiniBand support, please refer to NVIDIA Inbox Drivers , or MLNX_OFED guides.

- Make sure to stop the following services before running UFM Docker container, as it utilizes the same default ports that they do: Pacemaker, httpd, OpenSM, and Carbon.
- If firewall is running on the host, please make sure to add an allow rule for UFM used ports (listed below):

> **Note**
>
> If the default ports used by UFM are changed in UFM configuration files, make sure to open the modified ports on the host firewall.

  - 80 (TCP) and 443 (TPC) are used by WS clients (Apache Web Server)
  - 8000 (UDP) is used by the UFM server to listen for REST API requests (redirected by Apache web server)
  - 6306 (UDP) is used for multicast request communication with the latest UFM Agents
  - 8005 (UDP) is used as a UFM monitoring listening port
  - 8888 (TCP) is used by DRBD to communicate between the UFM Primary and Standby servers
  - 2022 (TCP) is used for SSH

Prerequisites for Upgrading UFM Docker Container

- Supported versions for upgrade are UFM v.6.10.0 and above.
- UFM files directory from previous container version mounted on the host.

Step 1: Loading UFM Docker Image

To load the UFM docker image, pull the latest image from docker hub:

```
docker pull mellanox/ufm-enterprise:latest
```

Note

You can see full usage screen for ufm-installation by running the container with `-h` or `-help` flag:

```
docker run --rm mellanox/ufm-enterprise-
installer:latest -h
```

If an Internet connection is not available, perform the following:

- Copy the UFM image to your machine.
- Load the image from the file using this command:

```
docker image load -i <image-path>
```

Step 2: Installing UFM Docker

Installation Command Usage

```
docker run -it --name=ufm_installer --rm \
-v /var/run/docker.sock:/var/run/docker.sock \
-v /etc/systemd/system/:/etc/systemd_files/ \
-v /opt/ufm/files/:/installation/ufm_files/ \
-v [LICENSE_DIRECTORY]:/installation/ufm_licenses/ \
mellanox/ufm-enterprise:latest \
--install [OPTIONS]
```

Modify the variables in the installation command as follows:

- `[UFM_LICENSES_DIR]`: UFM license file or files location.

> **Note**
>
> Example: If your license file or files are located under `/downloads/ufm_license_files/` then you must set this volume to be `-v /downloads/ufm_license_files/:/installation/ufm_licenses/`

- `[OPTIONS]`: UFM installation options. For more details see the table below.

Command Options

Flag	Description	Default Value
`-f | --fabric-interface`	IB fabric interface name.	ib0
`-g | --mgmt-interface`	Management interface name.	eth0
`-h | --help`	Show help	N/A

Flag	Description	Default Value
`-m | --multisubnet-consumer`	UFM Multisubnet Consumer mode	N/A

Installation Modes

UFM Enterprise installer supports several deployment modes:

- Installing UFM on Docker Container - High Availability Mode
- Installing UFM on Docker Container - Standalone Mode

Installing UFM on Docker Container - High Availability Mode

Pre-Deployments Requirements

- Install **pacemaker**, **pcs**, and **drbd-utils** on both servers

  For Ubuntu:

  ```
  apt install pcs pacemaker drbd-utils
  ```

  For CentOS/Red Hat:

  ```
  yum install pcs pacemaker drbd84-utils kmod-drbd84
  ```

  OR

```
yum install pcs pacemaker drbd90-utils kmod-drbd90
```

- A partition for DRBD on each server (**with the same name** on both servers) such as `/dev/sdd1`. Recommended partition size is 10-20 GB, otherwise DRBD sync will take a long time to complete.
- CLI command `hostname -i` must return the IP address of the management interface used for pacemaker sync correctly (update `/etc/hosts/` file with machine IP)
- Create the directory on each server under `/opt/ufm/files/` with read/write permissions on each server. This directory will be used by UFM to mount UFM files, and it will be synced by DRBD.
- Disable the firewall service (`/etc/init.d/iptables` stop), or ensure that the required ports are open (see the prerequisite script).
- Disable SELinux.

Installing UFM Containers

On the main server, install UFM Enterprise container with the command below:

```
docker run -it --name=ufm_installer --rm \
-v /var/run/docker.sock:/var/run/docker.sock \
-v /etc/systemd/system/:/etc/systemd_files/ \
-v /opt/ufm/files/:/installation/ufm_files/ \
-v /tmp/license_file/:/installation/ufm_licenses/ \
mellanox/ufm-enterprise:latest \
--install
```

On the standby (secondary) server, install the UFM Enterprise container like the following example with the command below:

```
docker run -it --name=ufm_installer --rm \
-v /var/run/docker.sock:/var/run/docker.sock \
-v /etc/systemd/system/:/etc/systemd_files/ \
-v /opt/ufm/files/:/installation/ufm_files/ \
mellanox/ufm-enterprise:latest \
--install
```

Downloading UFM HA Package

Download the UFM-HA package on both servers using the following command:

```
wget https://www.mellanox.com/downloads/UFM/ufm_ha_5.3.1-2.tgz
```

For Sha256:

```
wget https://download.nvidia.com/ufm/ufm_ha/5.6.0/ufm_ha_5.3.1-2.sha256
```

Installing UFM HA Package

For more information on the UFM-HA package and all installation and configuration options, please refer to UFM High Availability User Guide.

1. [On Both Servers] Extract the downloaded UFM-HA package under `/tmp/`
2. [On Both Servers] Go to the extracted directory `/tmp/ufm_ha_XXX` and run the installation script. For example, if your DRBD partition is `/dev/sda5` run the following command:

   ```
   ./install.sh -l /opt/ufm/files/ -d /dev/sda5 -p enterprise
   ```

Configuring UFM HA

There are the three methods to configure the HA cluster:

- Configure HA with SSH Trust (Dual Link Configuration) - Requires passwordless SSH connection between the servers.
- Configure HA without SSH Trust (Dual Link Configuration) - Does not require passwordless SSH connection between the servers, but asks you to run configuration commands on both servers.
- Configure HA without SSH Trust (Single Link Configuration) - Can be used in cases where only one link is available among the two UFM HA nodes/servers.

Configure HA with SSH Trust (Dual Link Configuration)

1. On the **master server only**, configure the HA nodes. To do so, from /tmp, run the `configure_ha_nodes.sh` command as shown in the below example

```
configure_ha_nodes.sh \
--cluster-password 12345678 \
--master-primary-ip 10.10.50.1 \
--standby-primary-ip 10.10.50.2 \
--master-secondary-ip 192.168.10.1 \
--standby-secondary-ip 192.168.10.2 \
--no-vip
```

> **Note**
>
> The script `configure_ha_nodes.sh` is is located under /usr/local/bin/, therefore, by default, you do not need to use the full path to run it.

Note

The `--cluster-password` must be at least 8 characters long.

Note

When using back-to-back ports with local IP addresses for HA sync interfaces, ensure that you add your IP addresses and hostnames to the `/etc/hosts` file. This is needed to allow the HA configuration to resolve hostnames correctly based on the IP addresses you are using.

Note

`configure_ha_nodes.sh` requires SSH connection to the standby server. If SSH trust is not configured, then you are prompted to enter the SSH password of the standby server during configuration runtime

2. Depending on the size of your partition, wait for the configuration process to complete and DRBD sync to finish. To check the DRBD sync status, run:

```
ufm_ha_cluster status
```

Configure HA without SSH Trust (Dual Link Configuration)

If you cannot establish an SSH trust between your HA servers, you can use `ufm_ha_cluster` directly to configure HA. You can see all the options for configuring HA in the Help menu:

```
ufm_ha_cluster config -h
```

To configure HA, follow the below instructions:

> **Note**
>
> Please change the variables in the commands below based on your setup.

1. [On **Standby Server**] Run the following command to configure **Standby Server**:

   ```
   ufm_ha_cluster config -r standby -e <peer ip address> -l
   <local ip address> -p <cluster_password>
   ```

2. [On **Master Server**] Run the following command to configure **Master Server**:

   ```
   ufm_ha_cluster config -r master -e <peer ip address> -l
   <local ip address> -p <cluster_password> -i <virtual ip
   address>
   ```

Configure HA without SSH Trust (Single Link Configuration)

> **Warning**
>
> This is not the recommended configuration and, in case of network failure, it might cause HA cluster split brain.

If you cannot establish an SSH trust between your HA servers, you can use **ufm_ha_cluster directly to configure HA. To configure HA, follow the below instructions:**

> **Note**
>
> Please change the variables in the commands below based on your setup.

1.

    1. [On **Standby Server**] Run the following command to configure **Standby Server**:

        ```
        ufm_ha_cluster config \
        -r standby \
        -e 10.212.145.5 \
        -l 10.212.145.6 \
        --enable-single-link
        ```

    2. [On **Master Server**] Run the following command to configure **Master Server**:

```
ufm_ha_cluster config -r master \
-e 10.212.145.6 \
-l 10.212.145.5 \
-i 10.212.145.50 \
--enable-single-link
```

You must wait until after configuration for DRBD sync to finish, depending on the size of your partition. To check the DRBD sync status, run:

```
ufm_ha_cluster status
```

IPv6 Example:

```
ufm_ha_cluster config -r standby -l
fcfc:fcfc:209:224:20c:29ff:fee7:d5f2 -e
fcfc:fcfc:209:224:20c:29ff:fecb:4962 --enable-single-link -
p some_secret
```

Starting HA Cluster

- To start UFM HA cluster:

```
ufm_ha_cluster start
```

- To check UFM HA cluster status:

```
ufm_ha_cluster status
```

- To stop UFM HA cluster:

```
ufm_ha_cluster stop
```

- To uninstall UFM HA, first stop the cluster and then run the uninstallation command as follows:

```
/opt/ufm/ufm_ha/uninstall_ha.sh
```

Installing UFM on Docker Container - Standalone Mode

1. Copy only your UFM license file(s) to a temporary directory which we're going to use in the installation command. For example: `/tmp/license_file/`

2. Run the UFM installation command according to the following example which will also configure UFM fabric interface to be ib1:

```
docker run -it --name=ufm_installer --rm \
-v /var/run/docker.sock:/var/run/docker.sock \
-v /etc/systemd/system/:/etc/systemd_files/ \
-v /opt/ufm/files/:/installation/ufm_files/ \
-v /tmp/license_file/:/installation/ufm_licenses/ \
mellanox/ufm-enterprise:latest \
--install \
--fabric-interface ib1
```

3. Reload systemd:

```
systemctl daemon-reload
```

4. To Start UFM Enterprise service run:

```
systemctl start ufm-enterprise
```

Replacing the Standby Node

- Install the HA package for the new node (standby).
- Disconnect the standby node (the old standby) and run the following command on the master node:

```
ufm_ha_cluster detach
```

- Configure the new standby node; please refer to the relevant section depending on the installation
- Connect the new standby to the cluster by running the command on the master node:

```
ufm_ha_cluster attach -l <local primary ip address> -e <peer
primary ip address> -E <peer secondary ip address> -p <clust
```

Activating Software License

1. Before starting the UFM software, copy your license file(s) downloaded from NVIDIA Licensing and Download Portal (*volt-ufm-<serial-number>.lic*) to the master server

under the */opt/ufm/files/licenses* directory. We recommend that you back up the license file(s).

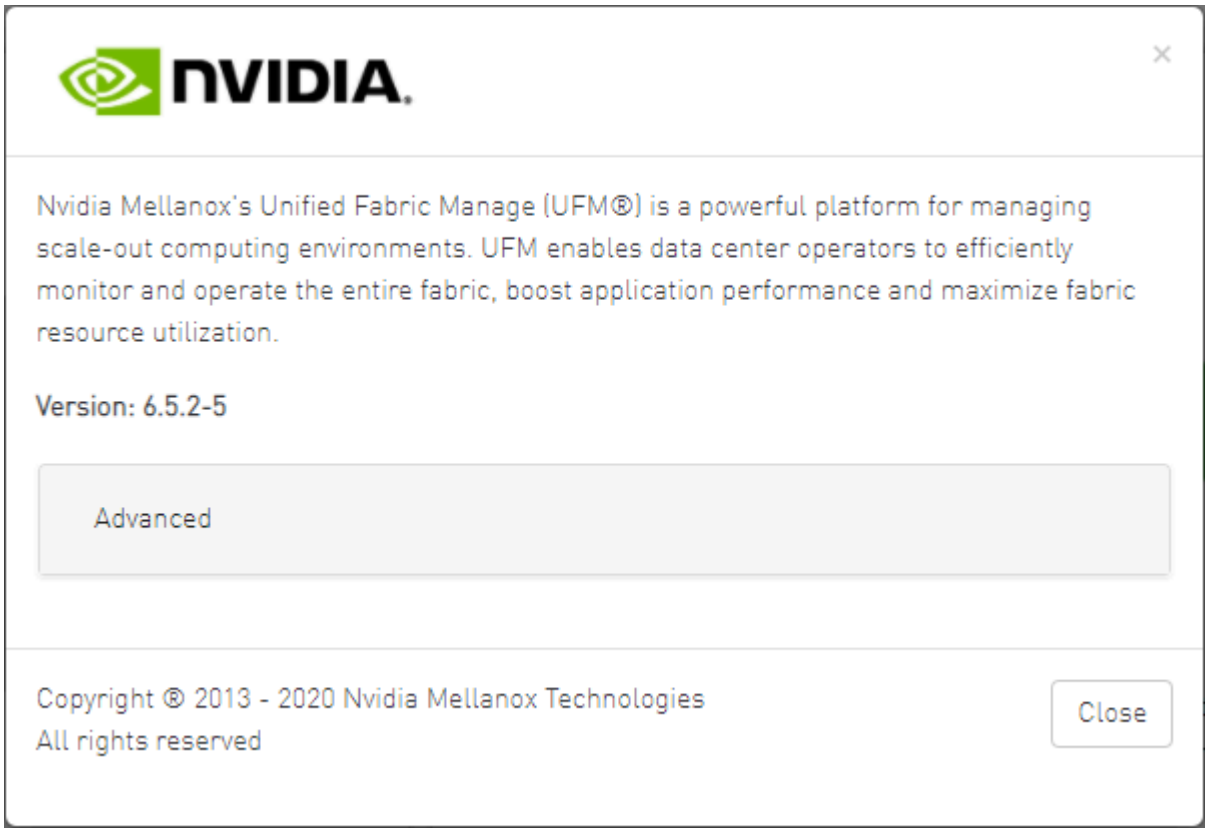
In High Availability mode, the license files are replicated to the standby machine automatically. Your software is now activated.

2. Run the UFM software as described in the following sections.

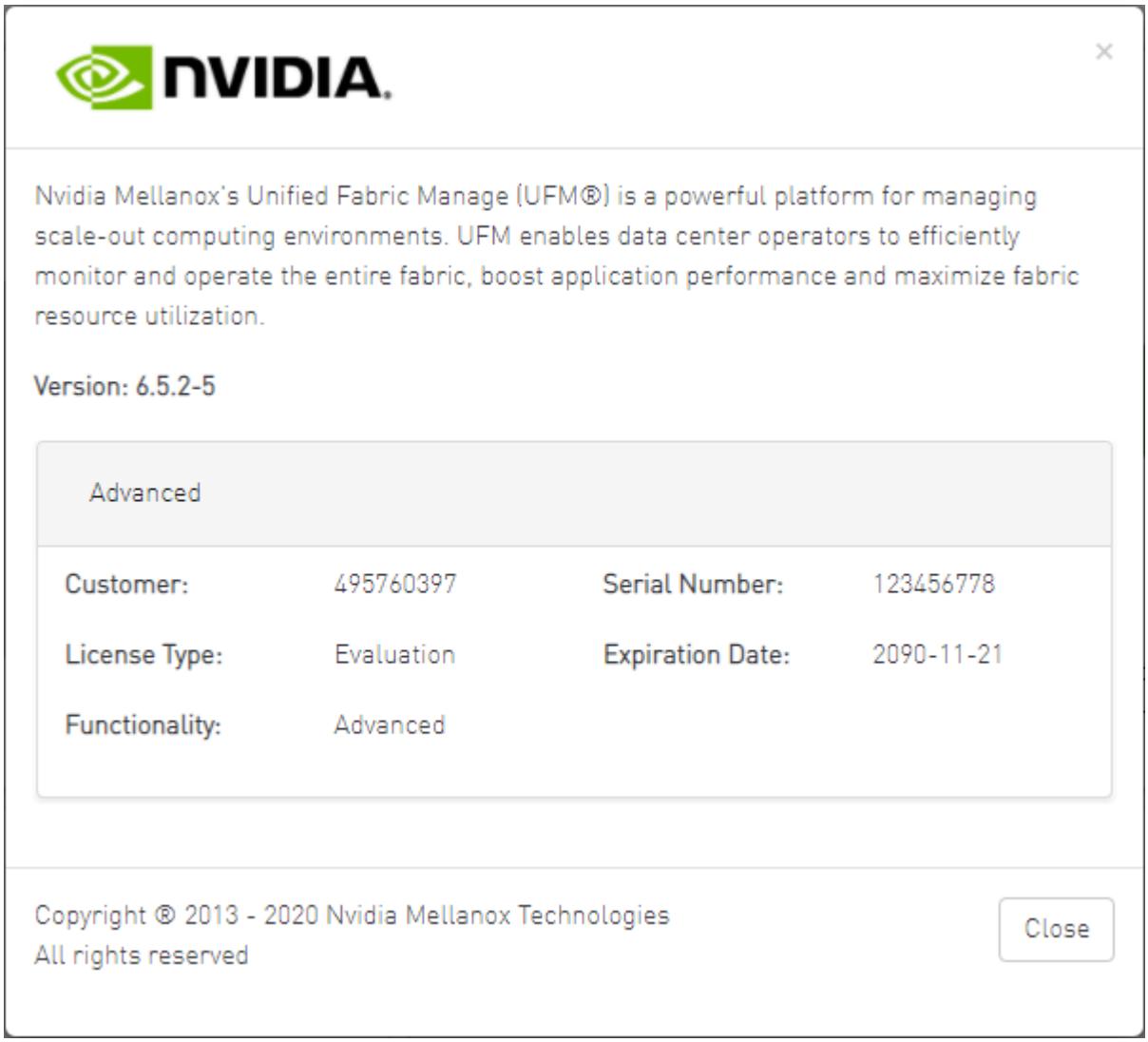
> **Note**
>
> When a UFM license is not provided for activation upon the first UFM installation, the UFM runs on an auto-generated evaluation license which expires after 30 days from the first start-up of the UFM.

Licensing

1. After installing and activating your software, you can view your licenses in the Web UI by clicking the About icon ( ) in the main window.



2. To view the advanced license information, click the Advanced button. The advanced license details will be displayed below.



3. Product Functionality is updated only after startup. If you replace the UFM license, UFM continues to work in the previous mode until the UFM server is restarted.

To view license information from the CLI:

➢

Run CLI Command "**ufmlicense**" to display information about all installed licenses on the UFM server under /opt/ufm/files/licenses. This includes invalid and expired license information.

There are two UFM HA licenses where each license includes 2 different MACs: one for the primary machine and one for the standby machine.

In a given time, for each license, only one MACs is detected to be "Valid" (exists on the local machine) where the other MAC is detected as "Invalid" (exist on the standby machine).

See below ou*tput example when running the CLI command* `ufmlicense` in SA and HA Modes.

HA Mode Output Example:

```
[root@ip-10-224-16-49-dg11 ~]# docker exec -ti ufm bash
root@ ip-10-224-16-49-dg11~# ufmlicense|-----------------------------------------------------------------------------------------------------
--------------------------------||NVIDIA Corp|xxxxxxx-xxxxxxx|UFM Enterprise|Subscription
|e4:43:4b:18:3c:e0|2025-08-29 |128 |3 Years |Invalid ||----------------------------------------------------------------------------------
----------------------------------------------------||NVIDIA Corp|xxxxxxx-xxxxxxx|UFM Enterprise|Subscription
|e4:43:4b:18:49:a0|2025-08-29 |128 |3 Years |Valid ||------------------------------------------------------------------------------------
--------------------------------------------------||NVIDIA Corp|xxxxxxx-xxxxxxx|UFM Enterprise|Subscription
|e4:43:4b:18:3c:e0|2025-07-13 |128 |3 |Invalid ||-----------------------------------------------------------------------------------------
--------------------------------------------||BBK Electronics Corp Ltd|xxxxxxx-xxxxxxx|UFM Enterprise|Subscription
|e4:43:4b:18:49:a0|2025-07-13 |128 |3 |Valid |
```

SA Mode Output Example:

```
root@ufm-production:~# ufmlicense
|----------------------------------------------------------------
----------------------------------------------------------------|
|   Customer ID   |     SN      |    swName       |         Type
|   MAC Address   |   Exp. Date    |Limit| Functionality | Status
|
|----------------------------------------------------------------
----------------------------------------------------------------|
|495760397          |123456778     |UFM Enterprise |Evaluation
|NA               |2090-11-21       |1024  |Advanced       |Valid    |
|----------------------------------------------------------------
----------------------------------------------------------------|
```

To remove a license:

➤

Delete the license file from /opt/ufm/files/licenses.

UFM Configuration

- Initial Configuration
- Optional Configuration

Initial Configuration

After installing the UFM® server software and before running UFM, perform the following:

- Mandatory Configuration:
  - Configure General Settings in gv.cfg
- Additional Configurations Options:
  - General Configuration options
  - Quality of Service
  - Activate and Enable Lossy Configuration Manager (Advanced License Only)
  - Activate and Enable Congestion Control Manager (Advanced License Only)

Configuring Fabric Interface

In most common cases, UFM is run in management mode; the UFM SM manages the InfiniBand fabric. In such cases, the only mandatory configuration is setting the **fabric_interface** parameter.

The fabric interface should be set to one of the InfiniBand IPoIB interfaces, which connect the UFM/SM to the fabric:

```
fabric_interface = ib0
```

Note

- By default, `fabric_interface` is set to `ib0`
- `fabric_interface` must be up and running before UFM startup. Otherwise, UFM will not be able to run.

For additional configuration options, please refer to the Additional Configuration - Optional.

Optional Configuration

General Settings in gv.cfg

Configure general settings in the `conf/gv.cfg` file.

Note

When running UFM in HA mode, the gv.cfg file is replicated to the standby server.

Enabling SHARP Aggregation Manager

SHARP Aggregation Manager is disabled by default. To enable it, set:

```
[Sharp]
sharp_enabled = true
```

Note

Upon startup of UFM or SHARP Aggregation Manager, UFM will resend all existing tenant allocations to SHARP AM.

Enabling Predefined Groups

```
enable_predefined_groups = true
```

Note

By default, pre-defined groups are enabled. In very large-scale fabrics, pre-defined groups can be disabled in order to allow faster startup of UFM.

Enabling Multi-NIC Host Grouping

```
multinic_host_enabled = true
```

Note

Upon first installation of UFM 6.4.1 and above, multi-NIC host grouping is enabled by default. However, if a user is upgrading from an older version, then this feature will be disabled for them.

Note

It is recommended to set the value of this parameter before running UFM for the first time.

Defining Node Description Black-List

Note

Node descriptions from the black-list should not be used for Multi-NIC grouping.

During the process of host reboot or initialization/bringup, the majority of HCAs receive a default label rather than an actual, real description. To prevent the formation of incorrect multi-NIC groups based on these default labels, this feature offers the option to establish a blacklist containing possible node descriptions that should be avoided when grouping Multi-NIC HCAs during host startup. Once a legitimate node description is assigned to the host, the HCAs are organized into multi-NIC hosts based on their respective descriptions. It is recommended to configure this parameter before initiating the UFM for the first time.

For instance, nodes initially identified with descriptions listed in the `exclude_multinic_desc` will not be initially included in Multi-NIC host groups until they obtain an updated, genuine node description.

Modify the `exclude_multinic_desc` parameter in the cv.fg file:

```
exclude_multinic_desc = localhost,generic_name_1,generic_name_2
```

Running UFM Over IPv6 Network Protocol

The default multicast address is configured to an IPv4 address. To run over IPv6, this must be changed to the following in section UFMAgent of gv.cfg.

```
[UFMAgent]
...
# if ufmagent works in ipv6 please set this multicast address to
FF05:0:0:0:0:0:0:15F
mcast_addr = FF05:0:0:0:0:0:0:15F
```

Adding SM Plugin (e.g. lossymgr) to event_plugin_name Option

The following options allow users to set the SM plugin options via the UFM configuration. Once SM is started by UFM, it will start the SM plugin with the specified options.

```
# Event plugin name(s)
event_plugin_name osmufmpi lossymgr
```

Add the plug-in options file to the `event_plugin_options` option:

```
# Options string that would be passed to the plugin(s)
event_plugin_options --lossy_mgr -f <lossy-mgr-options-file-name>
```

These plug-in parameters are copied to the opensm.conf file in Management mode only.

Multi-port SM

SM can use up to eight-port interfaces for fabric configuration. These interfaces can be provided via `/opt/ufm/conf/gv.cfg`. The users can specify multiple IPoIB interfaces or bond interfaces in /opt/ufm/conf/gv.cfg, subsequently, the UFM translates them to GUIDs and adds them to the SM configuration file (/opt/ufm/conf/opensm/opensm.conf). If users specify more than eight interfaces, the extra interfaces are ignored.

```
[Server]

# disabled (default) | enabled (configure opensm with multiple
GUIDs) | ha_enabled (configure multiport SM with high
availability)
 multi_port_sm = disabled
 # When enabling multi_port_sm, specify here the additional
fabric interfaces for OpenSM conf
 # Example: ib1,ib2,ib5 (OpenSM will support the first 8 GUIDs
where first GUID will
 # be extracted the fabric_interface, and remaining GUIDs from
additional_fabric_interfaces
 additional_fabric_interfaces =
```

Note

UFM treats bonds as a group of IPoIB interfaces. So, for example, if bond0 consists of the interfaces ib4 and ib8, then expect to see GUIDs for ib4 and ib8 in opensm.conf.

Note

Duplicate interface names are ignored (e.g. ib1,ib1,ib1,ib2,ib1 = ib1,ib2).

Configuring UDP Buffer

This section is relevant only in cases where `telemetry_provider=ibpm`. (By default, telemetry_provider=telemetry).

To work with large-scale fabrics, users should set the set_udp_buffer flag under the [IBPM] section to "yes" for the UFM to set the buffer size (default is "no").

```
# By deafult, UFM does not set the UDP buffer size. For large
scale fabrics
# it is recommended to increase the buffer size to 4MB (4194304
bits).
set_udp_buffer = yes
# UDP buffer size
udp_buffer_size = 4194304
```

Virtualization

This allows for supporting virtual ports in UFM.

```
[Virtualization]
# By enabling this flag, UFM will discover all the virtual ports
assigned for all hypervisors in the fabric
enable = false
# Interval for checking whether any virtual ports were changed in
the fabric
interval = 60
```

Static SM LID

Users may configure a specific value for the SM LID so that the UFM SM uses it upon UFM startup.

```
[SubnetManager]
# 1- Zero value (Default): Disable static SM LID functionality
and allow the SM to run with any LID.
#    Example: sm_lid=0
# 2- Non-zero value: Enable static SM LID functionality so SM
will use this LID upon UFM startup.
sm_lid=0
```

Note

To configure an external SM (UFM server running in sm_only mode), users must manually configure the opensm.conf file (`/opt/ufm/conf/opensm/opensm.conf`) and align the value of `master_sm_lid` to the value used for `sm_lid` in gv.cfg on the main UFM server.

Configuring Log Rotation

This section enables setting up the log files rotate policy. By default, log rotation runs once a day by cron scheduler.

```
[logrotate]
#max_files specifies the number of times to rotate a file before
it is deleted (this definition will be applied to
#SM and SHARP Aggregation Manager logs, running in the scope of
UFM).
#A count of 0 (zero) means no copies are retained. A count of 15
means fifteen copies are retained (default is 15)
max_files = 15
#With max_size, the log file is rotated when the specified size
is reached (this definition will be applied to
#SM and SHARP Aggregation Manager logs, running in the scope of
UFM). Size may be specified in bytes (default),
#kilobytes (for example: 100k), or megabytes (for example: 10M).
if not specified logs will be rotated once a day.
max_size = 3
```

Configuring UFM Logging

The [Logging] section in the gv.cfg enables setting the UFM logging configurations.

Field	Default Value	Value Options	Description
`level`	WARNING	CRITICAL, ERROR, WARNING, INFO, DEBUG	The definition of the maub logging level for UFM components.

Field	Default Value	Value Options	Description
`smclient_level`	WARNING	CRITICAL, ERROR, WARNING, INFO, DEBUG	The logging level for SM client log messages
`event_log_level`	INFO	CRITICAL, ERROR, WARNING, INFO, DEBUG	The logging level for UFM events log messages
`rest_log_level`	INFO	CRITICAL, ERROR, WARNING, INFO, DEBUG	Logging level for REST API related log messages
`authentication_service_log_level`	INFO	CRITICAL, ERROR, WARNING, INFO, DEBUG	logging level for UFM authenticatior log messages

Field	Default Value	Value Options	Description
log_dir	/opt/ufm/files/log	N/A	It is possible to change the default path to the UFM log directory. The configured log_dir must have read, write and execute permission for ufmapp user (ufmapp group).In case of HA, UFM should be located in the directory which is replicated between the UFM master and standby servers.A change of the default UFM log directory may affect UFM dump creation and inclusion of UFM logs in dump.
max_history_lines	100000	N/A	The maximum number of lines in log files to be shown in UI output for UFM logging.

```
[Logging]
# Optional logging levels: CRITICAL, ERROR, WARNING, INFO, DEBUG.
level = WARNING
smclient_level = WARNING
event_log_level = INFO
rest_log_level = INFO
authentication_service_log_level = INFO
# The configured log_dir must have read, write and execute
permission for ufmapp user (ufmapp group).
log_dir = /opt/ufm/files/log
max_history_lines = 100000
```

Configuring UFM Over Static IPv4 Address

Follow this procedure to to run UFM on a static IP configuration instead of DHCP:

1. Modify the defined management Ethernet interface network script to be static. Run:

   ```
   # vi /etc/sysconfig/network-scripts/ifcfg-enp1s0
   ```

   Update the required interface with the static IP configuration (IP address, netmask, broadcast, and gateway):

```
NAME="enp1s0"DEVICE="enp1s0"
ONBOOT="yes"
BOOTPROTO="static"
IPADDR="10.209.37.153"
NETMASK="255.255.252.0"
BROADCAST="10.209.39.255"
GATEWAY="10.209.36.1"
TYPE=Ethernet
DEFROUTE="yes"
```

2. Add host entries to the /etc/hosts file. Run:

```
# vi /etc/hosts
127.0.0.1   localhost localhost.localdomain localhost4
localhost4.localdomain4
::1         localhost localhost.localdomain localhost6
localhost6.localdomain6

10.209.37.153 <hostname>
```

3. Check hostname. Run:

```
# vi /etc/hostname
<hostname>
```

4. Set up DNS resolution at /etc/resolv.conf. Run:

```
# vi /etc/resolv.conf
search mtr.labs.mlnx
nameserver 8.8.8.8
```

5. Restart network service. Run:

```
service network restart
```

6. Check Configuration. Run:

```
# hostname
<hostname>
# hostname -i
10.209.37.153
```

Configuring Syslog

This configuration enables the UFM to send log messages to syslog, including remote syslog. The configuration described below is located in the [Logging] section of the gv.cfg file.

Field	Default Value	Value Options	Desc
`syslog`	false	True or False	Enab syslo

Field	Default Value	Value Options	Desc
`syslog_addr`	/dev/log # for remote rsyslog_hostname:514	N/A	UFM (syslo For w syslo /dev/l exter value host:p the d serve As th could serve confi remo The /e shoul secti uncor below syslo recep imud 514# recep imtcp 514 Resta servi `serv rest`
`ufm_syslog`	false	True or False	Sets main mess send.
`smclient_syslog`	false	True or False	Sets Open mess send.

Field	Default Value	Value Options	Desc
`event_syslog`	false	True or False	Sets event False Send
`rest_syslog`	false	True or False	Sets UFM mess send.
`authentication_syslog`	false	True or False	Set s authe mess send.
`syslog_level`	WARNING	CRITICAL, ERROR, WARNING, INFO, DEBUG	Sets mess The s comr comp The s sent highe level level secti
`syslog_facility`	LOG_USER	LOG_KERN, LOG_USER, LOG_MAIL, LOG_DAEMON, LOG_AUTH, LOG_SYSLOG, LOG_LPR, LOG_NEWS, LOG_UUCP ,LOG_CRON,LOG_AUTHPRIV, LOG_FTP, LOG_NTP,LOG_SECURITY, LOG_CONSOLE, LOG_SOLCRON	Inclu syslo value facilit

```
syslog = false
#syslog configuration (syslog_addr)
# For working with local syslog, set value to: /dev/log
# For working with external machine, set value to: host:port
syslog_addr = /dev/log
# The configured log_dir must have read, write and execute
permission for ufmapp user (ufmapp group).
log_dir = /opt/ufm/files/log
# Main ufm log.
ufm_syslog = false
smclient_syslog = false
event_syslog = false
rest_syslog = false
authentication_syslog = false
syslog_level = WARNING
# Syslog facility. By default - LOG_USER
#  possible facility codes:
LOG_KERN,LOG_USER,LOG_MAIL,LOG_DAEMON,LOG_AUTH,LOG_SYSLOG,
#  LOG_LPR,LOG_NEWS,LOG_UUCP,LOG_CRON,LOG_AUTHPRIV,LOG_FTP,
LOG_NTP,LOG_SECURITY,LOG_CONSOLE,LOG_SOLCRON
# for reference https://en.wikipedia.org/wiki/Syslog
syslog_facility = LOG_USER
```

Excluding Unhealthy Ports from Fabric Health Report

In gv.cfg file there is a section named **UnhealthyPorts** and parameters in this section are used for unhealthy ports managing in UFM.

Unhealthy port state could be defined by used using UI or REST API request or reported by OpenSM or ibutilities.

UFM has an ability to check periodically fabric ports healthiness and to report unhealthy ports out or to perform automatically predefined isolation action for unhealthy ports.

In addition, using `exclude_unhealthy_ports` key in **UnhealthyPorts** section unhealthy ports could be excluded from ibdiagnet report.

By default, the value for this parameter is set as *false*. It means that unhealthy ports will appear in ibdiagnet reports, but if need to exclude unhealthy port from ibdiagnet reports

this parameter should be set to true and UFM server should be restarted so this action will take effect.

UFM starting flow will configure indiagnet configuration file with appropriate parameters and unhealthy ports will not appear in UFM health and Fabric health reports.

```
[UnhealthyPorts]
enable_ibdiagnet = true
log_level = INFO
syslog = false
# scheduling_mode possible values: fixed_time/interval.
# If fixed_time - ibdiagnet will run every 24 hours on the
specified time - <fixed_time>.
# If interval - ibdiagnet will run first time after <start_delay>
minutes from UFM startup and every <interval> hours (default
scheduling mode).
scheduling_mode = interval
# First ibdiagnet start delay interval (minutes)
start_delay = 5
# ibdiagnet run interval (hours)
interval = 3
# ibdiagnet run at a fixed time (example: 23:17:35)
fixed_time = 23:00:00
# By enabling this flag all the discovered high ber ports will be
marked as unhealthy automatically by UFM
high_ber_ports_auto_isolation = false
# Auto isolation mode - which type of ports should be isolated.
# Options: switch-switch,switch-host,all (default: switch-switch).
auto_isolation_mode = switch-switch
# Trigger Partial Switch ASIC Failure whenever number of
unhealthy ports exceed the defined percent of the total number of
the switch ports.
switch_asic_fault_threshold = 20
# exclude unhealthy ports from ibdiagnet reports
exclude_unhealthy_ports=false
```

Configuration Examples in gv.cfg

The following show examples of configuration settings in the gv.cfg file:

- Polling interval for Fabric Dashboard information

```
ui_polling_interval = 30
```

- [**Optional**] UFM Server local IP address resolution (by default, the UFM resolves the address by gethostip). UFM Web UI should have access to this address.

```
ws_address = <specific IP address>
```

- HTTP/HTTPS Port Configuration

```
# WebServices Protocol (http/https) and Port
ws_port = 8088
ws_protocol = http
```

- Connection (port and protocol) between the UFM server and the APACHE server

```
ws_protocol = <http or https>
ws_port = <port number>
```

For more information, see Launching a UFM Web UI Session.

- SNMP `get-community` string for switches (fabric wide or per switch)

```
# default snmp access point for all devices
[SNMP]
port = 161
gcommunity = public
```

- Enhanced Event Management (Alarmed Devices Group)

```
[Server]
auto_remove_from_alerted = yes
```

- Log verbosity

```
[Logging]
# optional logging levels
#CRITICAL, ERROR, WARNING, INFO, DEBUG
level = INFO
```

For more information, see "UFM Logs".

- Settings for saving port counters to a CSV file

```
[CSV]
write_interval = 60
ext_ports_only = no
```

For more information, see "Saving the Port Counters to a CSV File".

- Max number of CSV files (UFM Advanced)

```
[CSV]
max_files = 1
```

For more information, see "Saving Periodic Snapshots of the Fabric (Advanced License Only)".

> **Note**
>
> The access credentials that are defined in the following sections of the conf/gv.cfg file are used only for initialization:
>
> - SSH_Server
> - SSH_Switch
> - TELNET
> - IPMI
> - SNMP
> - MLNX_OS
>
> To modify these access credentials, use the UFM Web UI. For more information, see "Device Access".

- Configuring the UFM communication protocol with MLNX-OS switches. The available protocols are:
  - http
  - https (default protocol for secure communication)

For configuring the UFM communication protocol after fresh installation and prior to the first run, set the MLNX-OS protocol as shown below.

Example:

```
[MLNX_OS]
protocol = https
port = 443
```

Once UFM is started, all UFM communication with MLNX-OS switches will take place via the configured protocol.

For changing the UFM communication protocol while UFM is running, perform the following:

1. Set the desired protocol of MLNX-OS in the conf/gv.cfg file (as shown in the example above).
2. Restart UFM.
3. Update the MLNX-OS global access credentials configuration with the relevant protocol port. Refer to "Device Access" for help.

   For the http protocol - default port is 80.

   For the https protocol - default port is 443.
4. Update the MLNX-OS access credentials with the relevant port in all managed switches that have a valid IP address.

Managing Dynamic Telemetry

The management of dynamic telemetry instances involves the facilitation of user requests for the creation of multiple telemetry instances. As part of this process, the UFM enables users to establish new UFM Telemetry instances according to their preferred counters and configurations. These instances are not initiated by the UFM but rather are monitored for their operational status through the use of the UFM Telemetry bring-up tool

For more information on the supported REST APIs, please refer to UFM Dynamic Telemetry Instances REST API.

The configuration parameters can be found in the gv.cfg configuration file under the DynamicTelemetry section.

Name	Description	De val
max_instances	Maximum number of simultaneous running UFM Telemetries.	5

Name	Description	De val
new_instance_delay	Delay time between the start of two UFM Telemetry instances, in minutes.	5
update_discovery_delay	The time to wait before updating the discovery file of each telemetry instance if the fabric has changed, in minutes.	10
endpoint_timeout	Telemetry endpoint timeout, in seconds.	5
bringup_timeout	Telemetry bringup tool timeout, in seconds.	60
initial_exposed_port	Initial port for the available range of ports (range(initial_exposed_port, initial_exposed_port + max_instances)).	90(
instances_sessions_compatibility_interval	Every instances_sessions_compatibility_interval minutes the UFM verifies compliance between instances and sessions to avoid zombie sessions. if 0 is configured this process won't be activate	10

SM Trap Handler Configuration

The SMTrap handler is the SOAP server that handles traps coming from OpenSM.

There are two configuration values related to this service:

- `osm_traps_debounce_interval` – defines the period the service holds incoming traps
- `osm_traps_throttle_val` – once `osm_traps_debounce_interval` elapses, the service transfers `osm_traps_throttle_val` to the Model Main

Note

By default, the SM Trap Handler handles up to 1000 SM traps every 10 seconds.

Setting CPU Affinity on UFM

This feature allows setting the CPU affinity for the major processes of the UFM (such as ModelMain, SM, SHARP, Telemetry).

In order to increase the UFM's efficiency, the number of context-switches is reduced. When each major CPU is isolated, users can decrease the number of context-switches, and the performance is optimized.

The CPU affinity of these major processes is configured in the following two levels:

- Level 1- The major processes initiation.
- Level 2- Preceding initiation of the model's main subprocesses which automatically uses the configuration used in level 1 and designates a CPU for each of the sub-processes.

According to user configuration, each process is assigned with affinity.

By default, this feature is disabled. In order to activate the feature, configure `Is_cpu_affinity_enabled` with true, check how many CPUs you have on the machine, and set the desired affinity for each process.

For example:

```
[CPUAffinity]
Is_cpu_affinity_enabled=true
Model_main_cpu_affinity=1-4
Sm_cpu_affinity=5-13
SHARP_cpu_affinity=14-22
Telemetry_cpu_affinity=22-23
```

The format should be a comma-separated list of CPUs. For example: 0,3,7-11.

The ModelMain should have four cores, and up to five cores. The SM should have as many cores as you can assign. You should isolate between the ModelMain cores and the SM cores.

SHARP can be assigned with the same affinity as the SM. The telemetry should be assigned with three to four CPUs.

Quality of Service (QoS) Support

Infiniband Quality of Service (QoS) is disabled by default in the UFM SM configuration file.

To enable it and benefit from its capabilities, set the qos flag to TRUE in the /opt/ufm/files/conf/opensm/opensm.conf file.

Example:

```
# Enable QoS setup
qos FALSE
```

Note

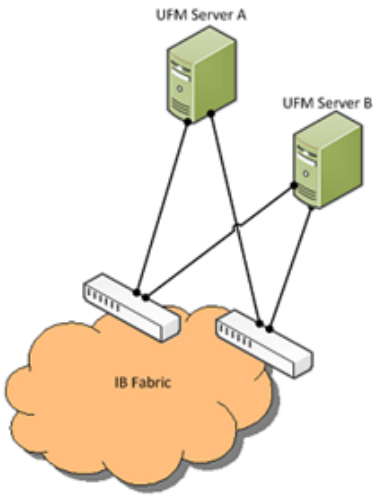
The QoS parameters settings should be carefully reviewed before enablement of the qos flag. Especially, sl2vl and VL arbitration mappings should be correctly defined.

For information on Enhanced QoS, see Appendix – SM Activity Report.

UFM Failover to Another Port

When the UFM Server is connected by two or more InfiniBand ports to the fabric, you can configure UFM Subnet Manager failover to one of the other ports. When failure is detected on an InfiniBand port or link, failover occurs without stopping the UFM Server or other related UFM services, such as mysql, http, DRDB, and so on. This failover process prevents failure in a standalone setup, and preempts failover in a High Availability setup, thereby saving downtime and recovery.

Network Configuration for Failover to IB Port



To enable UFM failover to another port:

- Configure bonding between the InfiniBand interfaces to be used for SM failover. In an HA setup, the UFM active server and the UFM standby server can be connected differently; but the bond name must be the same on both servers.

- Set the value of fabric_interface to the bond name. using the /opt/ufm/scripts/*change_fabric_config.sh* command as described in Configuring General Settings in gv.cfg. If ufma_interface is configured for IPoIB, set it to the bond name as well. These changes will take effect only after a UFM restart. For example, if bond0 is configured on the ib0 and ib1 interfaces, in gv.cfg, set the parameter fabric_interface to bond0.

- If IPoIB is used for UFM Agent, add bond to the ufma_interfaces list as well.

When failure is detected on an InfiniBand port or link, UFM initiates the give-up operation that is defined in the Health configuration file for OpenSM failure. By default:

- UFM discovers the other ports in the specified bond and fails over to the first interface that is up (SM failover)

- If no interface is up:

    - In an HA setup, UFM initiates UFM failover

    - In a standalone setup, UFM does nothing

If the failed link becomes active again, UFM will select this link for the SM only after SM restart.

Configuring Managed Switches Info Persistency

UFM uses a periodic system information-pulling mechanism to query managed switches inventory data. The inventory information is saved in local JSON files for persistency and tracking of managed switches' status.

Upon UFM start up, UFM loads the saved JSON files to present them to the end user via REST API or UFM WebUI.

After UFM startup is completed, UFM pulls all managed switches data and updates the JSON file and the UFM model periodically (the interval is configurable). In addition, the JSON files are part of UFM system dump.

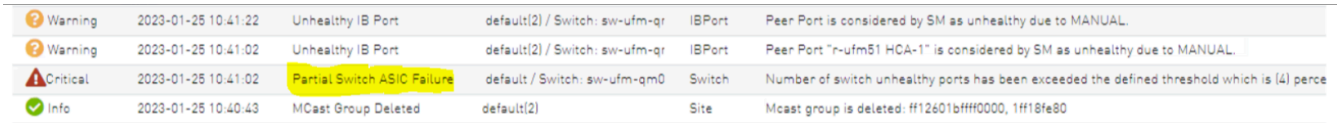
The following parameters allow configuration of the feature via gv.cfg fie:

```
[SrvMgmt]
# how often UFM should send json requests for sysinfo to switches
(in seconds)
systems_poll = 180
# To create UFM model in large setups might take a lot of time.
# This is an initial delay (in minutes) before starting to pull
sysinfo from switches.
systems_poll_init_timeout = 5
# to avoid sysinfo dump overloading and multiple writing to host
# switches sysinfo will be dumped to disc in json format every
set in this variable
# sysinfo request. If set to 0 - will not be dumped, if set to 1 -
will be dumped every sysinfo request
# this case (as example defined below) dump will be created every
fifth sysinfo request, so if system_poll is 180 sec (3 minutes)
sysinfo dump to the file will e performed every 15 minutes.
sysinfo_dump_interval = 5
# location of the sysinfo dump file (it is in /opt/ufm/files/logs
(it will be part of UFM dump)
sysinfo_dump_file_path = /opt/ufm/files/log/sysinfo.dump
```

Configuring Partial Switch ASIC Failure Events

UFM can identify switch ASIC failure by detecting pre-defined portion of the switch ports, reported as unhealthy. By default, this portion threshold is set to 20% of the total switch ports. Thus, the UFM will trigger the partial switch ASIC event in case the number of unhealthy switch ports exceeds 20% of the total switch ports.

You can configure UFM t o control Partial Switch ASIC Failure events. To configure, you may use the gv.cfg file by updating the value of `switch_asic_fault_threshold` parameter under the UnhealthyPorts section. For an example, in case the switch has 32 ports, once 7 ports are detected as unhealthy ports, the UFM will trigger the partial switch ASIC event. Example:



Warning	2023-01-25 10:41:22	Unhealthy IB Port	default(2) / Switch: sw-ufm-qr	IBPort	Peer Port is considered by SM as unhealthy due to MANUAL.
Warning	2023-01-25 10:41:02	Unhealthy IB Port	default(2) / Switch: sw-ufm-qr	IBPort	Peer Port "r-ufm51 HCA-1" is considered by SM as unhealthy due to MANUAL.
Critical	2023-01-25 10:41:02	Partial Switch ASIC Failure	default / Switch: sw-ufm-qm0	Switch	Number of switch unhealthy ports has been exceeded the defined threshold which is (4) perce
Info	2023-01-25 10:40:43	MCast Group Deleted	default(2)	Site	Mcast group is deleted: ff12601bffff0000, 1ff18fe80

Enabling Network Fast Recovery

> **Note**
>
> To enable the Network Fast Recovery feature, ensure that all switches in the fabric use the following MLNX-OS/firmware versions:
>
> - MLNX-OS version 3.10.6004 and up
> - Quantum firmware versions:
>   - Quantum FW v27.2010.6102 and up
>   - Quantum2 FW v31.2010.6102 and up

Fast recovery is a switch-firmware based facility for isolation and mitigation of link-related issues. This system operates in a distributed manner, where each switch is programmed with a simple set of rule-based triggers (conditions) and corresponding action protocols. These rules permit the switch to promptly react to substandard links within its locality, responding at a very short reaction time - as little as approximately 100 milliseconds. The policy is provided and managed via the UFM and SM channel. Moreover, every autonomous action taken by a switch in the network is reported to the UFM.

The immediate reactions taken by the switch enable SHIELD and pFRN. These mechanisms collaborate to rectify routing within the proximity of the problematic link before it can disrupt transactions at the transport layer. Importantly, this process occurs rapidly, effectively limiting the spreading of congestion to a smaller segment of the network.

To use the Network Fast Recovery feature, you need to enable the designated trigger (condition) in the gv.cfg file. By doing this, you can specify which of the below four triggers the UFM will support.

As stated in the gv.cfg file, the feature is disabled by default and the below are the supported fields and options:

```
[NetworkFastRecovery]
# Fast Recovery configuration.
# Supported values:
#    0: Ignore fast recovery related MADs and configuration
(default)
#    1: Disable fast recovery
#    2: Enable fast recovery
fast_recovery_mode = 0

# This will be supported by the Network Fast Recovery.
network_fast_recovery_conditions =
SWITCH_DECISION_CREDIT_WATCHDOG,SWITCH_DECISION_RAW_BER,SWITCH_DEC
```

Note

To enable the Network Fast Recovery feature, the value of `fast_recovery_mode` should be set to 2. For the change to take effect, restart of UFM Enterprise is required.

Parameter	Description
SWITCH_DECISION_CREDIT_WATCHDOG	The Switch decided to close the port due to Credit watchdog
SWITCH_DECISION_RAW_BER	The Switch decided to close the port due to High raw errors
SWITCH_DECISION_EFFECTIVE_BER	The Switch decided to close the port due to High effective errors (after FEC)
SWITCH_DECISION_SYMBOL_BER	The Switch decided to close the port due to High symbol errors (after PLR)

By default, the Network Fast Recovery feature operates in monitoring mode. This means the switch does not reset ports, however, it reports issues related to them.To view these port-related issues, you must deploy the UFM PMC (Packet Monitoring Collector) plugin and use its UI to access the relevant network events.

For more details on the PMC plugin, including deployment instructions and how to view Network Fast Recovery events, please refer to Packet Level Monitoring Collector (PMC) Plugin.

Disabling Rest Roles Access Control

By default, the Rest Roles Access Control feature is enabled. It can be disabled by setting the `roles_access_control_enabled` flag to false:

```
[RolesAccessControl]
roles_access_control_enabled = true
```

Enabling/Disabling Authentication

Kerberos Authentication

By default, Kerberos Authentication is disabled. To enable it, set the `kerberos_auth_enabled` flag to true. Additionally, provide the required configurations such as `kerberos_cred_key_path`, `kerberos_use_local_name` and `kerberos_auto_sign_up`.

```
[KerberosAuth]
# This section responsible to manage kerberos authentication
# Set to true to enable the kerberos auth feature, and set to false
to disable it. Default is false.
kerberos_auth_enabled = false
# The path of the keytab file containing credentials for GSSAPI
authentication.
kerberos_cred_key_path = /etc/kadm5.keytab
# Set to true to configure the Apache server to map authenticated
principal names (which represent different clients) to local
usernames,
# and set to false to use the principle names as usernames. Default
is true (this value will be reflected in the 'GssapiLocalName' directive
in Apache).
kerberos_use_local_name = true
# Set to true to enable auto sign up of users who do not exist in
UFM DB. Default is true.
kerberos_auto_sign_up = true
# The default role assigned to create users if they do not exist when
'kerberos_auto_sign_up' is set to true.
kerberos_default_role = System_Admin
```

kerberos_auth_enabled: By default, Kerberos authentication remains disabled. To activate it, the user must set this flag to 'true' and then restart UFM.

kerberos_cred_key_path: This specifies the path to the keytab file containing credentials for GSSAPI authentication.

kerberos_use_local_name: Set to true to configure the Apache server to map authenticated principal names (which represent different clients) to local usernames, and set to false to use the principal names as usernames. Default is true (this value will be reflected in the '`GssapiLocalName`' directive in Apache).

kerberos_auto_signup: For successful authentication via Kerberos, the user must already exist within the UFM database, otherwise, the authentication will be refused by UFM. If this property is set to 'true,' UFM will create the non-existing users in the UFM DB.

kerberos_default_role: The default role is assigned to create users if they do not exist when '`kerberos_auto_sign_up`' is set to true.

Finally, restart the UFM to use Kerberos authentication.

UFM Authentication Server

By default, UFM Authentication Server is enabled. To disable it, you need to set the "`auth_service_enabled`" parameter to '`false`' and then restart the UFM service to initiate the authentication server. Additionally, you can use enable/disable flags for Basic, Session, and Token authentication:

```
[AuthService]
auth_service_enabled = true
auth_service_interface = 127.0.0.1
auth_service_port = 8087 # the serving port for the authentication
server
basic_auth_enabled = true
session_auth_enabled = true
token_auth_enabled = true
```

Azure AD Authentication

By default, Azure AD Authentication is disabled. To enable it, set the `azure_auth_enabled` flag to 'true'. Additionally, provide the required configurations from the Azure AD Application such as TENANT_ID, CLIENT_ID and CLIENT_SECRET which can be found under the "**Overview**" section of the registered application in the Azure portal. Finally, the UFM Authentication Server should be enabled to use the Azure AD Authentication.

```
[AzureAuth]
azure_auth_enabled = false
# TENANT ID of app registration
TENANT_ID =
# Application (client) ID of app registration
CLIENT_ID =
# Application's generated client secret
CLIENT_SECRET =
```

Adjusting UFM Configuration Files Based on Fabric Size

This function allows users to automate the process of updating the UFM configuration files by parsing a primary configuration file called large_scale_subnet.cfg file and applying the values to multiple target files or resetting to default values using the small_scale_subnet.cfg.

The below are instructions on how to use a Python script to parse a configuration file (`large_scale_subnet.cfg`) and update the values of specific parameters in multiple target UFM configuration files (`gv.cfg, reports.cfg, opensm.cfg` , and `sharp_am.cfg`). The script can operate in two modes:

- **Large Scale Subnet Mode**: This mode directly updates the UFM configuration files based on the parsed configuration from the `large_scale_subnet.cfg` file.
- **Small Scale Subnet (Default) Mode**: Sets the UFM configuration files to their default values by parsing the `small_scale_subnet.cfg file` .

Configuration File and Parameters

The primary configuration file contains all the parameters, and their values must be updated over the multiple UFM configuration files.

Primary Configuration Files

- `/opt/ufm/files/conf/ large_scale_subnet.cfg`

- `/opt/ufm/files/conf/ small_scale_subnet.cfg`

Target UFM Configuration Files

- `/opt/ufm/files/conf/gv.cfg`
- `/opt/ufm/files/conf/reports.cfg`
- `/opt/ufm/files/conf/opensm/opensm.conf`
- `/opt/ufm/files/conf/sharp/sharp_am.cfg`

Example structure of `large_scale_subnet.cfg` and `small_scale_subnet.cfg`:

```
[GV]
[GV.Server]
# disabled (default) | enabled (configure opensm with multiple
GUIDs) | ha_enabled (configure multiport SM with high
availability).
multi_port_sm = ha_enabled
# report_events that will determine which trap to send  to ufm
all/security/none
report_events = security

[GV.FabricAnalysis]
# initial_delay (in minutes) - the initial delay for running
fabric analysis for the first time after UFM was started
initial_delay = 10


[GV.logrotate]
#max_files specifies the number of times to rotate a file before
it is deleted.
#A count of 0 (zero) means no copies are retained. A count of 10
means fifteen copies are retained (default is 10)
max_files = 10

[REPORTS]
[REPORTS.FabricHealth]
# Fabric health report timeout
timeout = 1800

[REPORTS.TopologyCompare]
# Topology compare report timeout
timeout = 1800

[REPORTS.FabricAnalysis]
# Fabric analysis report timeout
timeout = 1800
```

```
[OPENSM]
#Amount of physical port to handle in one shot
virt_max_ports_in_process = 512
max_op_vls = 2
qos = TRUE
# Single MAD Sl2vl for all ports
use_optimized_slvl = TRUE
# Timeout for long MAD config time. might need to change 1000
long_transaction_timeout = 500
routing_engine = ar_updn
use_ucast_cache = TRUE
root_guid_file = /opt/ufm/files/conf/opensm/root_guid.conf
pgrp_policy_file = /opt/ufm/files/conf/opensm/pgrp_policy.conf

[SHARP]
ib_qpc_sl = 1
fabric_update_interval = 10
lst_file_timeout = 10
lst_file_retries = 30
max_tree_radix = 80
generate_dump_files = TRUE
dynamic_tree_allocation = TRUE
dynamic_tree_algorithm = 1
smx_keepalive_interval = 10
```

Script Usage Example in the CLI:

- **Large Scale Subnet Mode:**
  ```
  /opt/ufm/scripts/set_ufm_scale_profile.sh --mode
  large_scale_subnet --force_update
  ```
- **Small Scale Subnet (Default) Mode:**
  ```
  /opt/ufm/scripts/ set_ufm_scale_profile.sh --mode
  small_scale_subnet
  ```

The `force_update` script parameter adds any parameters found in `large_scale_subnet.cfg` and `small_scale_subnet.cfg` that are not present in the UFM configuration files. For example, if a user adds a new parameter called `test_param` = 500 to the `large_scale_subnet.cfg` file under the [Server] section and this parameter does not exist in the `gv.cfg` file, running the script with the `--force_update` option will add `test_param` = 500 to the [Server] section of the `gv.cfg file.`

Expected Output

1. **Large Scale Subnet Mode:**

- The script reads `large_scale_subnet.cfg`.
- It updates the parameters in the target UFM configuration files based on the parsed data.
- It logs messages for any skipped parameters.

1. **Small Scale Subnet - Default Mode:**

- The script reads `small_scale_subnet.cfg`.
- It updates the parameters in the target UFM configuration files based on the parsed data.
- It logs messages for any skipped parameters or adds the parameter to the configuration file if the `force_update` was True.

> **Note**
>
> **Note:** In case of the script running failure, the script will reset the UFM configuration files to their default values.

Setting up Telemetry in UFM

Setting up telemetry deploys UFM Telemetry as bare metal on the same machine. Historical data is sent to SQLite database on the server and live data becomes available via UFM UI or REST API.

Enabling UFM Telemetry

The UFM Telemetry feature is enabled by default and the provider is the UFM Telemetry. The user may change the provider via flag in `conf/gv.cfg`

The user may also disable the History Telemetry feature in the same section.

```
[Telemetry]
history_enabled=True
```

Changing UFM Telemetry Default Configuration

There is an option to configure parameters on a telemetry configuration file which takes effect after restarting the UFM or failover in HA mode.The `launch_ibdiagnet_config.ini` default file is located under `/opt/ufm/conf/telemetry_defaults` and is copied to the telemetry configuration location ((`/opt/ufm/conf/telemetry`) upon startup UFM.

All values taken from the default file take effect at the deployed configuration file except for the following:

Note that normally the user does not have to do anything and they get two pre-configured instances – one for low frequency and one for higher-frequency sampling of the network.

Value	Description
`hca`	-
`scope_file`	-

Value	Description
`plugin_env_PROMETHEUS_ENDPOINT`	The port on which HTTP endpoint is configured
`plugin_env_PROMETHEUS_INDEXES`	Configures how data is indexed and stored in memory
`config_watch_enabled=true`	Configures network watcher to inform ibdiagnet that network topology has changed (as ibdiagnet lacks the ability to re-discover network changes)
`plugin_env_PROMETHEUS_CSET_DIR`	Specifies where the counterset files, which define the data to be retrieved and the corresponding counter names.
`num_iterations`	The number of iterations to run before 'restarting', i.e. rediscovering fabric.
`plugin_env_CLX_RESTART_FILE`	A file that is 'touched' to indicate that an ibdiagnet restart is necessary

The following attributes are configurable via the gv.cfg:

- sample_rate (gv.cfg → dashboard_interval) – only if manual_config is set to false
- prometheus_port

Supporting Generic Counters Parsing and Display

As of UFM v6.11.0, UFM can support any numeric counters from the HTTP endpoint. The list of supported counters are fetched upon starting the UFM from all the endpoints that are configured.

Some of the implemented changes are as follows:

1. Counter naming – all counters naming convention is extracted from the HTTP endpoint. The default `cset` file is configured as follows:

   "`Infiniband_LinkIntegrityErrors=^LocalLinkIntegrityErrorsExtended$`" to get this name to the UFM.

Counters received as floats should contain an "_f" suffix such as: `Infiniband_CBW_f=^infiniband_CBW$`

2. Attribute units – To see units of a specific counter on the UI graphs, configure the `cset` file to have the counter returned as "`counter_name_u_unit`".

3. Telemetry History:

   The SQLite history table `(/opt/ufm/files/sqlite/ufm_telemetry.db – telemetry_calculated)`, contains the new naming convention of the telemetry counters.

   In the case of an upgrade, all previous columns that were configured are renamed following the new naming convention, and then, the data is saved.if a new counter that is not in the table needs to be supported, the table is altered upon UFM start.

4. New counter/`cset` to fetch – if there is a new `cset` /counter that needs to be supported AFTER the UFM already started, preform system restart.

5. Created New API/UfmRestV2/telemetry/counters for the UI visualization. This API returns a dictionary containing the counters that the UFM supports, based on the fetched URLs and their units (if known).

Supporting Multiple Telemetry Instances Fetch

This functionality allows users to establish distinct Telemetry endpoints that are defined to their preferences.

Users have the flexibility to set the following aspects:

- Specify a list of counters they wish to pull. This can be achieved by selecting from an existing, predefined counters set (`cset file`) or by defining a new one.
- Set the interval at which the data should be pulled.

Upon initiating the Telemetry endpoint, users can access the designated URL to fetch the desired counter data.

To enable this feature, under the [Telemetry] section in `gv.cfg,` the flag named "`additional_cset_ur`l" holds the list of additional URLs to be fetched.

the URLs should be separated by " " (with a space) and should follow the following format: http://:/csv/. For example http://10.10.10.10:9001/csv/minimal http://10.10.10.10:9002/csv/test.

> **Note**
>
> Only csv extensions are supported.

Each UFM Telemetry instance run by UFM can support multiple cset (counters set) in parallel.If the user would like to have a second cset file fetched by UFM and exposed by the same UFM Telemetry instance, the new cset file should be placed under `/opt/ufm/files/conf/telemetry/prometheus_configs/cset/` and configured in gv.cfg to fetch its data as described above.

Low-Frequency (Secondary) Telemetry

As a default configuration, a second UFM Telemetry instance runs, granting access to an extended set of counters that are not available in the default telemetry session. The default telemetry session is used for the UFM Web UI dashboard and user-defined telemetry views. These additional counters can be accessed via the following API endpoint: **http://<UFM_IP>:9002/csv/xcset/low_freq_debug**.It is important to note that these exposed counters are not accessible through UFM's REST APIs.All the configurations for the second telemetry can be found under `/opt/ufm/files/conf/secondary_telemetry/,` where the defaults are located under `/opt/ufm/files/conf/secondary_telemetry_defaults/`. The second telemetry instance also allows telemetry data to be exposed on disabled ports, although this feature can be disabled if desired.

The relevant flags in the gv.cfg file are as follows:

- `secondary_telemetry` = true (To enable or disable the entire feature)
- `secondary_endpoint_port` = 9002 (The endpoint's exposed port)
- `secondary_disabled_ports` = true (If set to true, secondary telemetry will expose data on disabled ports)

- `secondary_slvl_support` = false (if set to true, low-frequency (secondary) Telemetry will collect counters per slvl, the corresponding supported xcset can be found under /opt/ufm/files/conf/secondary_telemetry/prometheus_configs/cset/low_freq_debug_

The counters that are supported by default, collected, and exposed can be located in the directory `/opt/ufm/files/conf/secondary_telemetry/prometheus_configs/cset/low_f` .

For the list of low-frequency (secondary) telemetry fields and available counters, please refer to Low-Frequency (Secondary) Telemetry Fields.

Low-Frequency (Secondary) Telemetry - Exposing IPv6 Counters

To allow the low-frequency (secondary) telemetry instance to expose counters on its IPv6 interfaces, perform the following:

1. Change the following flag in the gv.cfg:

```
secondary_ip_bind_addr =0:0:0:0:0:0:0:0
```

2. Restart UFM telemetry or restart UFM.

Stopping Telemetry Endpoint Using CLI Command

To stop low-frequency (secondary) telemetry endpoint only using the CLI you may run the following command:

```
/etc/init.d/ufmd ufm_telemetry_secondary_stop
```

Exposing Switch Aggregation Nodes Telemetry

To expose switches SHARP aggregation nodes telemetry, follow the below steps:

- Configure the low-frequency (secondary) telemetry instance. Run:

```
vi
/opt/ufm/files/conf/secondary_telemetry_defaults/launch_ibdiag
```

  - Set the following:
    - `arg_16=--sharp --sharp_opt dsc`
    - `plugin_env_CLX_EXPORT_API_SKIP_SHARP_PM_COUNTERS=0`
  - Add the wanted attributes to the default `xcset` or to a new one:
    - New `xcset` –
      - ```
        vi
        /opt/ufm/files/conf/secondary_telemetry/prometheus
        for your choise>.xcset
        ```

        ```
        ```
      - After restarting, query curl `http://<UFM_IP>:9002/csv/xcset/<chosen_name>`
    - Existing `xcset` –
      - ```
        vi
        /opt/ufm/files/conf/secondary_telemetry/prometheus
        ```
-
  -
    - Add the following attributes:

- `packet_sent`
- `ack_packet_sent`
- `retry_packet_sent`
- `rnr_event`
- `timeout_event`
- `oos_nack_rcv`
- `rnr_nack_rcv`
- `packet_discard_transport`
- `packet_discard_sharp`
- `aeth_syndrome_ack_packet`
- `hba_sharp_lookup`
- `hba_received_pkts`
- `hba_received_bytes`
- `hba_sent_ack_packets`
- `rcds_sent_packets`
- `hba_sent_ack_bytes`
- `rcds_send_bytes`
- `hba_multi_packet_message_dropped_pkts`
- `hba_multi_packet_message_dropped_bytes`

- Restart telemetry:

```
/etc/init.d/ufmd ufm_telemetry_stop
/etc/init.d/ufmd ufm_telemetry_start
```

Exposing Performance Histogram Counters for Egress Queue Depth Indications (Secondary) Telemetry

To enable the secondary telemetry instance to expose performance histogram counters for all VLs, perform the following:

1. Change the following flag in the `gv.cfg` file:

   ```
   queue_depth_indications_all_vls = true
   ```

   If this flag remains set to `false`, the secondary telemetry instance will only collect counters for VLs 0 and 1.

2. Restart UFM telemetry or restart UFM.

After the secondary telemetry instance restarts, you can find the collected counters at:

```
/opt/ufm/conf/secondary_telemetry/prometheus_configs/cset/low_freq.
```

Running UFM Server Software

Before running UFM:

- Perform Initial Configuration

- Ensure that all ports used by the UFM server for internal and external communication are open and available. For the list of ports, see Used Ports.

  You can run the UFM server software in the following modes:

-
  - Running UFM Server Software in Management Mode
  - Running UFM Software in High Availability Mode
  - Running UFM in High Availability with failover to an external SM

> **Note**
>
> In Management or High Availability mode, ensure that all Subnet Managers in the fabric are disabled *before* running UFM. Any remaining active Subnet Managers will prevent UFM from running.

Running UFM Server Software in Management Mode

After installing, run the UFM Server by invoking:

```
systemctl start ufm-enterprise.service
```

> **Note**
>
> /etc/init.d/ufmd - Available for backward compatibility.

Log files are located under */opt/ufm/files/log* (the links to log files are in */opt/ufm/log*).

Running UFM Software in High Availability Mode

On the Master server, run the UFM Server by invoking:

```
ufm_ha_cluster start
```

You can specify additional command options for the `ufmha` service.

ufm_ha_cluster Command Options

Command	Description
start	Starts UFM HA cluster.
stop	Stops UFM HA cluster.
failover	Initiates failover (change mastership from local server to remote server).
takeover	Initiates takeover (change mastership from remote server to local server).
status	Shows current HA cluster status.
cleanup	Cleans the HA configurations on this node.
help	Displays help text.

HTTP/HTTPS Configuration

By default, UFM is configured to work with the secured HTTPS protocol.

After installation, the user can change the the Web Server configuration to communicate in secure (HTTPS) or non-secure (HTTP) protocol.

For changing the communication protocol, use the following parameter under the [Server] section in the `gv.cfg` file:

- ws_protocol = https

Changes will take effect after restarting UFM.

UFM Internal Web Server Configuration

UFM uses Apache as the main Web Server for client external access. The UFM uses an internal web server process to where the Apache forwards the incoming requests.

By default, the internal web server listens to the local host interface (127.0.0.1) on port 8000.

For changing the listening local interface or port, use the following parameters under the [Server] section in the `gv.cfg` file:

- rest_interface = 127.0.0.1
- rest_port = 8000

Changes will take effect after restarting UFM.

User Authentication

UFM User Authentication is based on standard Apache User Authentication. Each Web Service client application must authenticate against the UFM server to gain access to the system.

The UFM software comes with one predefined user:

- Username: admin
- Password: 123456

You can add, delete, or update users via User Management Tab.

UFM Authentication Server

The UFM Authentication Server, a centralized HTTP server, is responsible for managing various authentication methods supported by UFM.

Configurations of the UFM Authentication Server

The UFM Authentication Server is designed to be configurable and is initially turned off by default. This means that existing authentication methods are managed either by the native Apache functionality (such as Basic, Session, and Client Certificate authentication) or at the UFM level (including Token-Based authentication and Proxy Authentication).

Enabling the UFM Authentication Server provides a centralized service that oversees all supported authentication methods within a single service, consolidating them under a unified authentication API.

Apache utilizes the authentication server's APIs to determine a user's authentication status.

To enable the UFM Authentication Server, refer to Enabling UFM Authentication Server.

All activities of the UFM Authentication Server are logged in the `authentication_service.log` file, located at `/opt/ufm/files/log`.

Azure AD Authentication

Microsoft Azure Authentication is a service provided by Microsoft Azure, the cloud computing platform of Microsoft. It is designed to provide secure access control and authentication for applications and services hosted on Azure.

UFM supports Authentication using Azure Active Directory, and to do so, you need to follow the following steps:

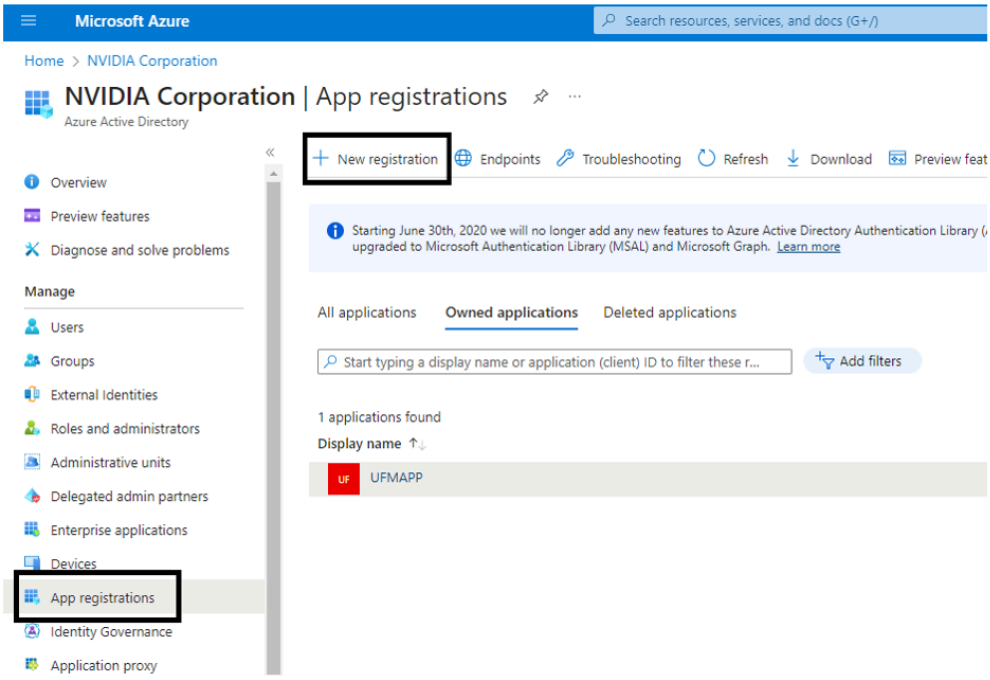
Register UFM in Azure AD Portal

To log in via Azure, UFM must be registered in the Azure portal using the following steps:

1. Log in to Azure Portal, then click "**Azure Active Directory"** in the side menu.

2. If you have access to more than one tenant, select your account in the upper right. Set your session to the Azure AD tenant you wish to use.

3. Under "**Manage"** in the side menu, click App Registrations > New Registration.

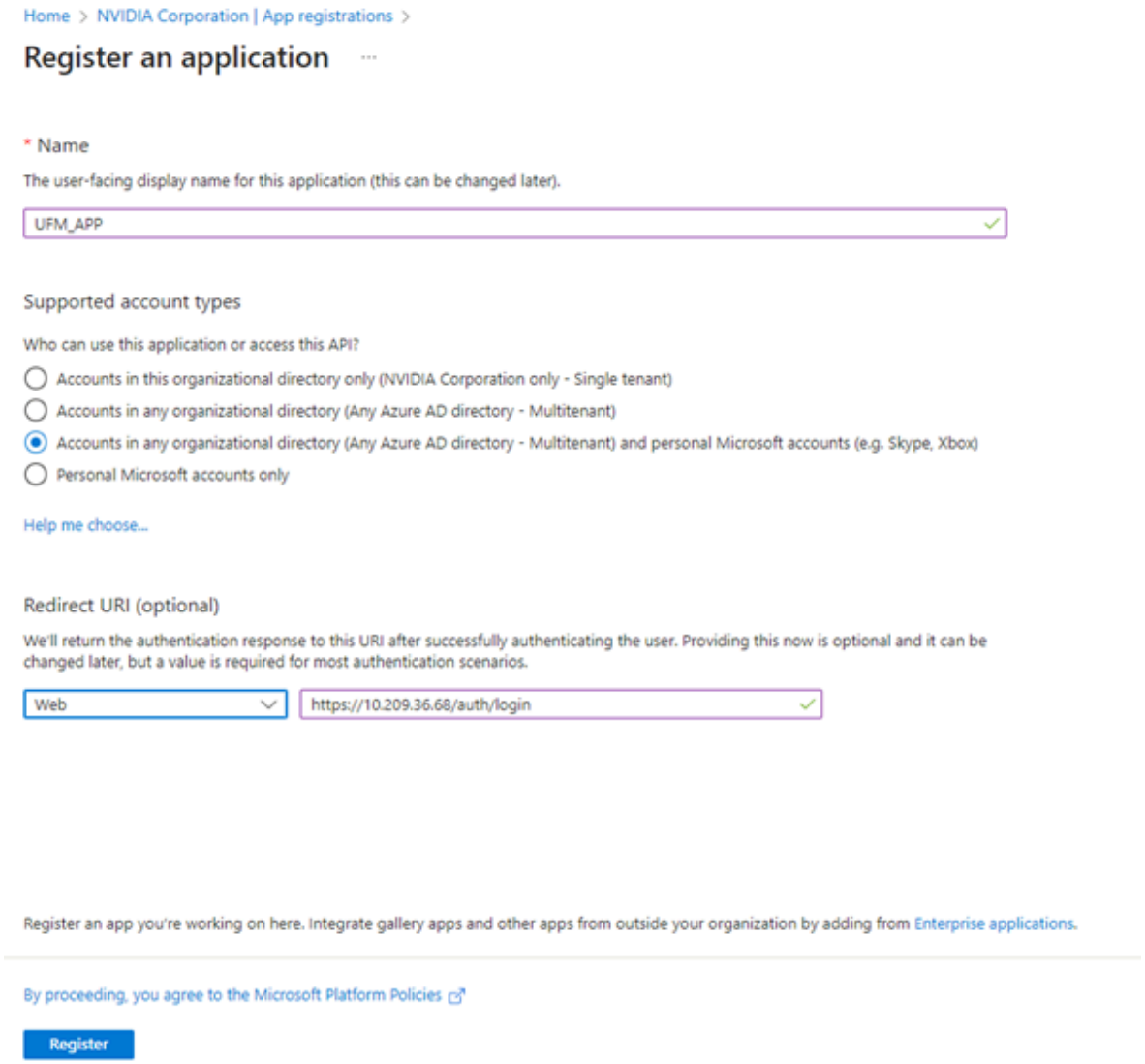


4. Provide the application details:

    1. **Name**: Enter a descriptive name.

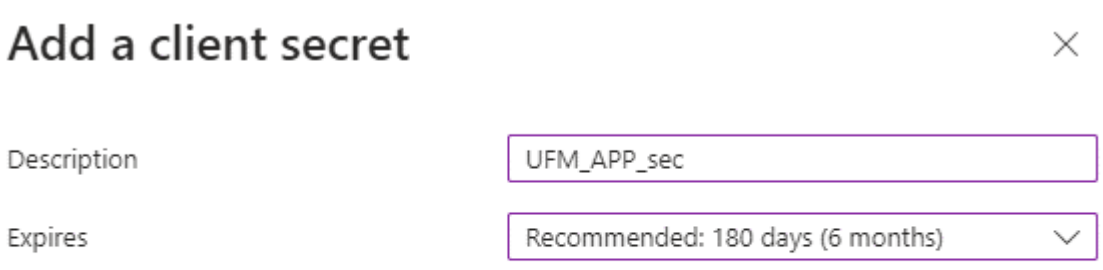
    2. **Supported account types**: Account types that are allowed to login and use the registered application.

    3. **Redirect URL**: select the app type **Web**, and Add the following redirect URL https:///auth/login



Then, click **Register**. The app's **Overview** page opens.

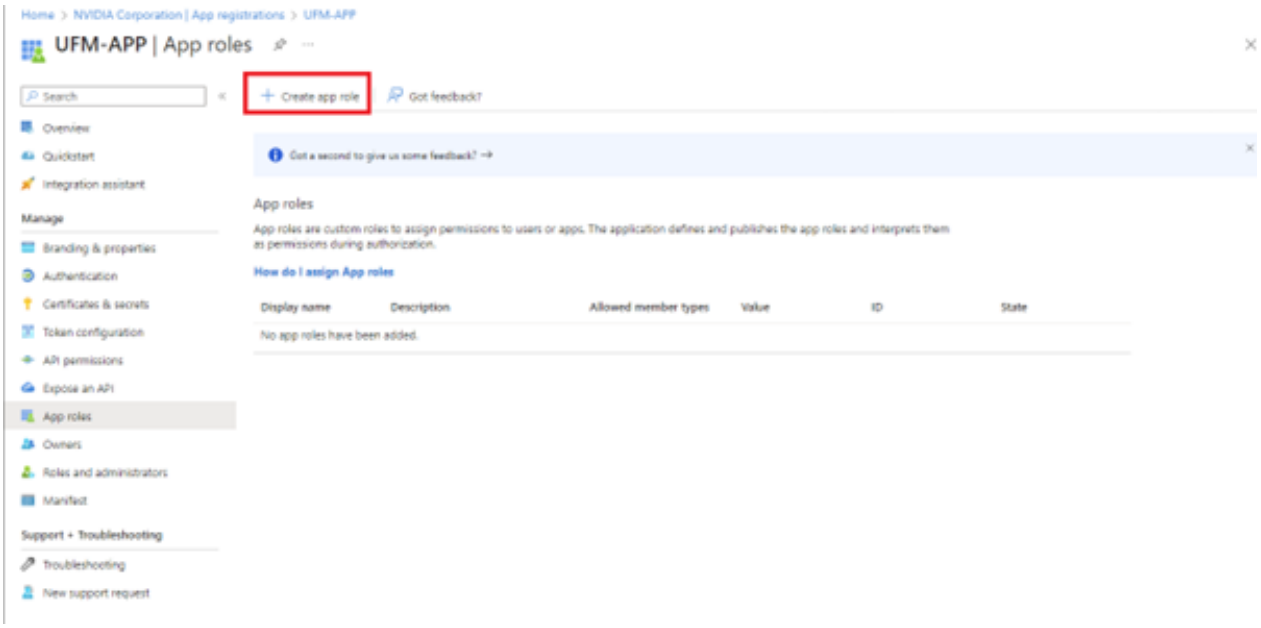
5. Under **Manage** in the side menu, click **Certificates & Secrets** > New client secret.



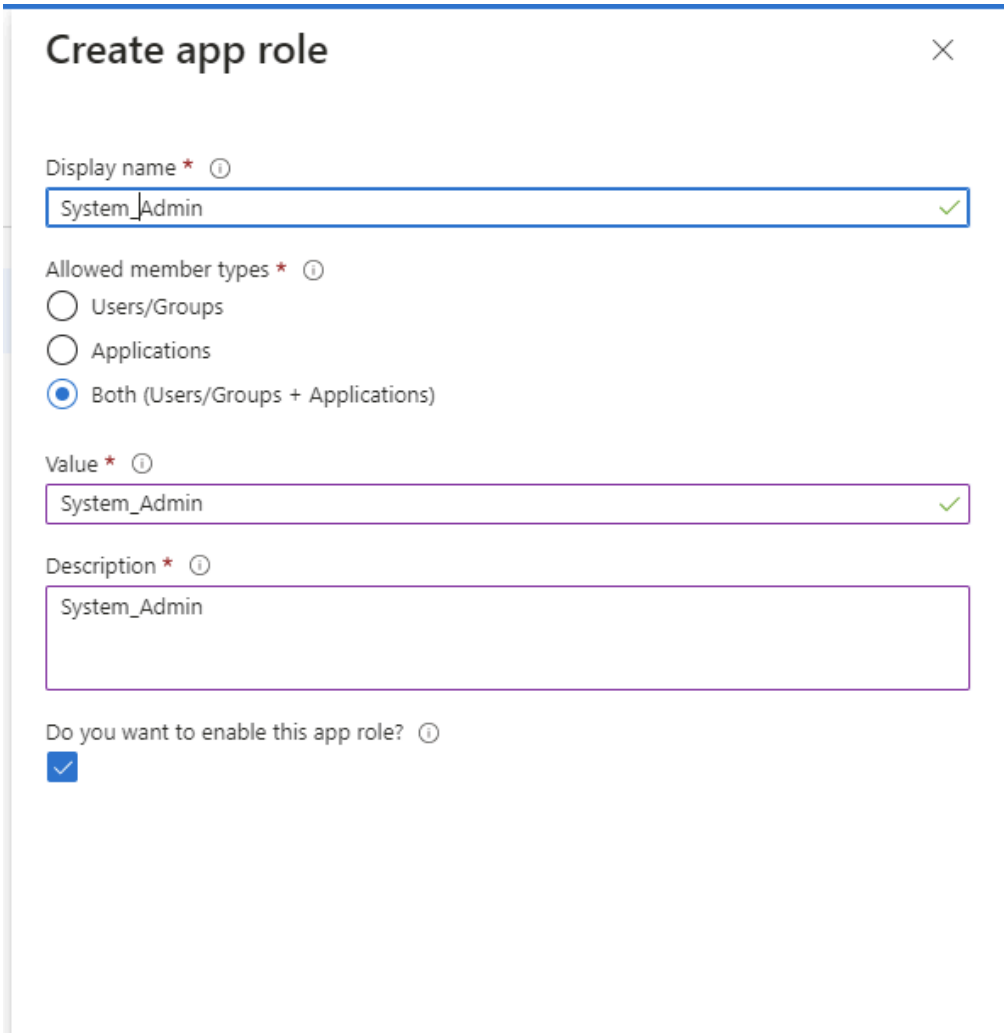
Provide a description for the client secret and set an expiration time, then click "**Add**."

6. Copy the client secret key value which will be needed to configure the UFM with Azure AD (Please note that the value of the generated secret will be hidden and will not be able to be copied/read after you leave the page.

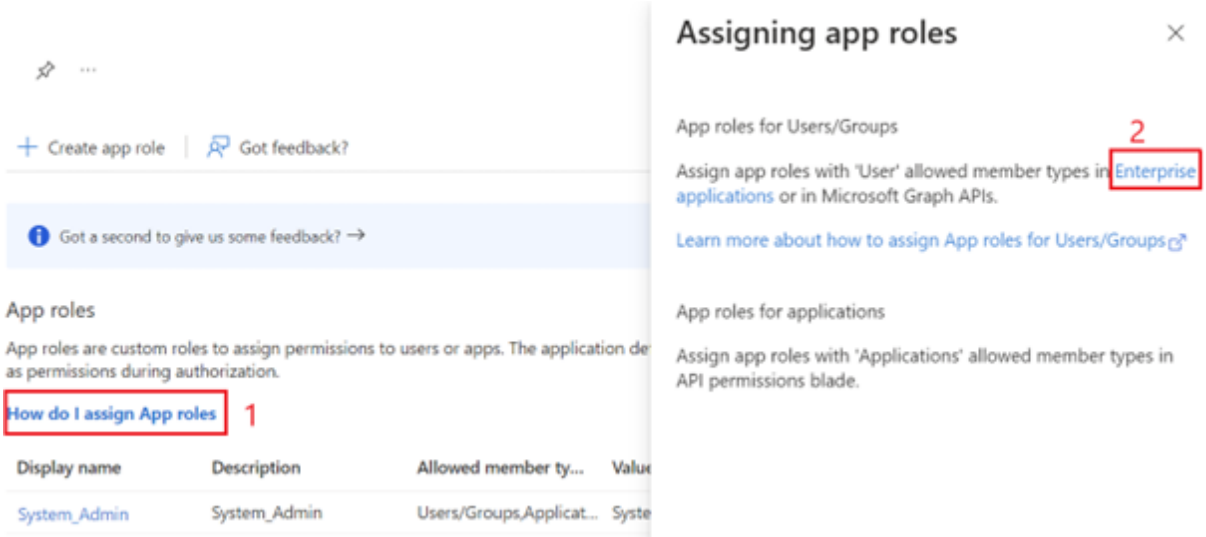
   Under "**Manage"** in the side menu, click App roles > Create app role.

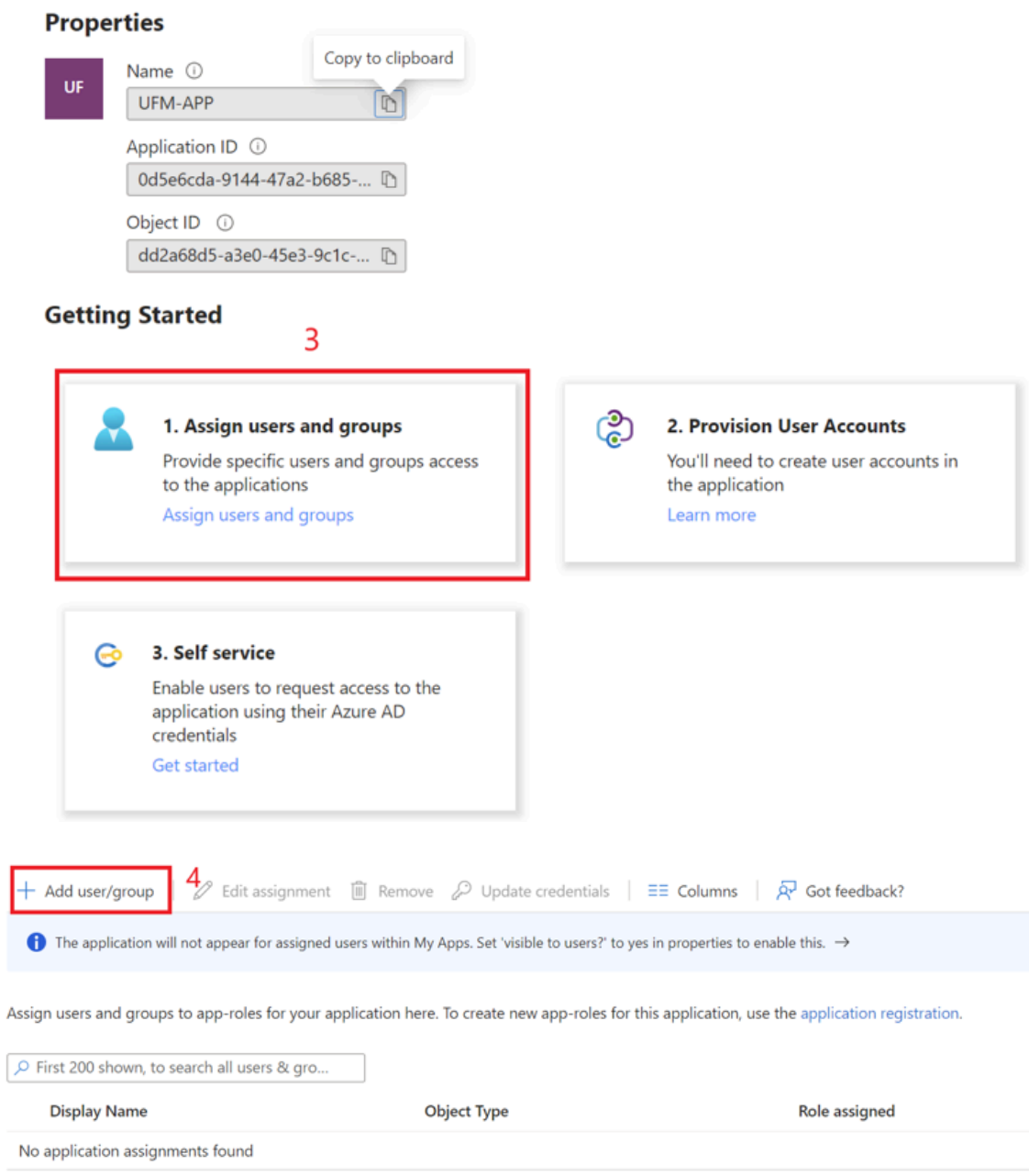


7. Provide the role details. Please note that the role value must be a valid UFM role; otherwise, the login will fail.

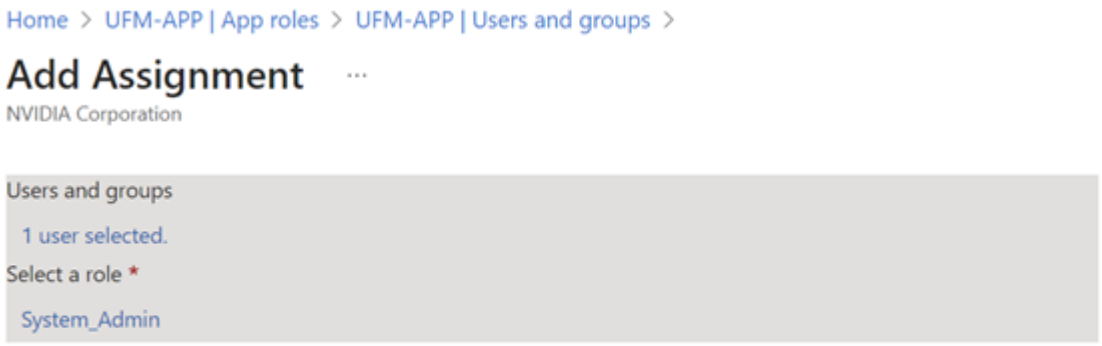


8. Assign the created role to the user. Follow the below steps:




Properties
UF
Name
UFM-APP
Copy to clipboard
Application ID
0d5e6cda-9144-47a2-b685-...
Object ID
dd2a68d5-a3e0-45e3-9c1c-...
Getting Started
3
1. Assign users and groups
Provide specific users and groups access to the applications
Assign users and groups
2. Provision User Accounts
You'll need to create user accounts in the application
Learn more
3. Self service
Enable users to request access to the application using their Azure AD credentials
Get started

Add user/group
4
Edit assignment
Remove
Update credentials
Columns
Got feedback?
The application will not appear for assigned users within My Apps. Set 'visible to users?' to yes in properties to enable this.
Assign users and groups to app-roles for your application here. To create new app-roles for this application, use the application registration.
First 200 shown, to search all users & gro...
Display Name
Object Type
Role assigned
No application assignments found



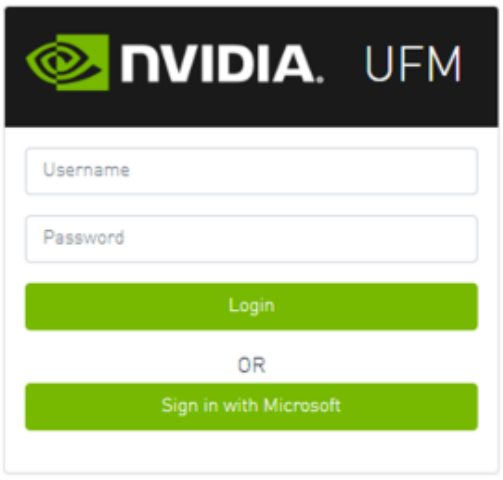
9. Click on "**Overview**" in the side menu to view the application information, such as tenant ID, client ID, and other details.

Enable Azure Authentication From UFM

Azure authentication is disabled by default. To enable it, please refer to Enabling Azure AD Authentication.

Azure Authentication Login Page

After enabling and configuring Azure AD authentication, an additional button will appear on the primary UFM login page labeled 'Sign In with Microsoft,' which will leads to the main Microsoft sign-in page:



Kerberos Authentication

Kerberos is a network authentication protocol designed to provide strong authentication for client-server applications by using secret-key cryptography.

The Kerberos protocol works on the basis of tickets, it helps ensure that communication between various entities in a network is secure. It uses symmetric-key cryptography, which means both the client and servers share secret keys for encrypting and decrypting communication.

To enable Kerberos Authentication, refer to Enabling Kerberos Authentication.

Setting Up Kerberos Server Machine

To set up a system as a Kerberos server, perform the following:

1. Install the required packages:

```
#Redhat
sudo yum install krb5-libs krb5-server
# Ubuntu
sudo apt-get install krb5-kdc krb5-admin-server
```

2. Edit the Kerberos configuration file '`/etc/krb5.conf`' to reflect your realm, domain and other settings:

```
[libdefaults]
    default_realm = YOUR-REALM

[realms]
    YOUR-REALM = {
        kdc = your-kdc-server
        admin_server = your-admin-server
    }

[domain_realm]
    your-domain = YOUR-REALM
    your-domain = YOUR-REALM
```

3. Use the `kdb5_util` command to create the Kerberos database:

```
kdb5_util create -r YOUR-REALM -s
```

4. Add administrative principals:

```
 Kadmin.local addprinc -randkey HTTP/YOUR-HOST-NAME@YOUR-
REALM
```

5. Start KDC and Kadmin services:

```
sudo systemctl start krb5kdc kadmin
sudo systemctl enable krb5kdc kadmin
```

6. Generate a keytab file. The keytab file contains the secret key for a principal and is used to authenticate the service.

```
 kadmin.local ktadd -k /path/to/your-keytab-file HTTP/YOUR-
HOST-NAME@YOUR-REALM
```

Replace `/path/to/your-keytab-file` with the actual path where you want to store the keytab file.

Setting Up Kerberos Client Machine

Follow the below steps to set up a system as a Kerberos client.

1. Install the required packages. When installing the UFM, the following packages will be installed as dependencies:

```
#Redhat
krb5-libs krb5-workstation mod_auth_gssapi
# Ubuntu
krb5-config krb5-user libapache2-mod-auth-gssapi
```

2. Configure the /etc/krb5.conf file to reflect your realm, domain, local names map and other settings:

```
[libdefaults]
    default_realm = YOUR-REALM

[realms]
    YOUR-REALM = {
        kdc = your-kdc-server
        admin_server = your-admin-server
        auth_to_local_names = {
            your-principle-name = your-local-user
        }

    }

[domain_realm]
    your-domain = YOUR-REALM
    your-domain = YOUR-REALM
```

3. Copy the keytab file from the Kerberos server to the machine where your service runs (the client). It is important to ensure that it is kept confidential.

   Please ensure that the keytab file exists and that Apache has the necessary read permissions to access the keytab file; otherwise, Kerberos authentication will not function properly.

4. Obtain a Kerberos ticket-granting ticket (TGT):

```
kinit -k -t /path/to/your-keytab-file HTTP/YOUR-HOST-
NAME@YOUR-REALM
```

5. Enable Kerberos Authentication from UFM. Kerberos authentication is disabled by default. To enable it, please refer to Enabling Kerberos Authentication.

6. Test the Kerberos Authentication. You can use curl to test whether the user can authenticate to UFM REST APIs using Kerberos.

```
curl --negotiate -i -u : -k 'https://ufmc-eos01/ufmRestKrb/app/tokens'
```

Licensing

UFM license is subscription-based featuring the following subscription options:

- 1-year subscription
- 3-year subscription
- 5-year subscription
- Evaluation 30-day trial license

> **Note**
>
> UFM will continue to support old license types, but they are no longer available to obtain.

2 months before the expiration of your subscription license, UFM will warn you that your license will expire soon. After the subscription expires, UFM will continue to work with the expired license for two months beyond its expiration.

During this extra two-month period, UFM will generate a critical alarm indicating that the UFM license has expired and that you need to renew your subscription. Failing to do so within that 2-month period activates UFM Limited Mode. Limited mode blocks all REST APIs and access to the UFM web UI.

UFM enables functionality based on the license that was purchased and installed. This license determines the functionality and the maximum allowed number of nodes in the fabric.

To renew your UFM subscription, purchase a new license and install the new license file by downloading the license file to a temp directory on the UFM master server and then

copying the license file to /opt/ufm/files/licenses/ directory.

> **Note**
>
> UFM may not detect new license files if downloaded directly to /opt/ufm/files/licenses. If UFM does not detect the new license file, a UFM restart may be required.

If several licenses are installed on the server (more than one license file exists under /opt/ufm/files/licenses/), UFM uses only the strongest license and takes into consideration the expiration date, and the managed device limits on it, regardless of any other licenses that may exist on the server.

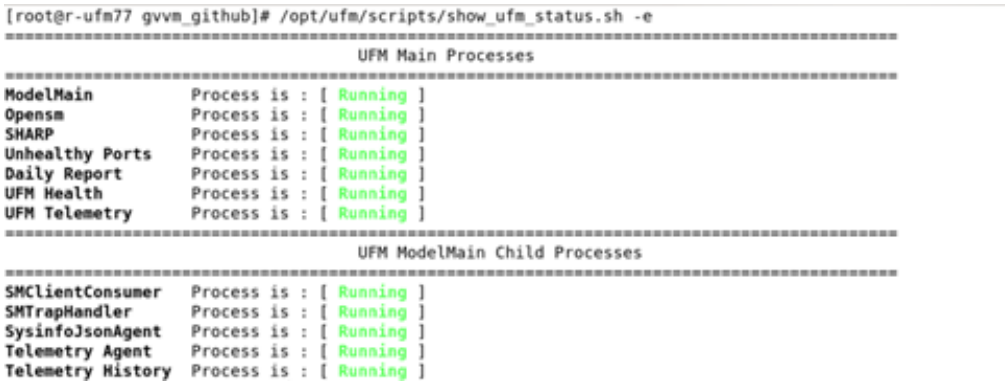
Showing UFM Processes Status

This functionality allows users to view the current status of main processes handled by the UFM.

- To view the main UFM processes, run the script `show_ufm_status.sh` under the `/opt/ufm/scripts` `.` Example: `/opt/ufm/scripts/show_ufm_status.sh`
- To view the UFM main and child processes, run the script `show_ufm_status.sh` with `-e` (extended_processes).

  Example: `/opt/ufm/scripts/show_ufm_status.sh` `-e`

```
[root@r-ufm77 gvvm_github]# /opt/ufm/scripts/show_ufm_status.sh
================================================================================
                              UFM Main Processes
================================================================================
ModelMain           Process is : [ Running ]
Opensm              Process is : [ Running ]
SHARP               Process is : [ Running ]
Unhealthy Ports     Process is : [ Running ]
Daily Report        Process is : [ Running ]
UFM Health          Process is : [ Running ]
UFM Telemetry       Process is : [ Running ]
```



Upgrading UFM Software

After UFM installation, UFM detects existing UFM versions previously installed on the machine and prompts you to run a clean install of the new version or to upgrade. We recommend backing up the UFM configuration before upgrading the UFM as specified in the section UFM Database and Configuration File Backup.

> **Info**
>
> Upgrading the UFM Enterprise software version is supported up to two previous GA software versions (GA -1 or GA -2).
>
> For example, if you wish to upgrade to UFM Enterprise v6.11.0, it is possible to do so only from UFM Enterprise v6.9.0 or v6.10.0.

- Upgrading UFM on Bare Metal Server
- Upgrading UFM on Docker Container

Upgrading UFM on Bare Metal Server

Upgrading UFM on Bare Metal - Standalone Server Upgrade

You can upgrade the UFM standalone server software for InfiniBand from the previous UFM version.

To upgrade the UFM server software:

1. Create a temporary directory (for example */tmp/ufm*).
2. Open the UFM software zip file that you downloaded. The zip file contains the following installation files for:
    - RedHat 7/CentOS 7/OEL 7: ufm-X.X -XXX.el7.x86_64.tgz
    - RedHat 8/CentOS 8/OEL 8: ufm-X.X -XXX.el8.x86_64.tgz
    - Ubuntu 18.04: ufm-X.X -XXX.ubuntu18.x86_64.tgz
    - Ubuntu 20.04: ufm-X.X -XXX.ubuntu20.x86_64.tgz
    - Ubuntu 22.04: ufm-X.X-XXX.ubuntu22.x86_64.tgz
3. Extract the installation file for your system's OS to the temporary directory that you created.
4. Stop the UFM server. Run:

```
systemctl stop ufm-enterprise
```

5. From within the temporary directory, run the following command as root:

```
./upgrade.sh
```

> **Note**
>
> A configuration backup ZIP file will be created in the running directory (e.g. /tmp/ufm). The backup file name is ufm_X.X.X_bkp.zip (X.X.X is the previous version).

1. Upgrade from the previous version: the existing UFM data and configuration are preserved.

2. In case upgrade.sh script stops before completion (e.g. missing prerequisite), the upgrade procedure can be resumed by fixing the issue (e.g. installing missing prerequisite) and rerunning ./upgrade.sh again.

6. Restart the UFM server. Run:

```
systemctl start ufm-enterprise.service
```

> **Note**
>
> /etc/init.d/ufmd start - Available for backward compatibility.

7. After the upgrade, remove the temporary directory

Upgrading UFM on Bare Metal - High Availability Upgrade

> **Note**
>
> As of UFM version 6.14.0, UFM upgrade on HA supports in-service upgrade, meaning UFM can continue running during the steps of the upgrade, and there is no need to stop UFM before the upgrade (although this is also supported).

You can upgrade the UFM server HA software for InfiniBand from the previous release. The upgrade is performed on both servers.

To upgrade the UFM server software:

Upgrading the UFM Enterprise Package

1. On the standby server, extract the new UFM Enterprise package to the /tmp folder:

```
tar -xzf ufm-X.X.X-XXXXX.tgz -C /tmp
```

2. On the standby server, enter to the installation folder and upgrade script:

```
standby# cd /tmp/ ufm-X.X.X-X.<OS_NAME>.x86_64.mofed5/
```

3. Run the UFM upgrade script on the standby server:

```
./upgrade.sh
```

4. After the completion of the upgrade script, the UFM code will undergo an upgrade, while the UFM data will remain unchanged. The automatic upgrade of UFM data will take place during the next startup of UFM. To initiate this process, execute a failover from the Master node (or perform a takeover from the Standby node).

```
master# ufm_ha_cluster failover
```

> **Note**
>
> UFM will log the data upgrade to the syslog of the server, in case of issue a backup of the UFM data is saved prior to the upgrade in /opt/ufm/BACKUP directory and can be restored. For more information, refer to Appendix – Restoring UFM Data.

5. Once UFM is operational on the upgraded node (formerly the standby node), proceed to replicate steps 1 to 3 on the non-upgraded node (previously the master node).

Upgrading the UFM HA Package

1. On **both servers**, download latest UFM-HA package:

   ```
   wget https://www.mellanox.com/downloads/UFM/ufm_ha_5.3.1-2.tgz
   ```

   For Sha256:

   ```
   wget https://download.nvidia.com/ufm/ufm_ha/5.6.0/ufm_ha_5.3.1-2.sha256
   ```

2. On **both servers**, extract the HA package under `/tmp/` and enter the new directory

3. Stop the UFM HA cluster, run the following command on the **Master server**:

   ```
   ufm_ha_cluster stop
   ```

4. On the UFM **Standby server**, run the upgrade command from within the extracted HA package located in `/tmp`:

   ```
   ./install.sh --upgrade
   ```

5. On the UFM **Master server,** run the upgrade command from within the extracted HA package located in `/tmp`:

   ```
   ./install.sh --upgrade
   ```

6. Start the UFM HA cluster, run the following command on the **Master server**:

```
ufm_ha_cluster start
```

7. Run the following command to verify that the UFM HA cluster is up and running:

```
ufm_ha_cluster status
```

Upgrading UFM on Docker Container

> **Note**
>
> Upgrade the UFM container based on the existing UFM configuration files that are mounted on the server. It is important to use that same directory as a volume for the UFM installation command.
>
> In the below example /opt/ufm_files is used.

Upgrading UFM on Docker Container in Standalone Mode

1. Stop the UFM Enterprise service. Run:

```
systemctl stop ufm-enterprise
```

2. Remove the existing docker image. Run:

```
docker rmi mellanox/ufm-enterprise:latest
```

3. Load the new UFM Enterprise docker image. Run:

```
docker pull mellanox/ufm-enterprise:latest
```

4. Run the docker upgrade command:

```
docker run -it --name=ufm_installer --rm \
-v /var/run/docker.sock:/var/run/docker.sock \
-v /etc/systemd/system/:/etc/systemd_files/ \
-v /opt/ufm/files/:/opt/ufm/shared_config_files/ \
mellanox/ufm-enterprise:latest --upgrade
```

5. Reload system manager configuration:

```
systemctl daemon-reload
```

6. Start UFM Enterprise service:

```
systemctl start ufm-enterprise
```

Upgrading UFM Container in High Availability Mode

> **Note**
>
> As of UFM version 6.14.0, UFM upgrade on HA supports in-service upgrade, meaning UFM can continue running during the steps of the upgrade, and there is no need to stop UFM before the upgrade (although this is also supported).

Upgrading the UFM Enterprise Package

1. Remove the old docker image from the **Standby** server. Run:

```
Stand-by# docker rmi mellanox/ufm-enterprise:latest
```

2. Pull the new UFM Enterprise docker image on the **Standby** server. Run:

```
docker pull mellanox/ufm-enterprise:latest
```

> **Note**
>
> At this stage, the UFM container has been updated with the latest code. The UFM data, however, will be updated during the next UFM run.

3. Perform a failover to start UFM on the upgraded node. On the **Master** node, run:

```
ufm_ha_cluster failover
```

> **Note**
>
> When UFM starts, it will automatically update the UFM configuration.

4. Repeat steps 1-2 on the un-upgraded node (previous **Master** node).

Upgrading the UFM HA Package

1. On **both servers**, download and extract the latest UFM HA package. Run:

```
wget https://www.mellanox.com/downloads/UFM/ufm_ha_5.6.0-4.tgz
```

For Sha256:

```
wget https://download.nvidia.com/ufm/ufm_ha/5.6.0/ufm_ha_5.6.0-4.sha256
```

2. Stop the UFM HA cluster, run the following command on the **Master** server:

```
ufm_ha_cluster stop
```

3. On both the **Master** and **Standby** servers, execute the upgrade command from within the extracted HA package. Ensure you run it first on the **Standby** server, then on the **Master** server:

```
./install.sh --upgrade
```

4. Start the UFM HA cluster by running this command on the **Master** server:

```
ufm_ha_cluster start
```

5. Run the following command to verify that the UFM HA cluster is up and running:

```
ufm_ha_cluster status
```

Uninstalling UFM

The UFM Server can be uninstalled by running an uninstall script in the different server modes:

- Uninstalling UFM in Standalone Mode
- Uninstalling UFM in High Availability
- Uninstalling UFM in Docker Deployment

Uninstalling UFM in Standalone Mode

To uninstall the UFM Server:

1. Go to /opt/ufm.
2. Run ./uninstall.sh.

> **Note**
>
> Child interfaces are not deleted.

3. To delete primary interfaces, restart /etc/init.d/openibd.

Uninstalling UFM in High Availability

To uninstall the UFM Server in high availability mode:

1. Run the following on the master and slave to clean up the UFM HA configuration:

```
ufm_ha_cluser cleanup
```

2. To uninstall the UFM HA configuration, run:

```
/opt/ufm/ufm_ha/uninstall_ha.sh
```

3. To uninstall UFM Enterprise software, run the following on the master and slave:

```
/opt/ufm/uninstall.sh
```

Uninstalling UFM in Docker Deployment

To uninstall the UFM Server in high availability mode:

1. Run the following on the master and slave:

```
ufm_ha_cluser cleanup
```

2. Run:

```
/opt/ufm/ufm_ha/uninstall_ha.sh
```

3. Run the following on the master and slave:

```
/opt/ufm/files/uninstall.sh
```

UFM Configuration Backup and Restore

Overview

UFM migration enables backup and restores UFM configuration files.

Backup UFM configuration

By default, the following folders (placed in `/opt/ufm/files`) are being backed up:

- conf
- dashboardViews
- licenses
- networkViews
- scripts
- sqlite
- templates/user-defined
- ufmhealth/scripts
- userdata
- users_preferences

Note

The user may also backup the UFM historical telemetry data ("-t" argument).

UFM (Bare Metal)

```
/opt/ufm/scripts/ufm_backup.sh --help
usage: ufm_backup.pyc [-h] [-f BACKUP_FILE] [-t]
```

Optional Arguments

-h	--help	show this help message and exit
-f	--backup-file BACKUP_FILE	full path of zip file to be generated
-t	--telemetry	backup UFM historical telemetry

UFM Docker Container

1. Backup UFM configuration. Run:

   ```
   docker exec ufm /opt/ufm/scripts/ufm_backup.sh
   ```

2. Copy the backup file from UFM docker container to the host. Run:

   ```
   docker cp ufm:/root/<backup file> <path on host>
   ```

UFM Appliance

1. Backup UFM configuration. Run:

```
ufm data backup [with-telemetry]
```

2. Upload the backup file to a remote host. Run:

```
ufm data upload <backup file> <upload URL>
```

> **Note**
>
> More details can be found in the log file `/tmp/ufm_backup.log.`

Restore UFM Configuration

> **Note**
>
> All folders which are a part of the UFM backup are restored (filter is done during the backup stage).

UFM Bare Metal

```
/opt/ufm/scripts/ufm_restore.sh --help
usage: ufm_restore.pyc [-h] -f BACKUP_FILE [-u] [-v]
```

Optional Arguments

-h	--help	show this help message and exit
-f BACKUP_FILE	--backup-file BACKUP_FILE	full path of zip file generated by backup script
-u	--upgrade	upgrades the restored UFM files
-v	--verbose	makes the operation more talkative

UFM Docker Container

1. Stop UFM. Run:

```
docker exec ufm /etc/init.d/ufmd stop
```

2. Copy the backup file from the host into UFM docker container. Run:

```
docker cp <backup file> ufm:/tmp/<backup file>
```

3. Restore UFM configuration. Run:

```
docker exec ufm /opt/ufm/scripts/ufm_restore.sh -f
/tmp/<backup file> [--upgrade]
```

4. Start UFM. Run:

```
docker exec ufm /etc/init.d/ufmd start
```

UFM Appliance

1. Stop UFM. Run:

```
no ufm start
```

2. Copy the backup file from a remote host into UFM appliance. Run:

```
ufm data fetch <download URL>
```

3. Restore UFM configuration. Run:

```
ufm data restore <backup file>
```

4. Start UFM. Run:

```
ufm start
```

Note

When restoring the UFM configuration from host to a container, the following parameters in `/opt/ufm/files/conf/gv.cfg` may be reset the following:

- fabric_interface
- ufma_interfaces
- mgmt_interface

Note

UFM configuration upgrade during restore is not supported in UFM Appliance GEN2/GEN2.5

More details can be found in the log files `/tmp/ufm_restore.log` and `/tmp/ufm_restore_upgrade.log`

UFM Factory Reset

This section provides a comprehensive guide on resetting UFM to its original factory settings.

Note

WARNING!!! this operation will remove all user data and configuration and will restore UFM to its factory defaults.

Note

The UFM Factory-Reset will exclusively revert UFM to its original factory settings, leaving HA configurations unaffected. To remove HA, it is essential to execute `ufm_ha_cluster cleanup` before initiating the factory reset.

UFM Docker Container Factory Reset

To reset UFM to its factory defaults when using UFM on a Docker container, follow these steps.

1. Ensure that UFM is not up and running. If UFM is running, stop it.

   For Stand-alone (SA) installations:

   ```
   systemctl stop ufm-enterprise
   # validate that ufm is not running
   systemctl status ufm-enterprise
   ```

   For High-Availability setups (perform the following on the master node only):

```
ufm_ha_cluster stop
# validate that ufm is not running
ufm_ha_cluster status
```

2. Run `mellanox/ufm-enterprise` Docker Container with the following flags:

> **Note**
>
> WARNING: This operation will erase all user data and configurations, resetting UFM to its factory defaults.
>
> CAUTION: This step does not require user confirmation, meaning UFM will be restored to factory defaults immediately once initiated.

```
docker run -it --name=ufm_installer --rm \
                -v /var/run/docker.sock:/var/run/docker.sock
\
                -v /tmp:/tmp \
                -
v /opt/ufm/files/:/opt/ufm/shared_config_files/ \
                mellanox/ufm-enterprise:latest \
                --factory-reset
```

Flag	Type	Description
`--name=ufm_installer`	Mandatory	The container name must be called ufm_installer

Flag	Type	Description
`-v /var/run/docker.sock:/var/run/docker.sock`	Mandatory	The docker socket must be mounted on the docker container.
`-v /tmp:/tmp`	Optional	Logs of the operation can be viewed in `/tmp` on the host in case it is mounted
`-v /opt/ufm/files/:/opt/ufm/shared_config_ufm/`	Mandatory	For the factory reset to persist, it is essential to have the /opt/ufm/files directory mounted from the host.
`mellanox/ufm-enterprise:latest`	Mandatory	The docker image name.
`--factory-reset`	Mandatory	This action will signal the UFM container to initiate the factory reset process.

UFM Enterprise Factory Reset

To restore UFM Enterprise to factory defaults:

1. Ensure that UFM is not up and running. If UFM is running, stop it.

   For Stand-alone (SA) installations:

```
systemctl stop ufm-enterprise
# validate that ufm is not running
systemctl status ufm-enterprise
```

For High-Availability setups (perform the following on the master node only):

```
ufm_ha_cluster stop
# validate that ufm is not running
ufm_ha_cluster status
```

2. Run the `ufm_factory_reset.sh` script:

> **Note**
>
> WARNING: This operation will erase all user data and configurations, resetting UFM to its factory defaults.

```
/opt/ufm/scripts/ufm_factory_reset.sh [-y]
```

Flag:

Flag	Type	Description
`-y`	Optional	Does not require user confirmation.

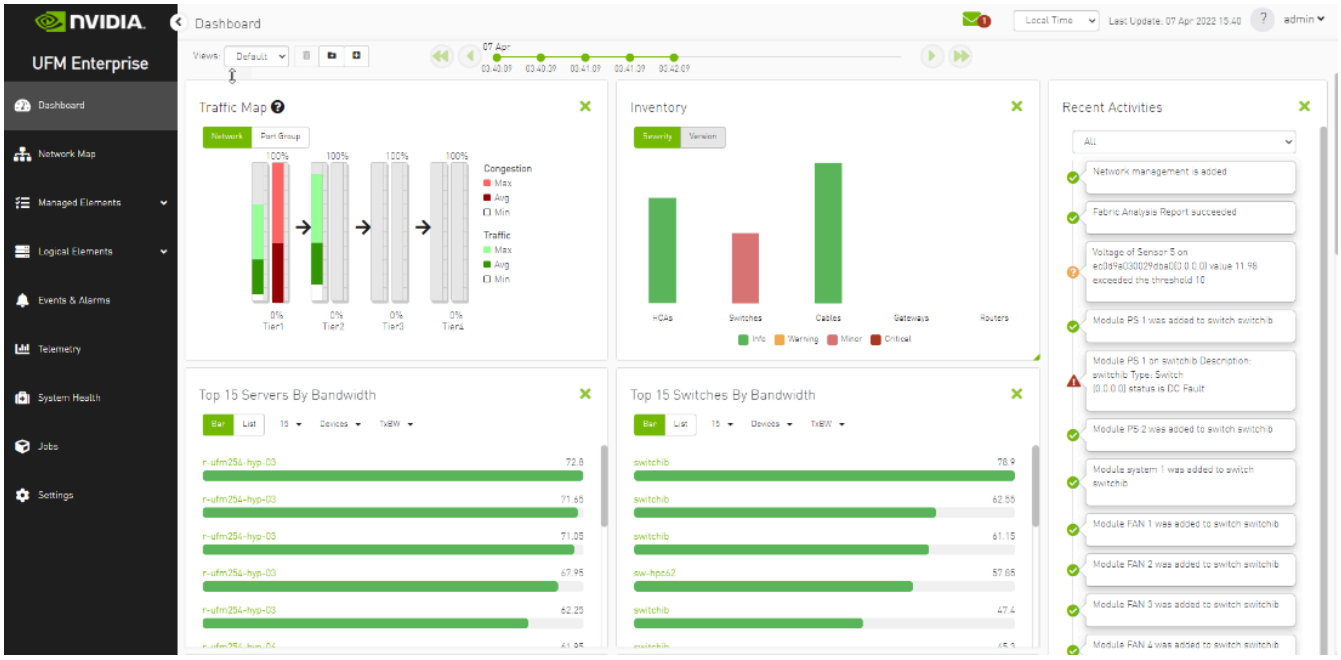
UFM Web UI

This section is constituted by the following sub-sections:

Fabric Dashboard

The dashboard window summarizes the fabric's status, including events, alarms, errors, traffic and statistics.

Fabric Dashboard View

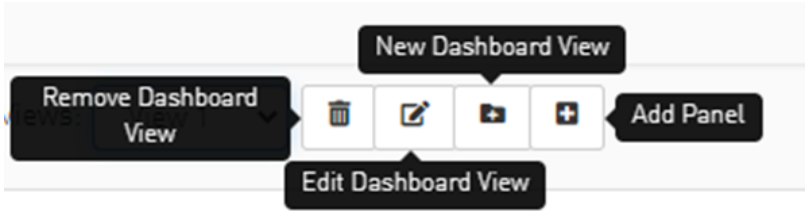


The Fabric Dashboard view consists of the following six dashboards, which provide real-time information about the fabric.

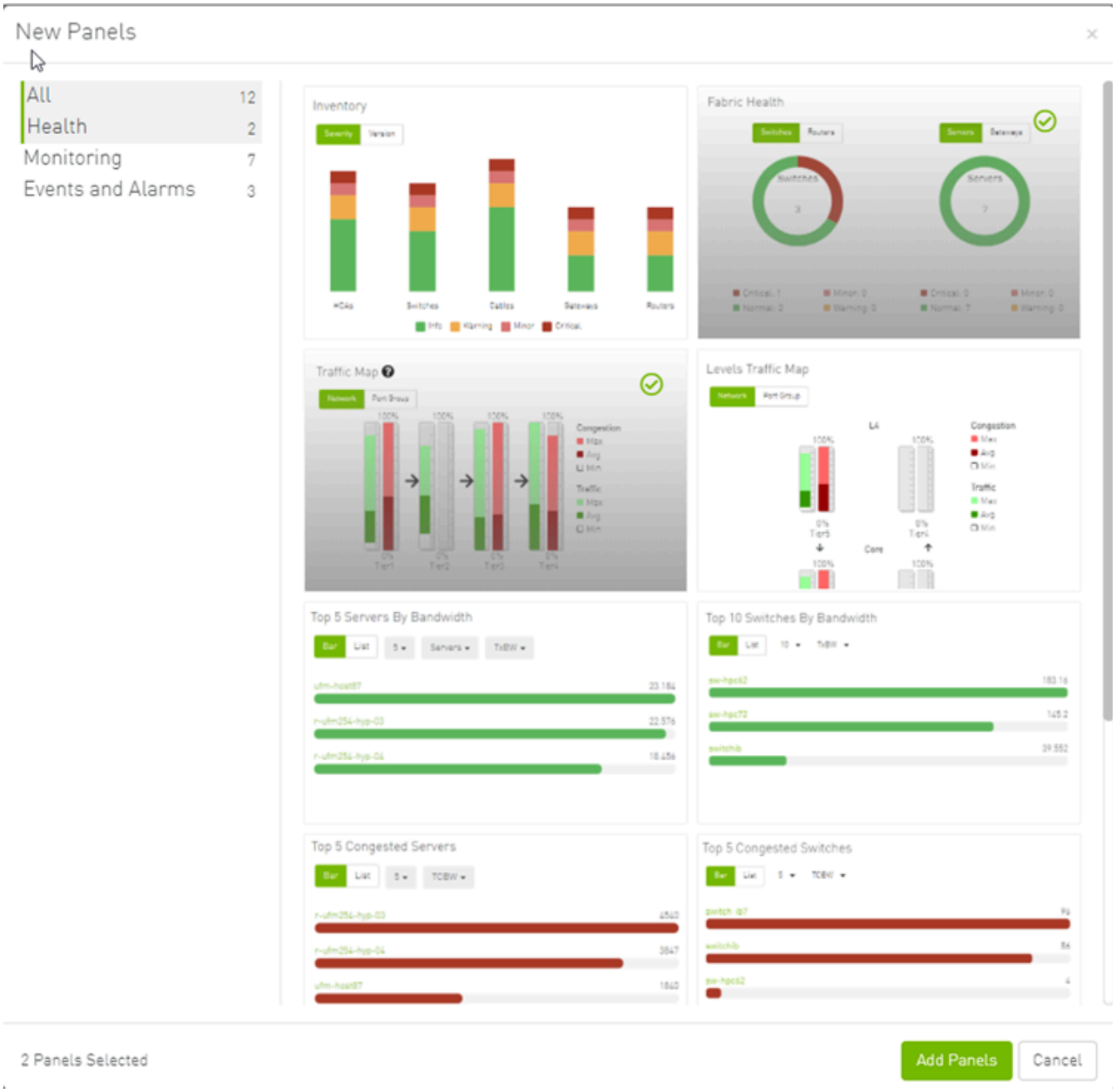
Dashboard Views and Panel Management

UFM is installed with a default view of the most important panels. These panels are resizable and draggable. Users can customize their default view or create new views altogether

The dashboard views and panels are managed by a set of action buttons appearing at the top of the main dashboard screen:



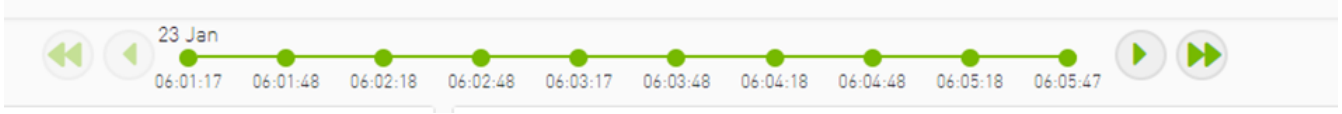
Clicking on the Add Panel button will show a model to select which panels you wish to add to the current dashboard view.



Dashboard Timeline Snapshots

Once the user is logged into the UFM Enterprise, the UFM will start recording snapshots of the dashboard panel data every 30 seconds.

The user is able to navigate between these snapshots and load the dashboard data of a specific data snapshot.



Dashboard Panels

The Fabric Dashboard view consists of the following 12 panels, which are categorized into 3 main categories and provide real-time information about the fabric.

- Health:
    - Inventory
    - Fabric Health
- Monitoring:
    - Traffic Map
    - Levels Traffic Map
    - Top X Servers by bandwidth
    - Top X Switches by bandwidth
    - Top X congested servers
    - Top X congested switches
    - Top X utilized Pkeys
- Events and Alarms:
    - Recent Activities
    - Top X alarmed servers
    - Top X alarmed switches
    - Events History

Top N Servers/Switches by Rx or Tx Bandwidth

The Top N servers/switches by Rx or Tx Bandwidth component shows the top elements that are transmitting or receiving the most bandwidth per second. These elements are classified top-down according the defined Transmit (Tx) or Receive (Rx) bandwidth (MB/sec Rate).

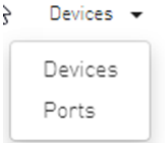
Bandwidth is measured as a rate in bytes/sec.

- Transmitted (Tx) bandwidth is measured by N server/switch ports in MB/sec
- Received (Rx) bandwidth is measured by N server/switch ports in MB/sec

> **Note**
>
> N can be 5, 10, 15, or 20.

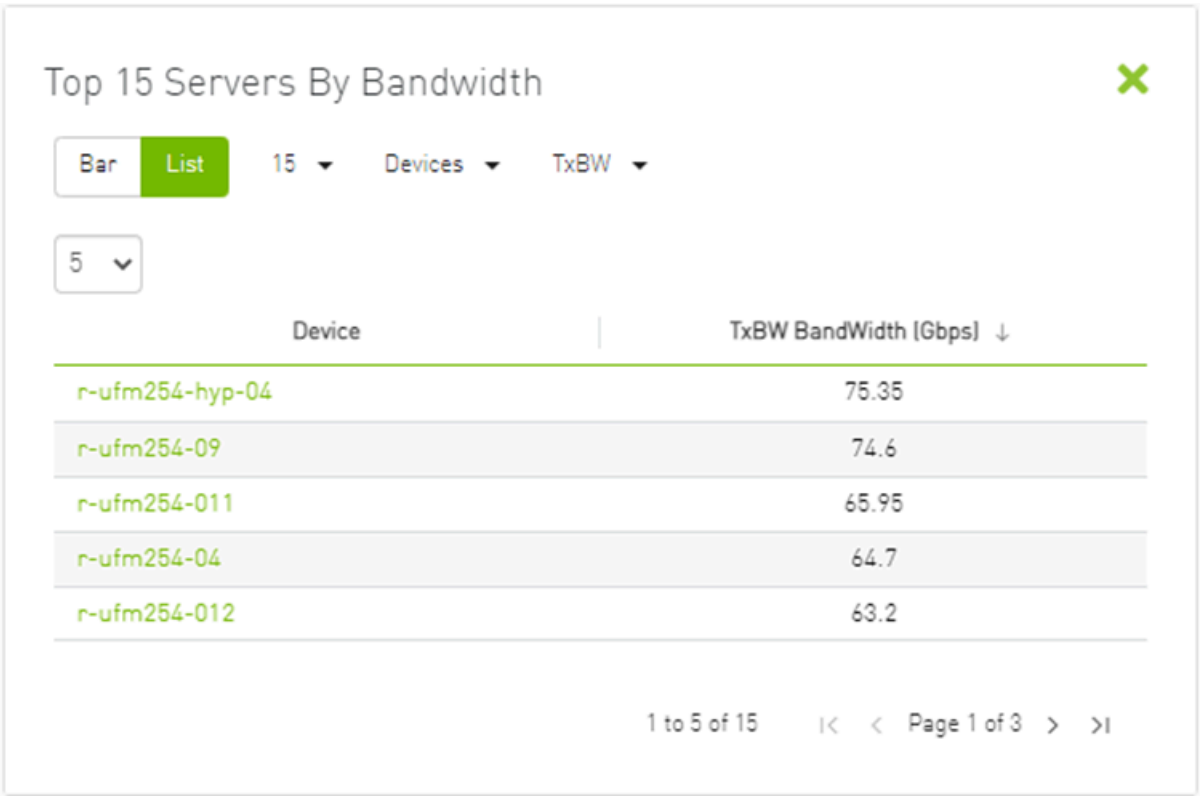
The following table lists the icons of this component:

Options	Description
List view 	Shows the top N elements as a list Each element is shown in a row with the name of the element and the bandwidth rate
Bar view 	Shows the top N nodes as a bar graph • X axis shows the rate as a value • Y axis shows the Node (server) name
Drop-down menu 	Selects the number of items to display Default: 10 nodes
Monitoring attributes 	Selects the attribute for monitoring: • TxBW – Transmit Bandwidth • RxBW – Receive Bandwidth

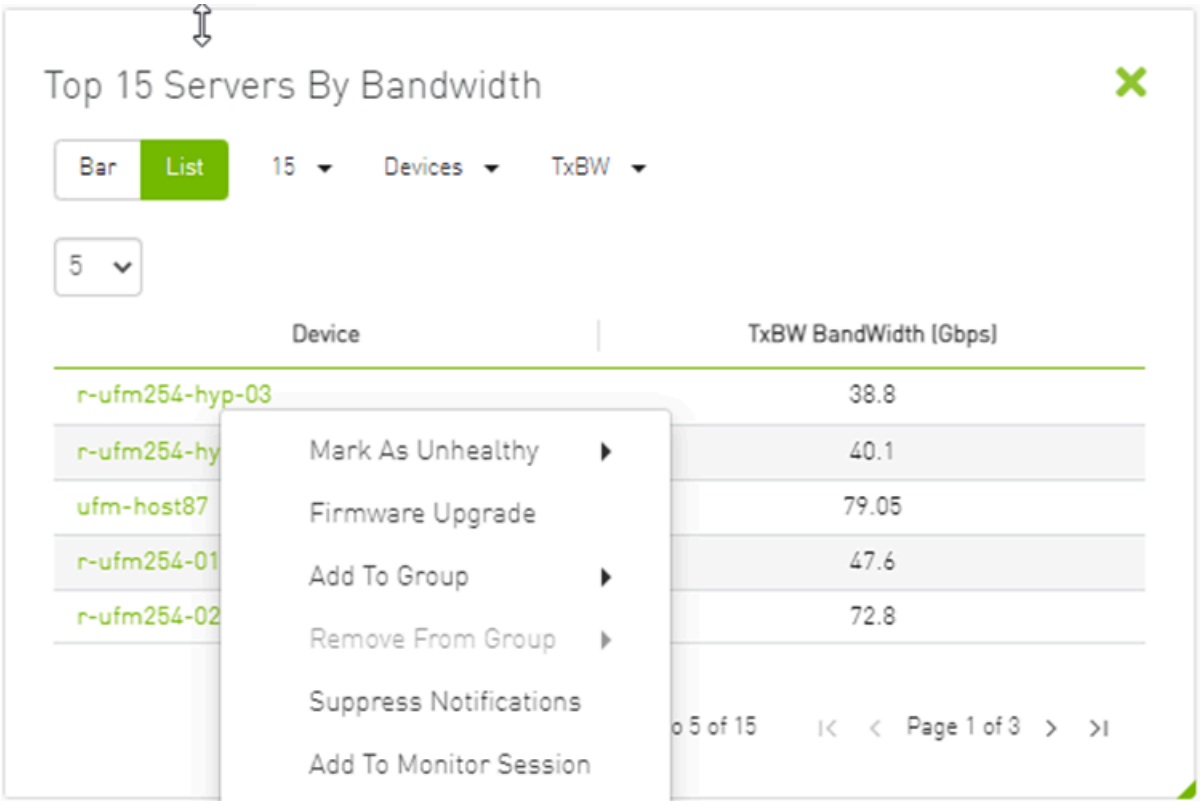
Options	Description
View by port/element 	Switches view to t op 5 elements by bandwidth or top 5 ports by bandwidth. Nodes view is presented by default. • Clicking a specific port in the ports view under the port column redirects to the ports table and highlights that particular port • Clicking a specific device in the devices view under the device column redirects to the Devices table and highlights that particular node
Filter toggle	Toggles the filter textbox

Top Servers/Switches by Bandwidth—Bar View

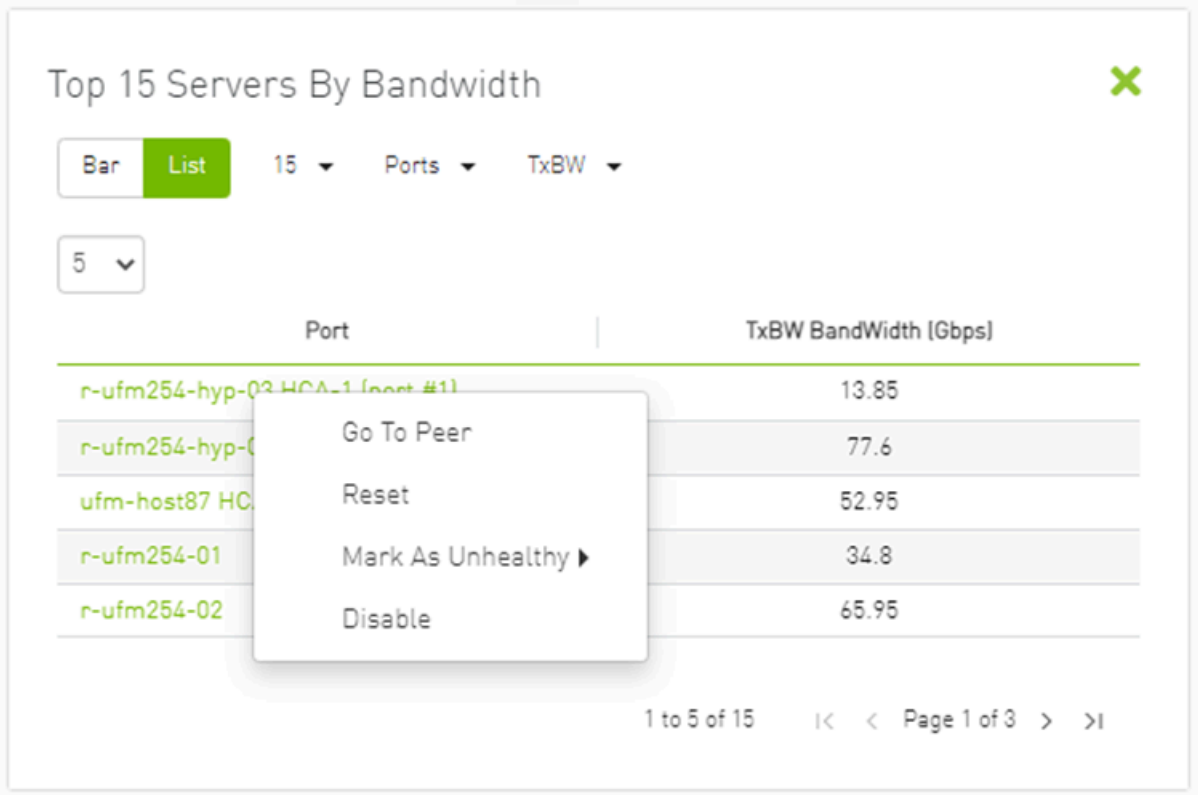
Top Servers/Switches by Bandwidth—List View



Right-clicking a device displays a list of the actions that can be performed. These actions (shown in the following screenshot) are the same actions available in the devices table (see Devices Actions table under Devices Window).



Right-clicking a port displays a list of the actions that can be performed. These actions (shown in the following screenshot) are the same actions available in the Ports table (see Ports Window for more information).



Top N Congested Servers/Switches by Rx/Tx Bandwidth

The Top N Congested devices by Rx or Tx Bandwidth component shows the top congested devices, classified top-down according to the defined Transmit (Tx) or Receive (Rx) bandwidth.

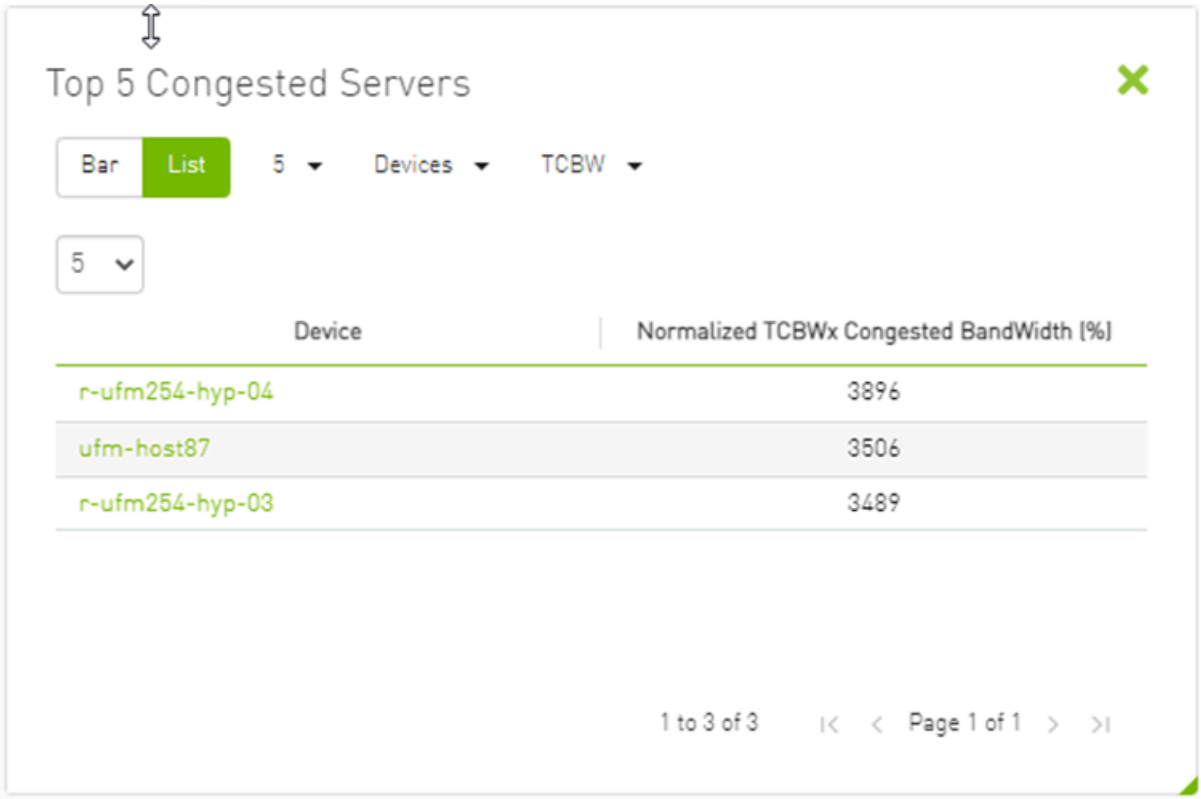
Bandwidth is measured as congestion bandwidth rate (CBW) by percentage.

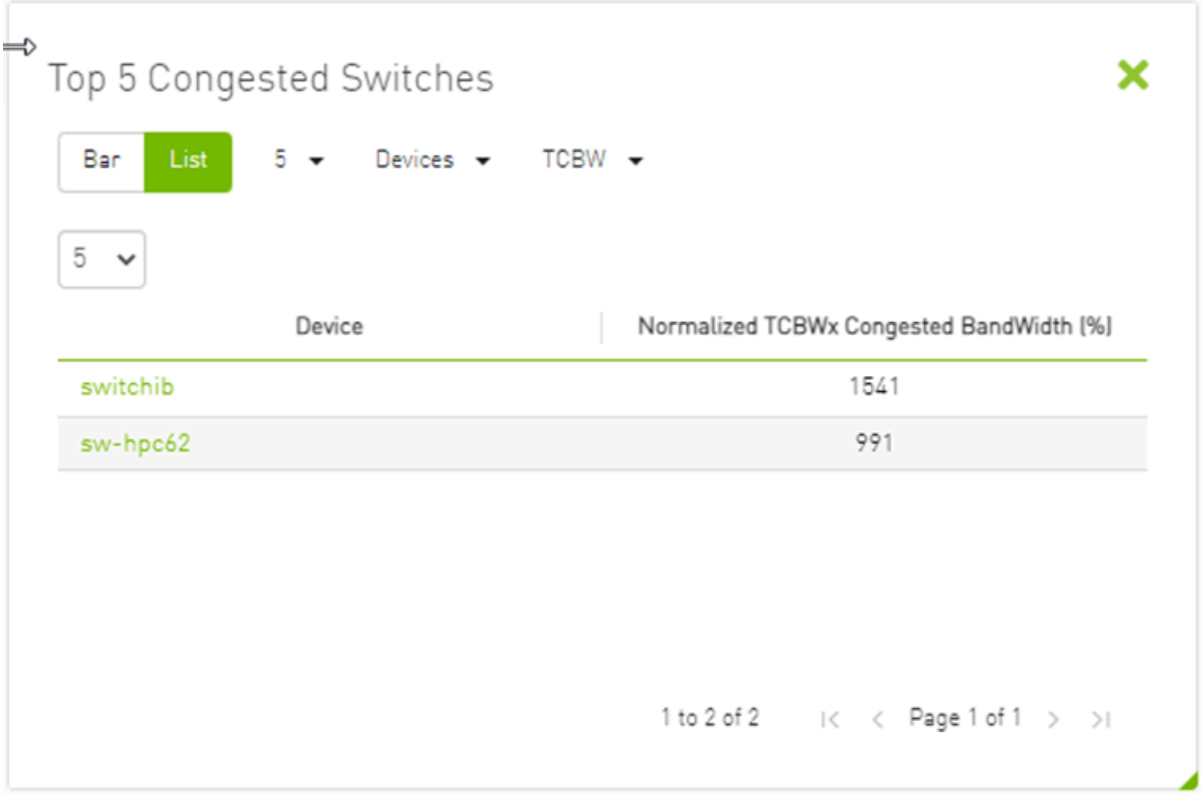
- For Tx, congestion is measured by N HCA ports.
- For Rx, congestion is measured by N switch ports connected to HCAs.

Note

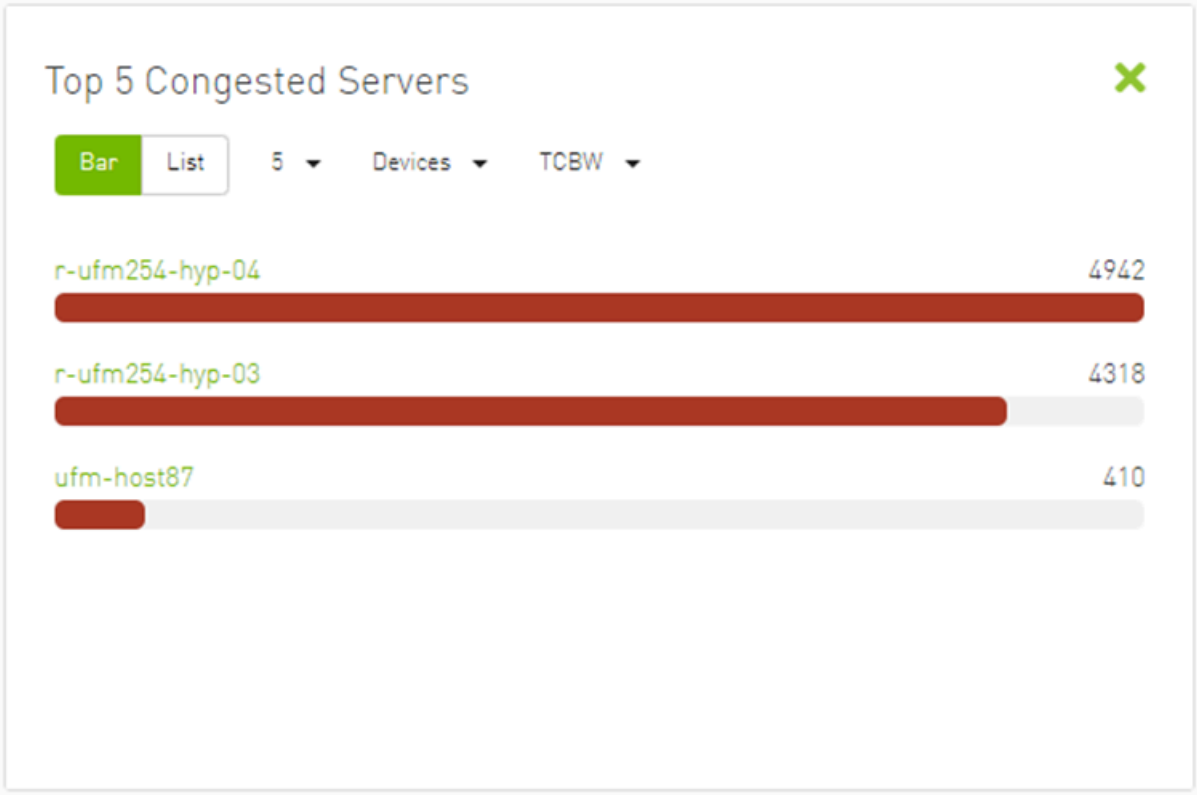
N can be 5, 10, 15, or 20.

Top N Congested Servers by Bandwidth—List View





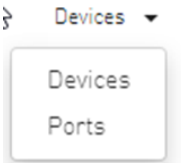
Top N Congested Servers/Switches by Bandwidth—Bar View



The following table describes the options available in this component.

Top N Congested Devices by Rx/Tx Bandwidth

Options	Description
Bar view Bar List	Shows the top N congested devices as a bar graph • X axis shows the rate as a percentage • Y axis shows the congested Node (server) name
List view Bar List	Shows the top N congested nodes as a list Each congested node is shown in a row with the name of the node and its picture. It also shows the bandwidth rate
Drop-down menu 5 5 10 15 20	Enables selecting the number of top N congested nodes Default: 10 nodes

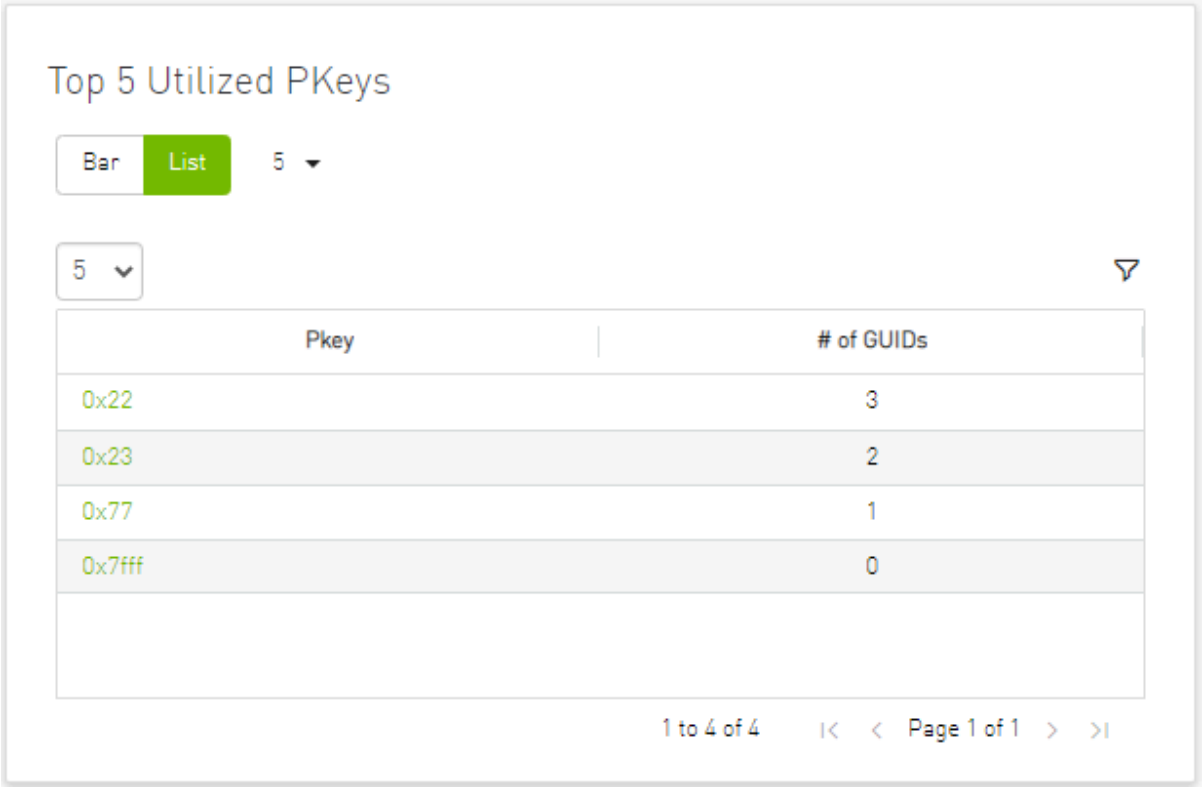
Options	Description
View by port/element 	Switches view to Top 5 elements By Bandwidth or Top 5 Ports By Bandwidth. Devices view is presented by default. • Clicking a specific port in the Ports view under the Port column redirects to the Ports table and highlights that particular port • Clicking a specific device in the Nodes view under the Device column redirects to the Devices table and highlights that particular node
Monitoring attributes 	• RCBW – Receive Congested Bandwidth (percentage) • TCBW – Transmit Congested Bandwidth (percentage)

Top N Utilized PKeys

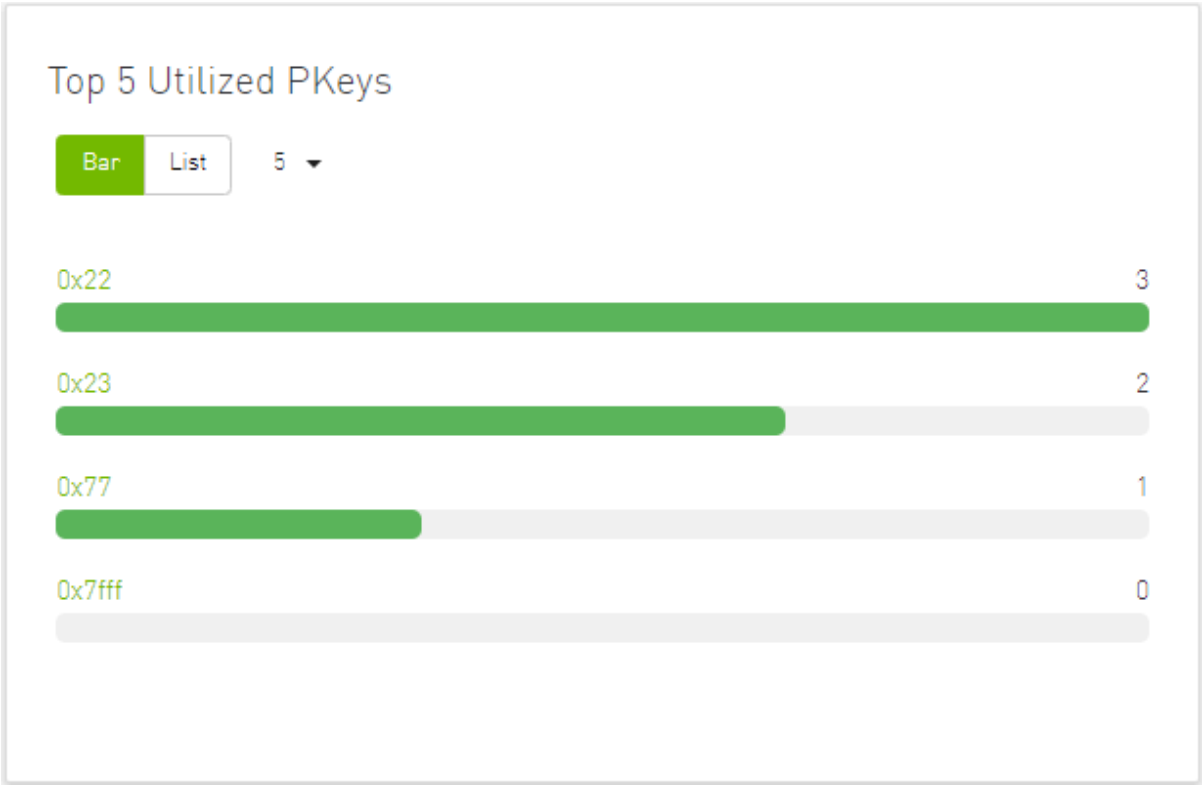
Top N Utilized PKeys displays the top utilized PKeys based on the number of the PKey members.

> **Note**
>
> N can be 5, 10, 15, or 20.

Top N Utilized PKeys—List View



Top N Utilized PKeys—Bar View



The following table describes the options available in this component.

Top N Utilized PKeys

Options	Description
Bar view 	Shows the top N ***Utilized Pkeys*** as a bar graph • X axis shows the number of members • Y axis shows the names of the PKeys
List view 	Shows the top N ***Utilized PKeys*** as a list Each PKey is shown in a row with the name of the PKey and the number of its members
Drop-down menu 	Enables selecting the number of top N ***Utilized PKeys*** Default: 10 ***Utilized PKeys***

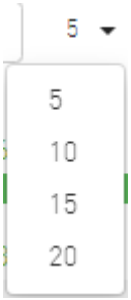
Top N Alarmed Servers/Switches

The Top N Alarmed Servers/Switches component shows the top nodes with alarms classified in a descending order. Alarmed nodes are measured according to the following:

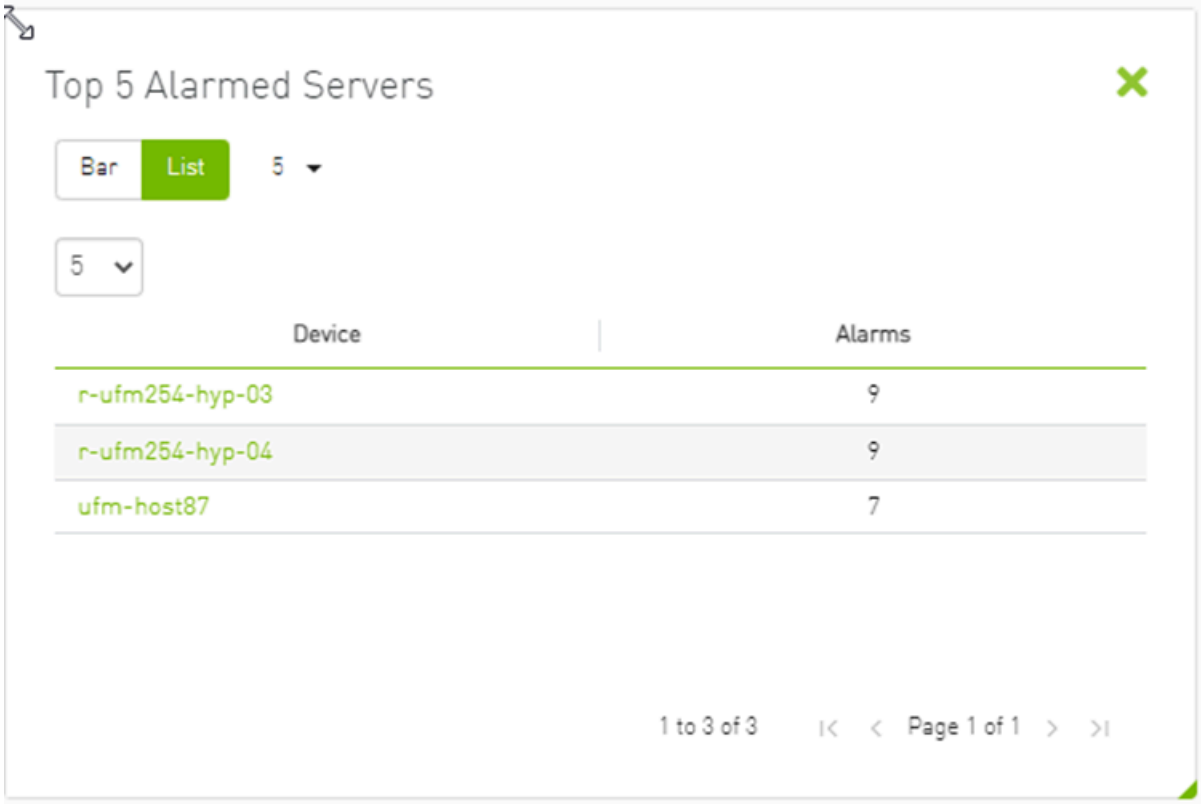
- Severity – only the top nodes, in order of severity:
    - Critical
    - Minor
    - Warning
    - Normal
- Alarm – numbers (N can be 5, 10, 15, or 20)

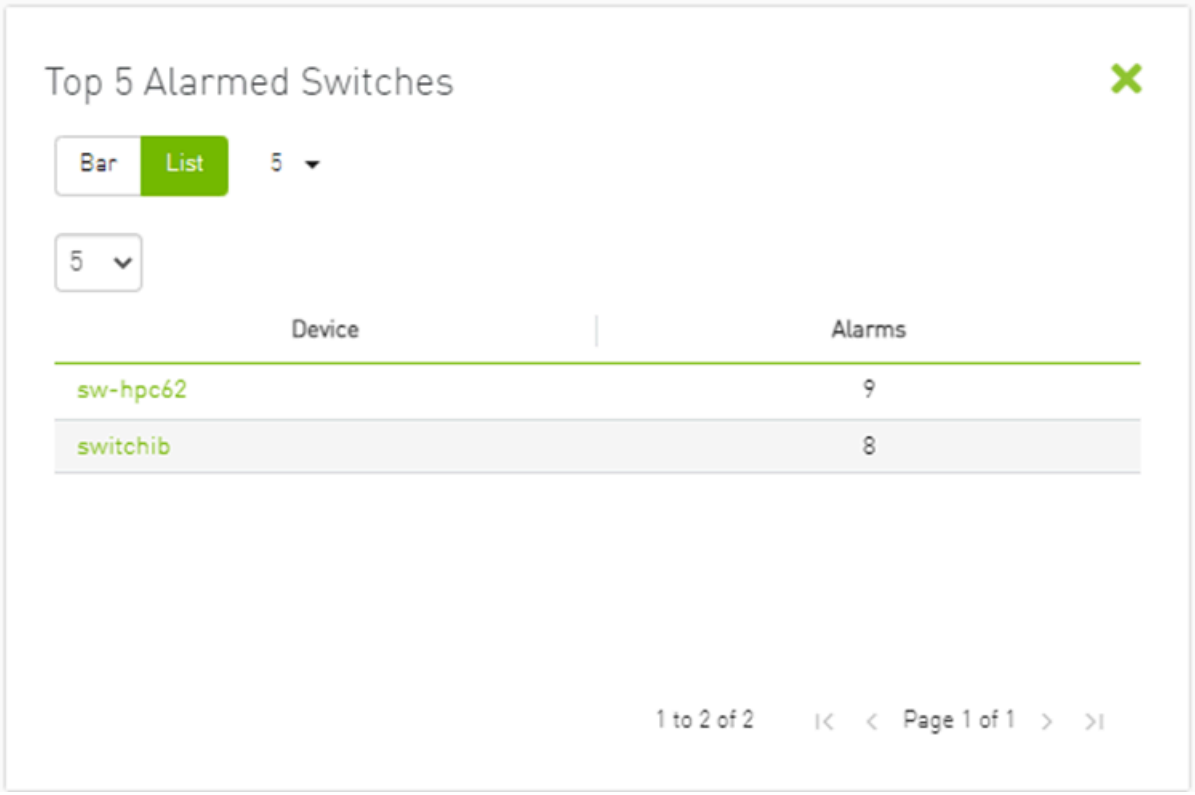
The following table lists the components.

Top N Alarmed Servers/Switches

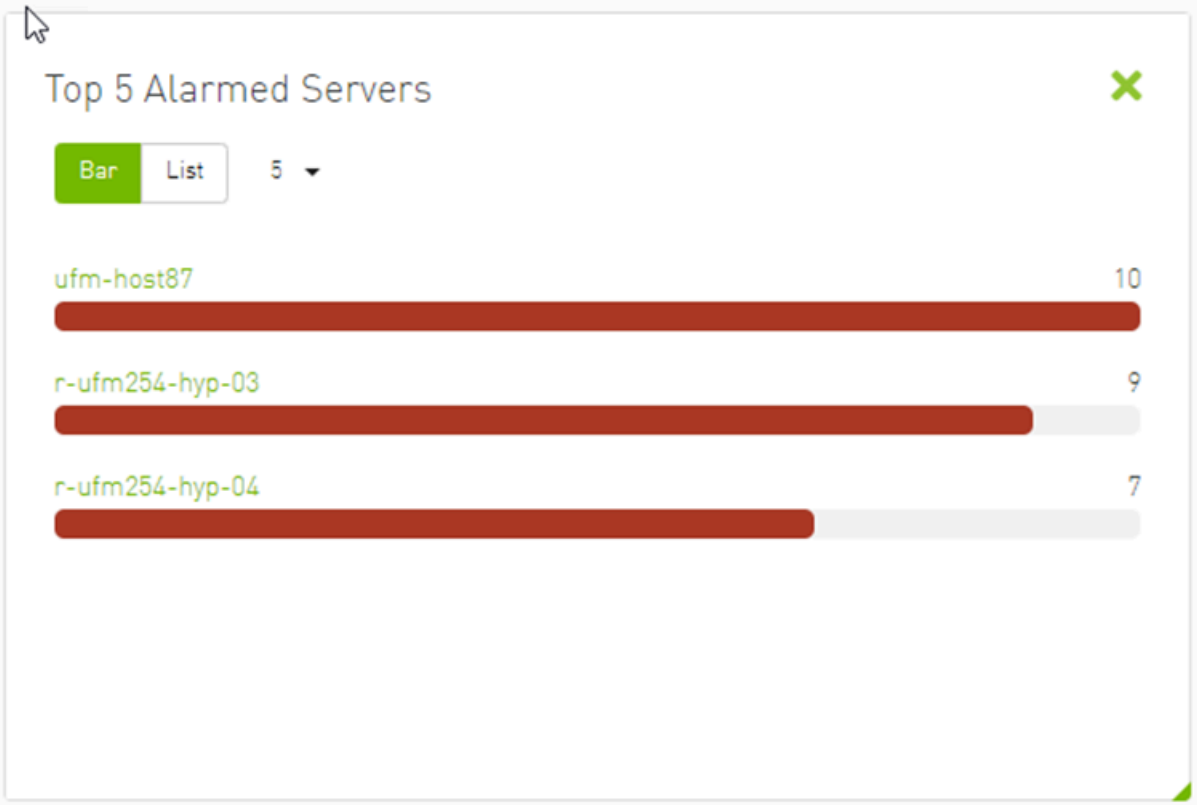
Options	Description
List view 	Shows the top N alarmed servers/switches as a list. Each alarmed device is shown in a row with the name of the node and the number of alarms.
Bar view 	Shows the top N alarmed devices as a bar graph. • X axis shows the number of alarms • Y axis shows the names of the alarmed nodes (servers)
Drop down menu 	Enables selecting the number of top N alarmed nodes. Selects the number of items to display. Default: 10 alarmed nodes
Filter toggle 	Toggles the Filter textbox

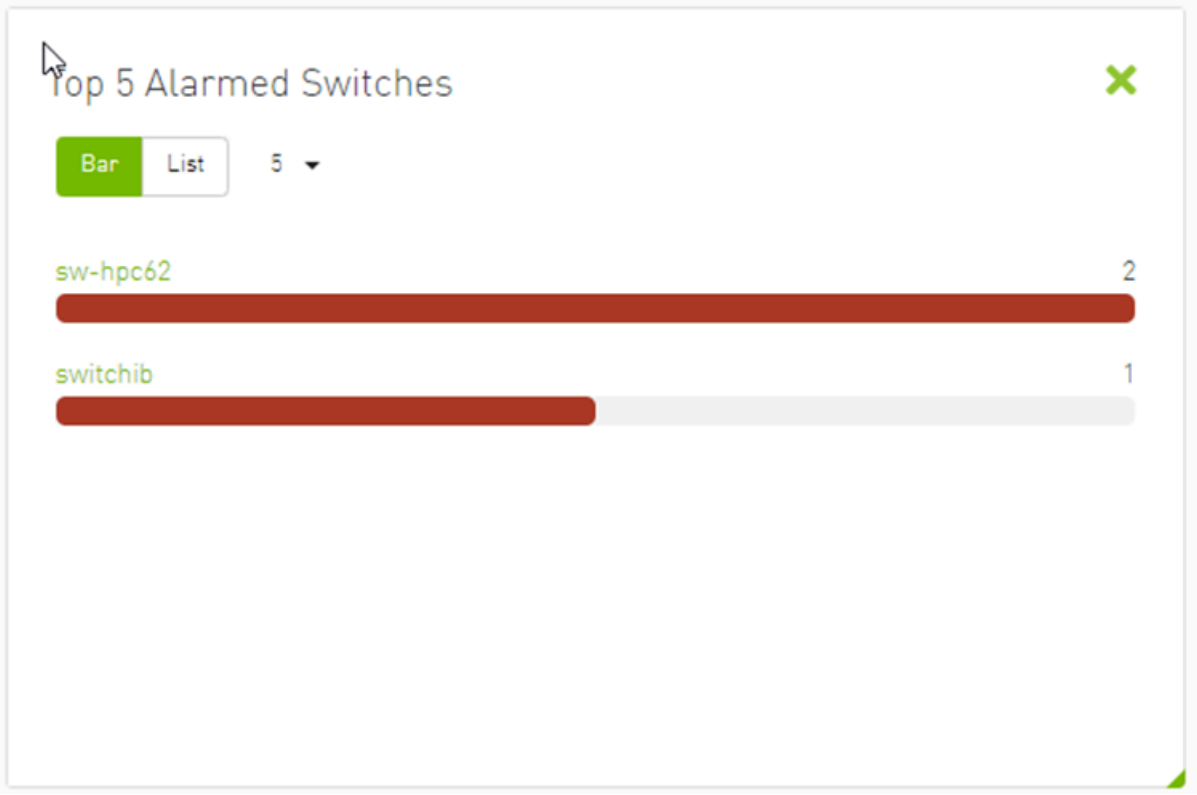
Top Alarmed Servers/Switches—List View





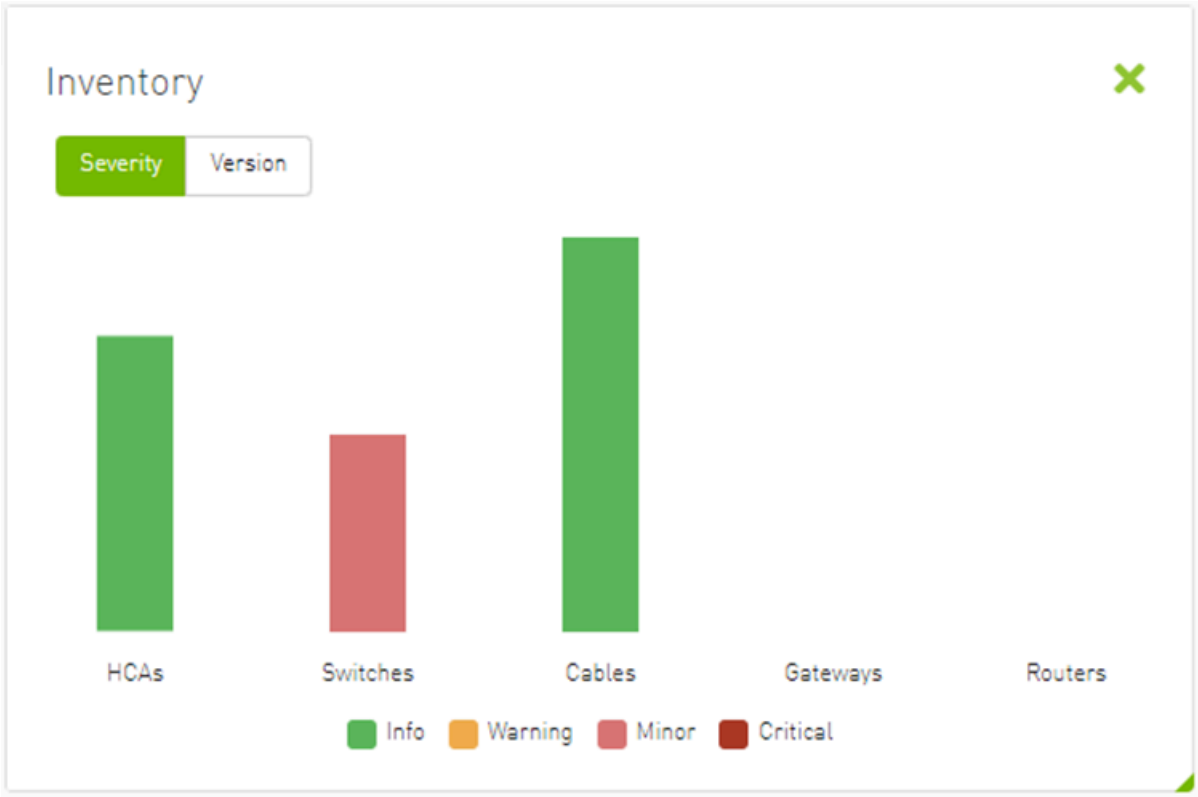
Top N Alarmed Servers/Switches—Bar View

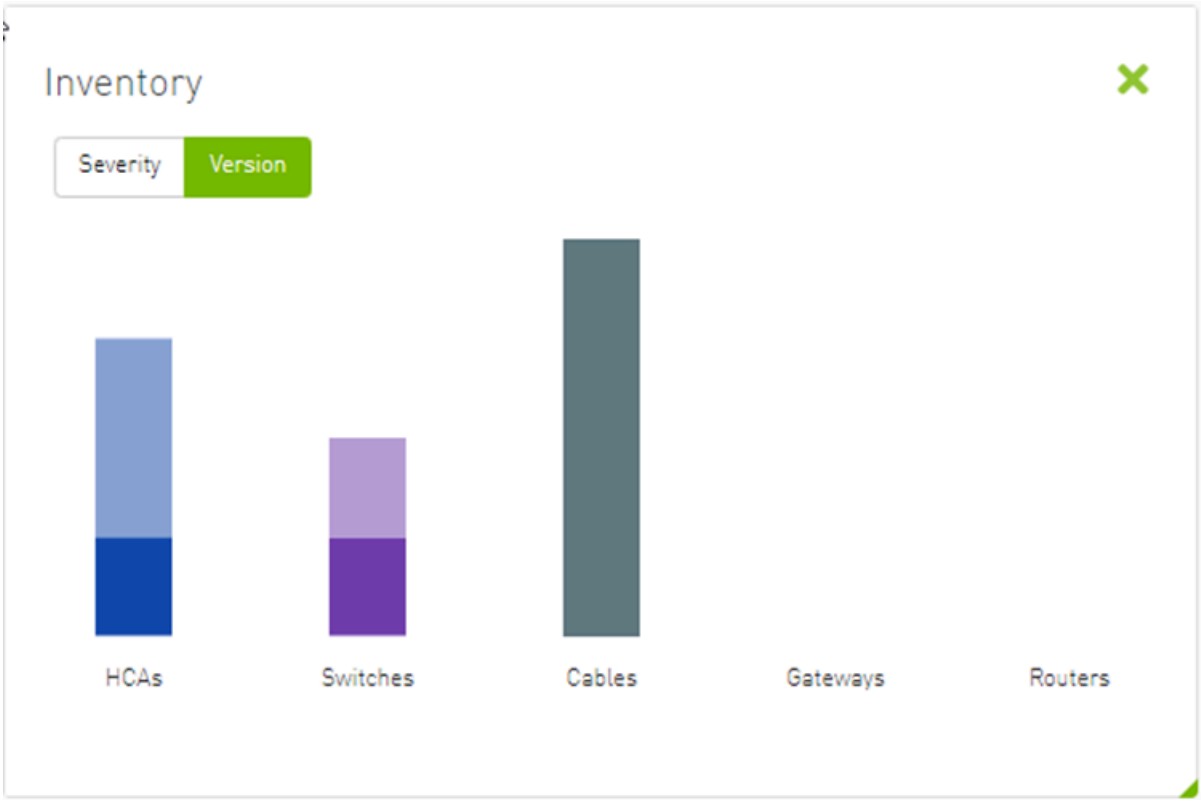




Inventory Summary

The Fabric Inventory Summary component shows a summary of your fabric inventory (HCAs, Switches, Gateways, Routers and Cables) categorized by the element's severity or firmware version.





Clicking on one bar element with specific severity/firmware version will redirect you to the clicked element's table.

Fabric Utilization

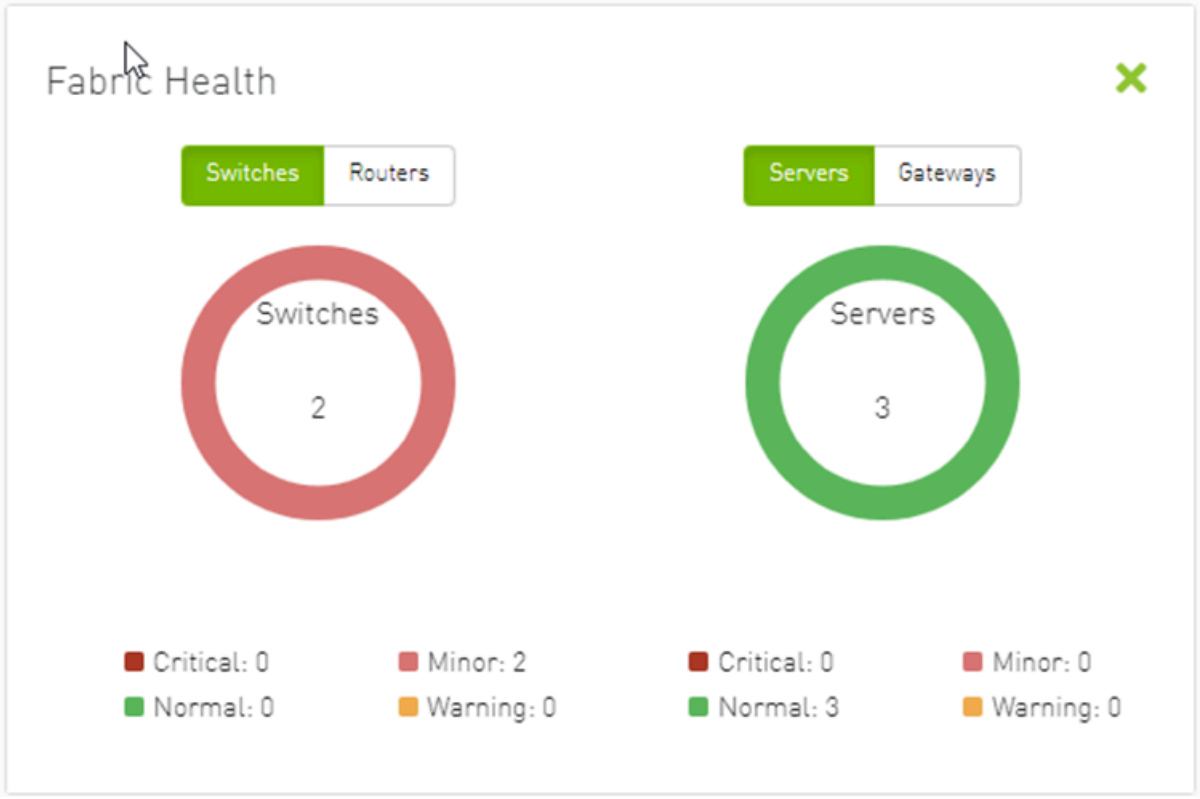
The Fabric Utilization component shows the number of alarmed objects, categorized by the alarm's severity. They are as follows:

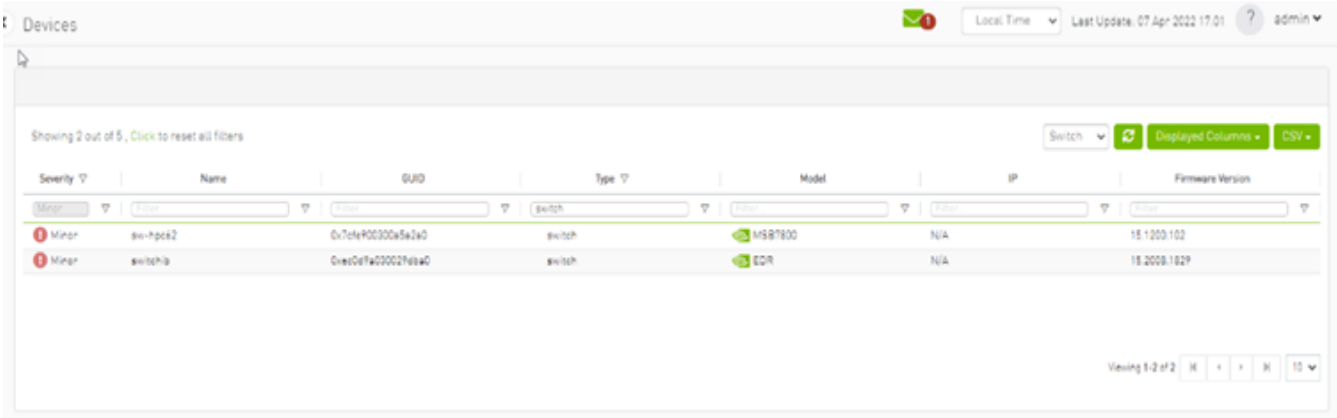
1. Warning
2. Minor
3. Normal
4. Critical

If Server X has 2 minor alarms, 1 warning alarm and 2 critical alarms, and Server Y has 0 minor alarms, 2 warning alarms and 1 critical alarm, the **Fabric Resource Utilization** pie chart will show 2 servers in the critical slice, 2 servers in the warning slice and 1 server in the minor slice.

You can filter for both switches and nodes of a specific severity level by clicking the specific pie slice indicating the severity.

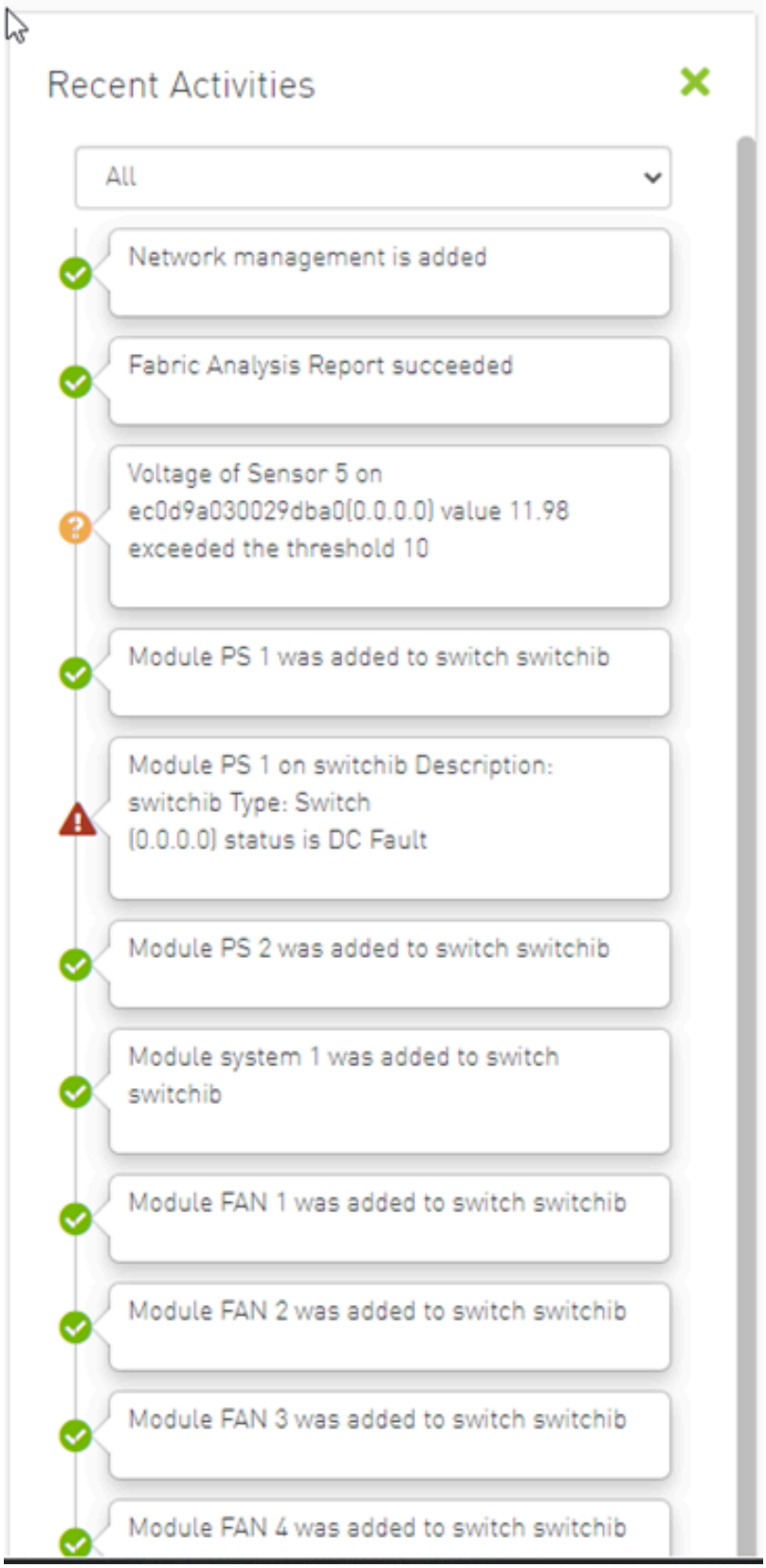
In the example below, the Devices table lists all the switches of severity level "Minor" after clicking the red (Minor) slice from the Switches pie chart.





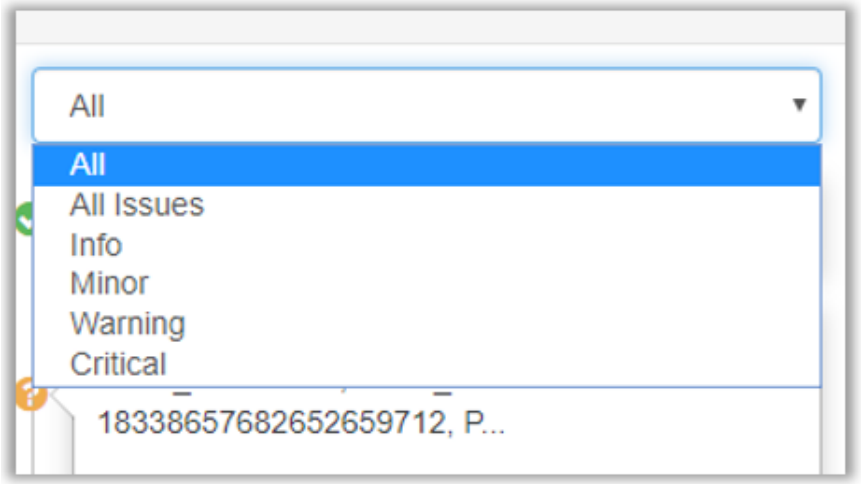
Recent Activities

The Recent Activities component lists the recent events detected by the UFM system.



You can filter for the events you would like to see in one list using the drop-down menu that provides the following options:

- All – shows all recent activities
- All issues – shows all non-Info activities
- Info – shows all activities with Info severity or higher
- Minor – shows you all activities with Minor severity or higher
- Warning – shows you all activities with Warning severity or higher
- Critical – shows you all activities with Critical severity

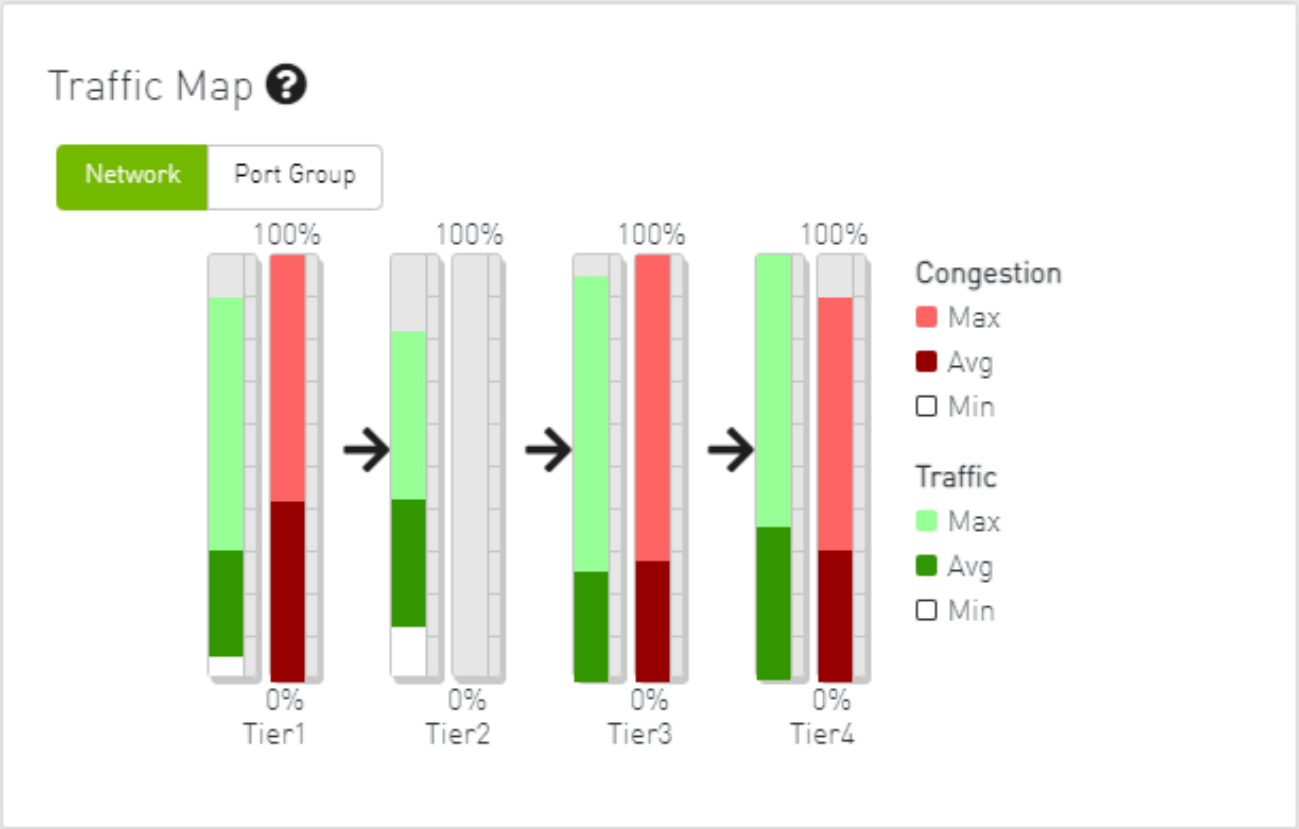


Traffic Map

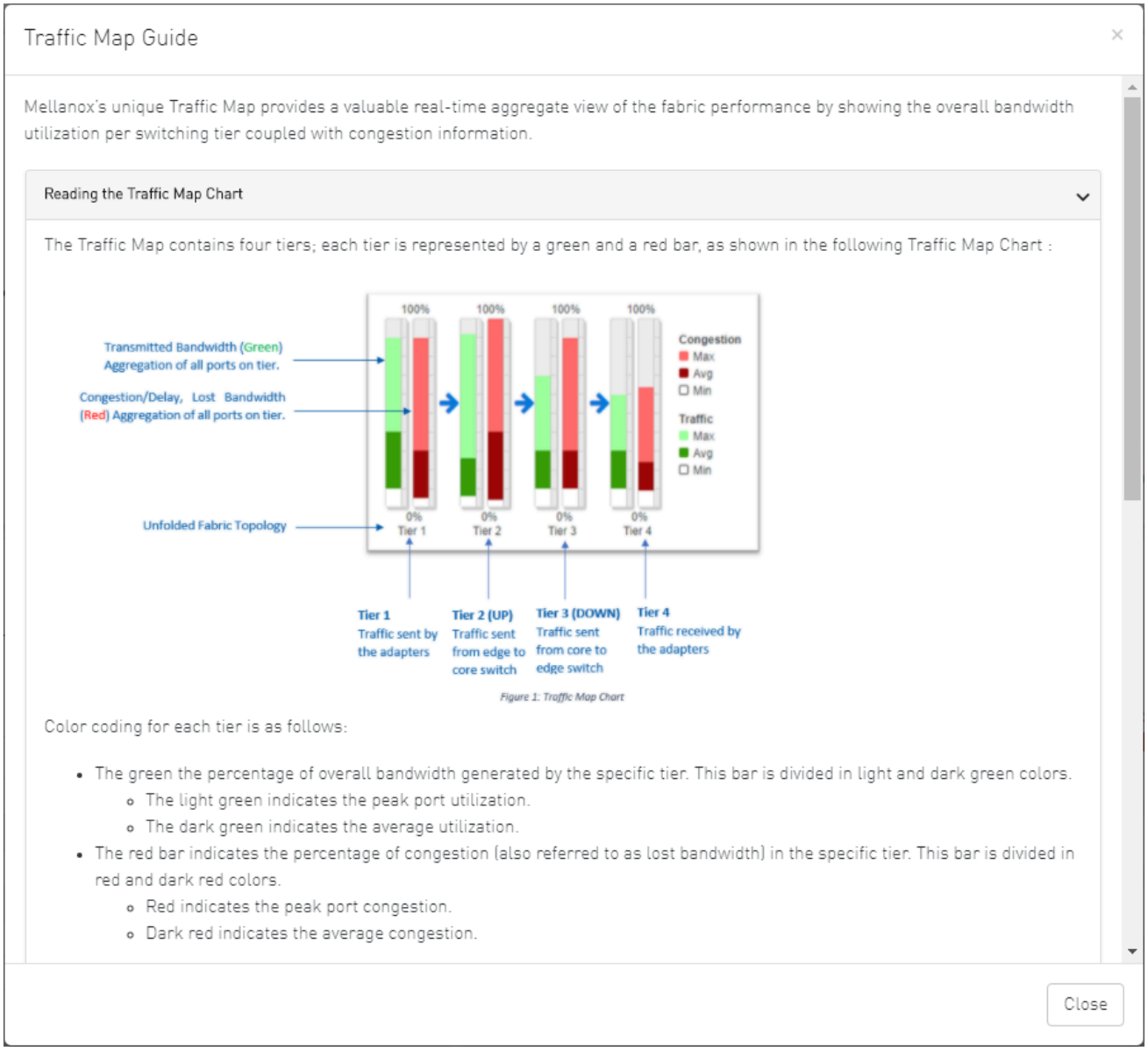
The Traffic Map dashboard shows the normal traffic versus congested traffic distributed on switch tiers and on port groups. This view, together with the **Top N Congestion** dashboard, gives a full status of the traffic congestion of the fabric.

Network Traffic Map

Four double bars represent the transmitted bandwidth (normalized transmit data) and normalized congested bandwidth (CBW), both measured in bytes/sec with minimum, average, and maximum bandwidth values.



An explanatory window on traffic map opens once clicked on the icon.



The percentage of total theoretical bandwidth (TBW) is calculated based on the underlying InfiniBand technology (SDR, DDR, QDR, FDR or EDR). The speed can be viewed when checking the ports.

- The vertical axis shows the following:
  - Bandwidth (BW) is represented by a green bar and is measured in percentages
  - Congested Bandwidth (CBW) is represented by a red bar and is measured in percentages
  - Minimum, average, and maximum bandwidth are represented in each bar by a subset color

- The horizontal axis represents the tiers.

  The bottom of the dashboard represents the tier-related transmitted traffic, which is divided into four segments by measurement ports:

- Tier 1 – represents the traffic injected by all adapters
- Tier 2 – represents the traffic sent from the edge switches to the core of the fabric (in case of a single Director switch, this tier indicates traffic utilization inside the Director between the line and fabric boards)
- Tier 3 – represents the traffic sent from the core to the edge switches
- Tier 4 – represents the traffic sent from the edge switch to the adapters

Note

The illustrations at the bottom of the tiers show a four-tier topology:

Server [tier 1] Switch [tier 2] Director Switch [tier 3] Switch [tier 4] Server.

Levels Network Traffic Map

Different representation of the fabric traffic map that based on the devices/ports levels.

The level of the device/port is the distance between the device and the nearest server/gateway.

Levels Calculations:

- The levels calculations are configurable from the `gv.cfg` file under TopologyLevels section enable item and it is disabled by default.
- The levels names are configurable from the `gv.cfg` file under TopologyLevels section levels item and by default we are defining up to 4 levels levels equals server, leaf, spine, core
    - Server: hosts and gateways.
    - Leaf: switches and routers that are directly connected to the server
    - Spine: switches and routers that are directly connected to the leaf
    - Core: switches and routers that are directly connected to the spine

If the fabric has more than 4 levels, the level value will be L + distance e.g., L4, L5, L(N), and if levels was empty, the levels will start from L0, L1, L2, etc.

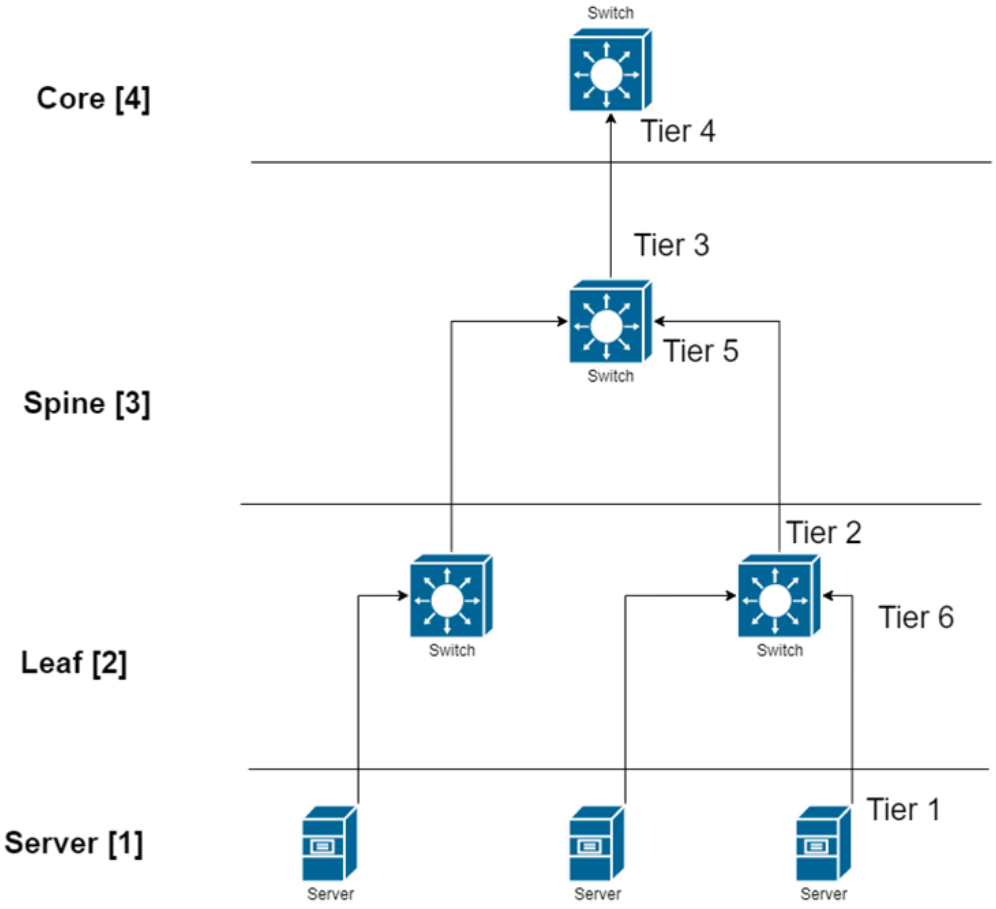
The levels calculations are done at either the discovery stage or once the topology changes.

Ports Tiers calculations based on the levels:

If the levels calculations is enabled, the port’s tier will be calculated as the following steps:

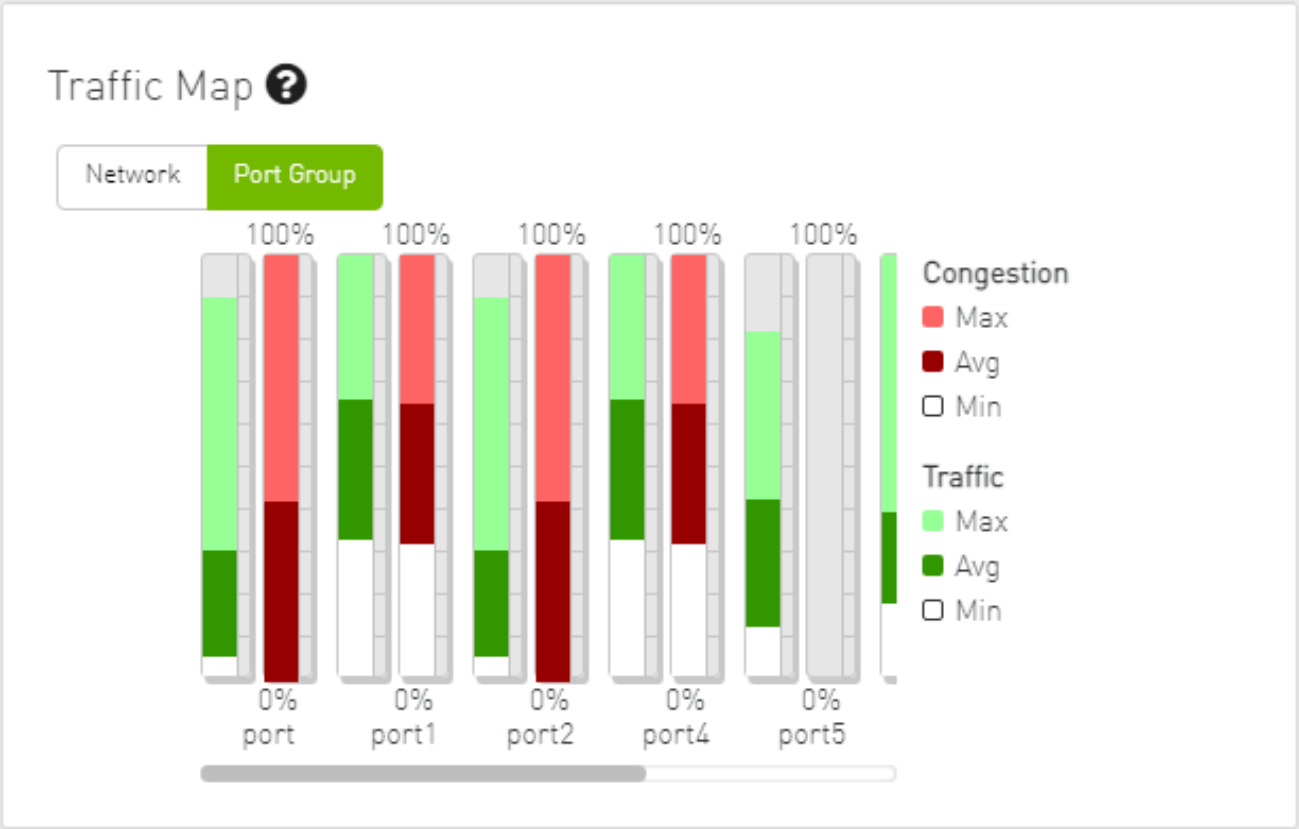
1. Get the level for both port's parent device and port's peer parent device
2. Decide whether the port’s data flow is the up or down direction, by checking the order of the parent and peer parent level:
    1. If the parent's level order is less than or equals the parent peer level, then the port's flow is up and tier is the parent level order
    2. If the port's flow is down and the tier is the distance between the host to the root device and the distance between the root to the parent device

Example:



If the level calculations are disabled, the tier calculations will be done as mentioned in this section.

Port Group Traffic Map



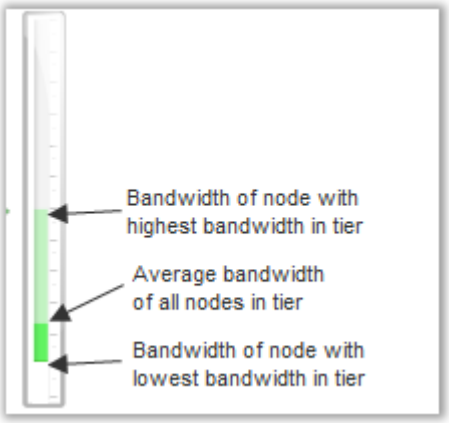
Traffic Map Bar Chart

- **Bandwidth Bars**

  The bandwidth graph shows how traffic is traversing the fabric and how traffic is being transmitted between the servers. For example, the following considerations could be evaluated:

  - The size of the difference between max bandwidth and min bandwidth.
  - The traffic that is flowing in the middle tiers and whether it would be more efficient to move the traffic to the edges to save the uplinks.

  Bandwidth levels are measured in percentages, as shown below:

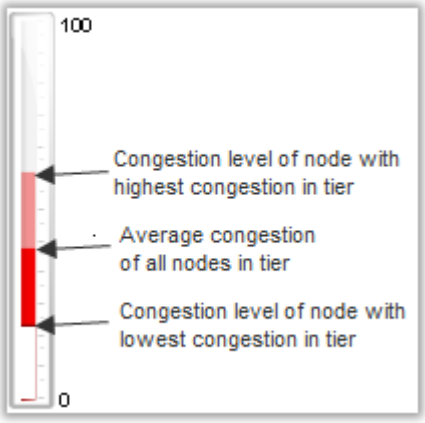


- **Congestion Bars**

  The Congestion graph shows where congestion starts. For example, the following considerations could be evaluated:

  - If congestion is in the first or second tier, there is probably a routing problem
  - If there is no red bar, it means that there is no congestion or no routing problems

  Congestion levels are measured in percentages, as shown:

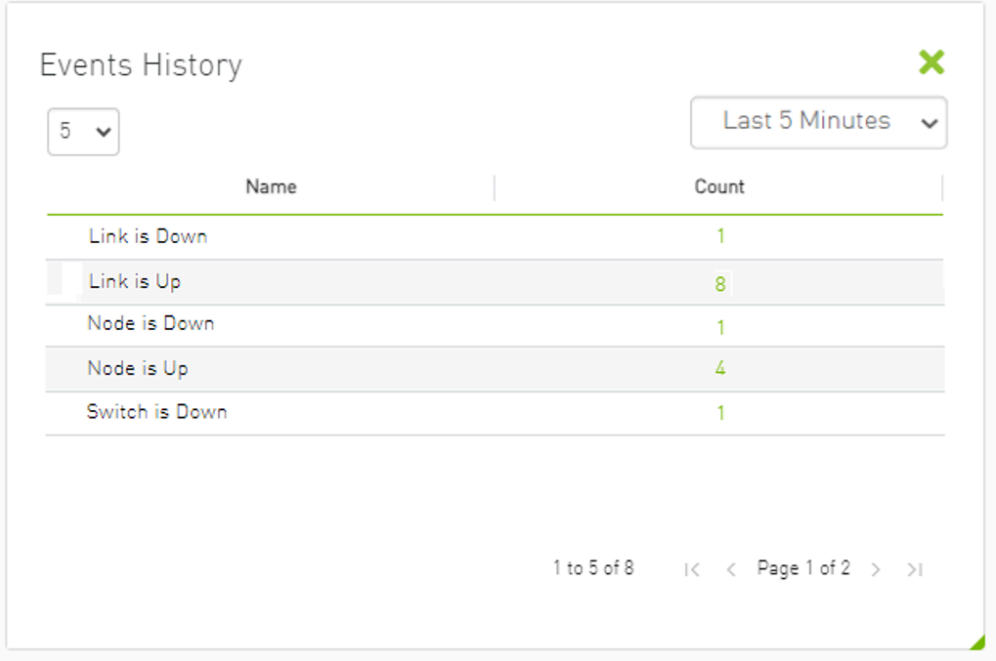


Events History

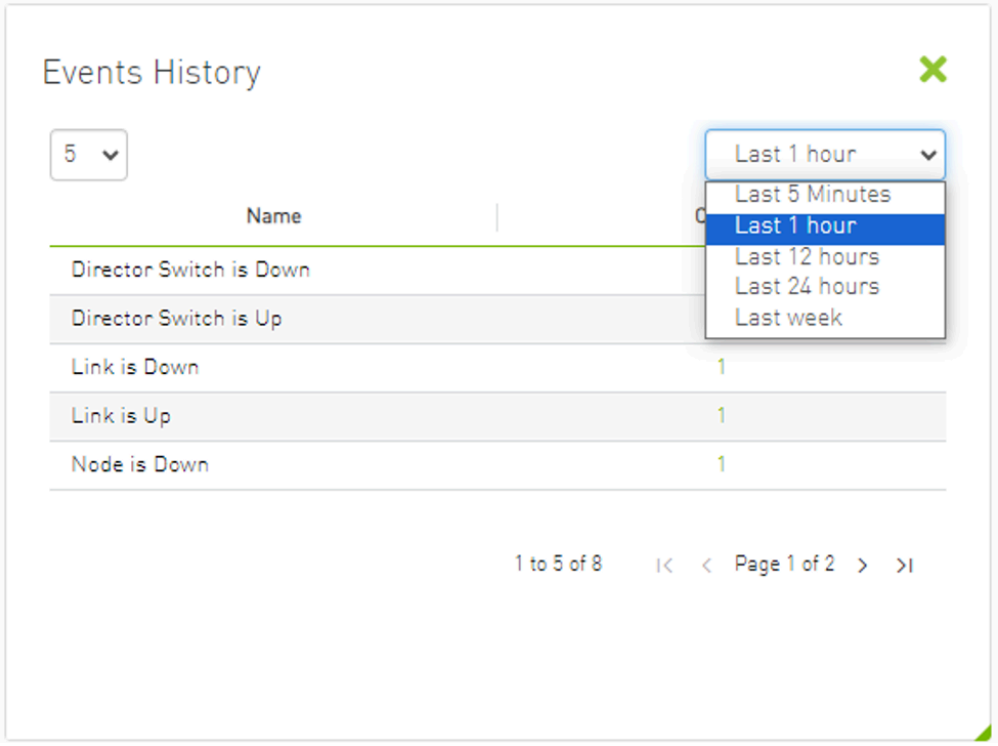
Note

To view the Event History panel in the dashboard, the System Monitoring feature must be enabled. Otherwise, the panel will be hidden. Users can enable System Monitoring by setting the `system_monitoring_metrics` flag under the `SystemMonitoring` section in the `gv.cfg` file to true.

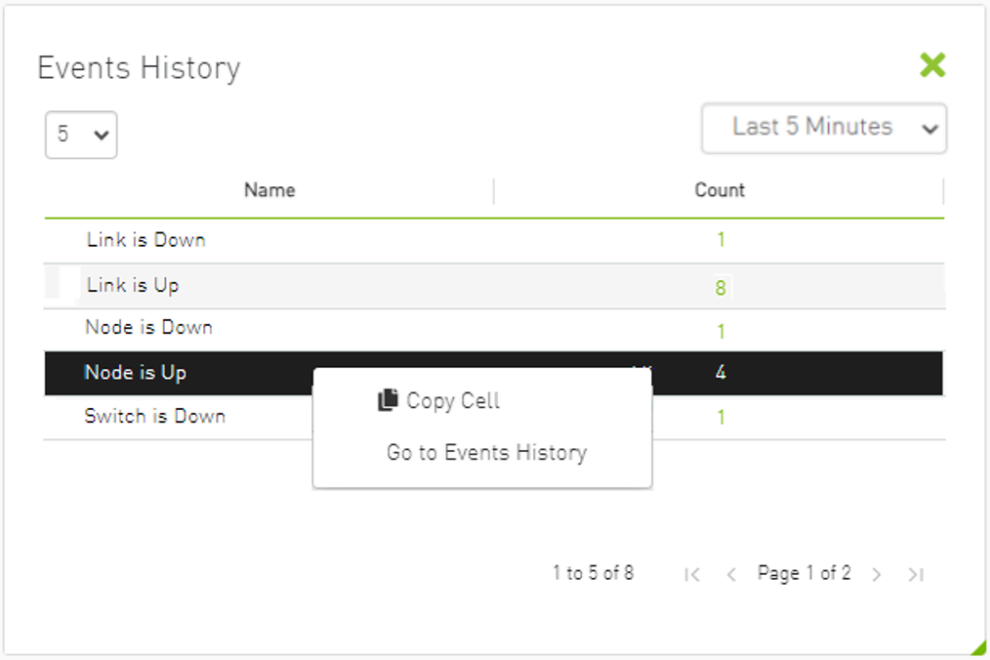
The Events History panel presents the topology change events in a table along with their respective counts.



The user can filter the event count by selecting the desired time interval.

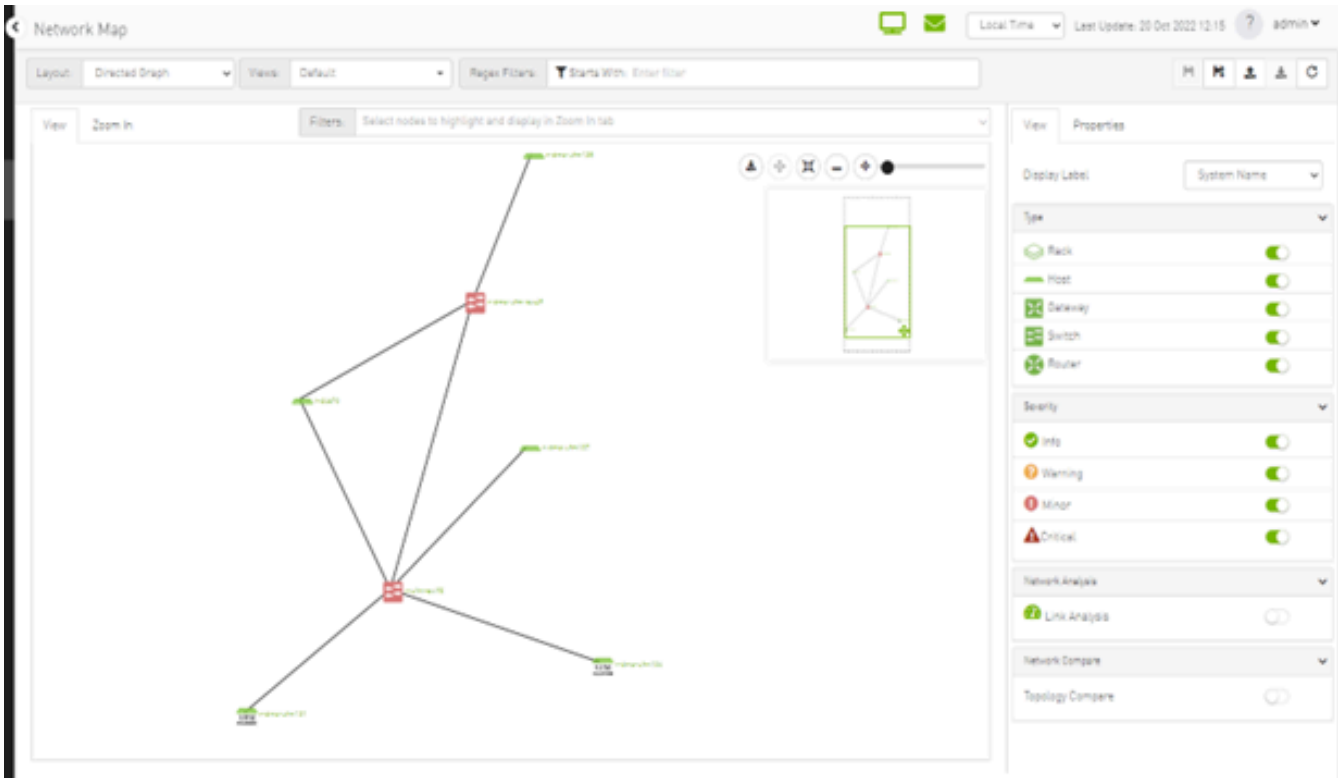


Users can navigate to the ‘Device/Link Status Events’ tabs by either clicking on the counter value or by right-clicking and selecting 'Go to Events History'.



Network Map

The Network Map window shows the fabric, its topology, elements and properties. UFM performs automatic fabric discovery and displays the fabric elements and their connectivity. In the Network Map window, you can see how the fabric and its elements are organized (e.g., switches and hosts).



Network Map Components

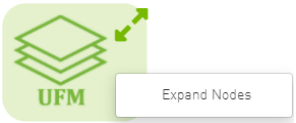
Component	Icon	Description
Switches		Represents third party switches discovered/managed by UFM
Hosts		Represents the computer (host) connected to the discovered/managed switches
Routers		Represents third party routers discovered/managed by UFM

Component	Icon	Description
Gateways		Represents third party gateways discovered/managed by UFM
Links		Represents the connections between devices on the fabric
Racks		Represents all nodes (hosts) physically connected to a switch

Note

The level of severity of devices affects the color they are displayed in.
For further information, refer to table "Device Severity Levels".

- To zoom in/out of the map, scroll the mouse wheel up and down or using the slider on the right top corner
- To move around in the map, press and hold down the left key while you move sideways and up/down
- To see the hosts inside a rack, right-click the Rack icon and click "Expand Hosts"



Selecting Map Elements

Users are able to select elements from the Network Map. Right-clicking an element opens a context menu which allows users to perform actions on it.

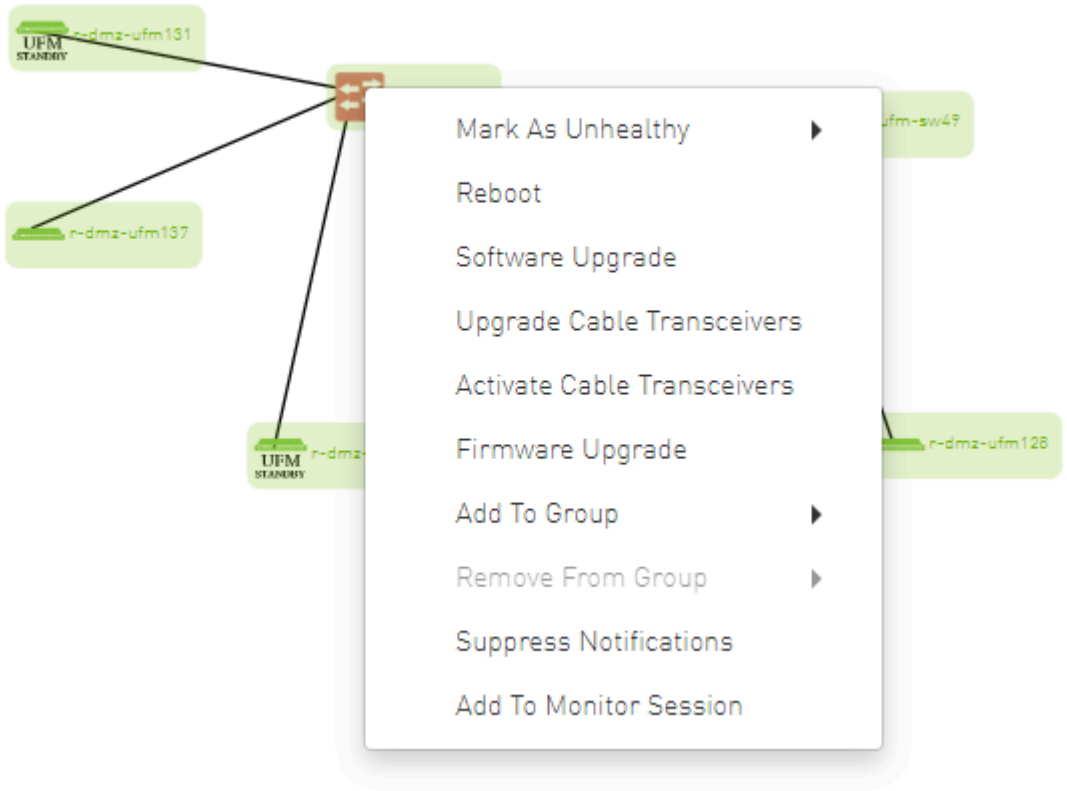
It is possible to select multiple elements at once using any of the following methods:

- By holding down Ctrl or Shift and dragging their mouse across the map.

> **Note**
>
> Please note that Ctrl starts new selection, while Shift adds to the current selection.

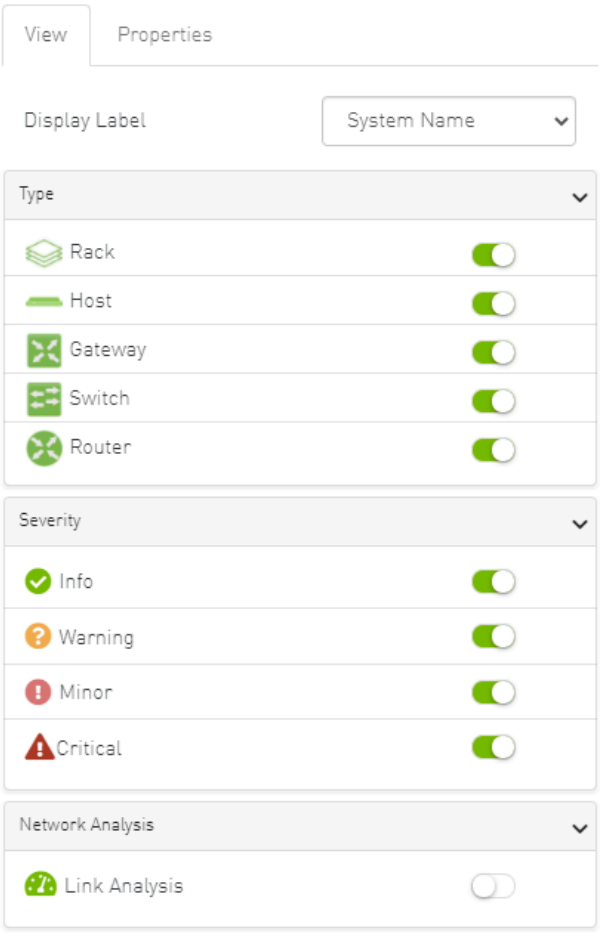
- By holding down Shift and clicking a new element on the map.

Multi-select makes it possible for users to perform actions on multiple devices with one right-click rather than repeating the same process per device.

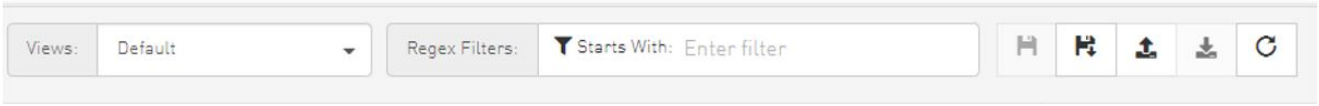


Map Information and Settings

The right pane of the Network Map view enables you to control the view settings, as well as obtain further information on selected elements from the map.



The customized views created using the type and severity filters, selected fabric nodes, zoom level, and Expand/Collapse All Racks options can be saved for later access. These customized views can be saved and accessed using the bar available on top of the Network Map:

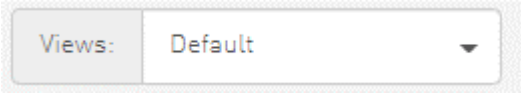


- "Save As" icon ( ) saves newly created customized views
- "Save" icon ( ) saves edits performed on existing views
- “Import” icon ( ) import map from local device. The file format should be txt

- “Export” icon ( ) export network as text file

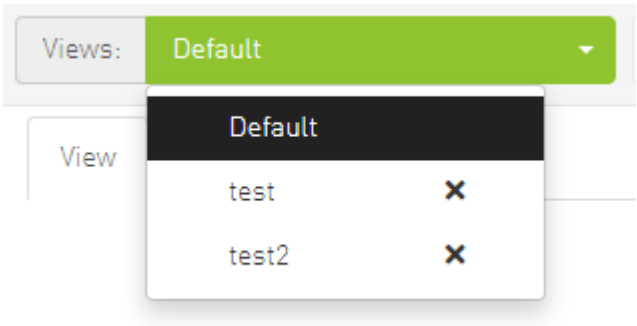
- To reload/refresh the network map, use the refresh icon ( ).

- Drop down menu gives access to all previously saved views

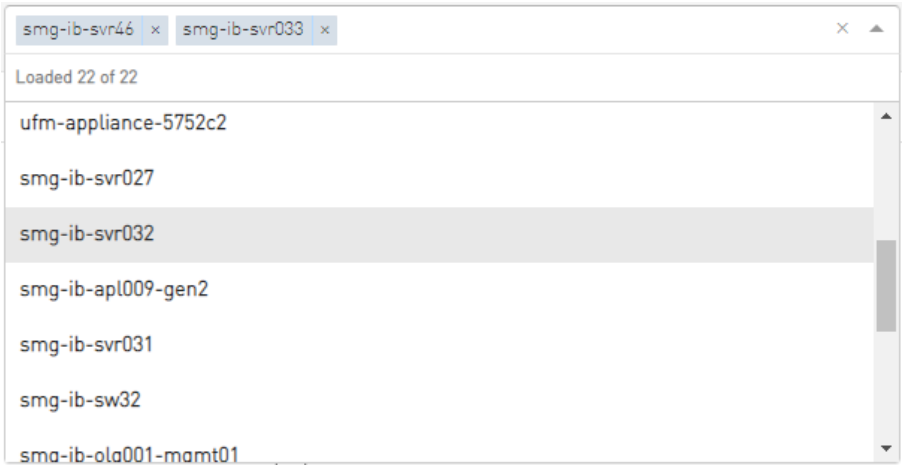


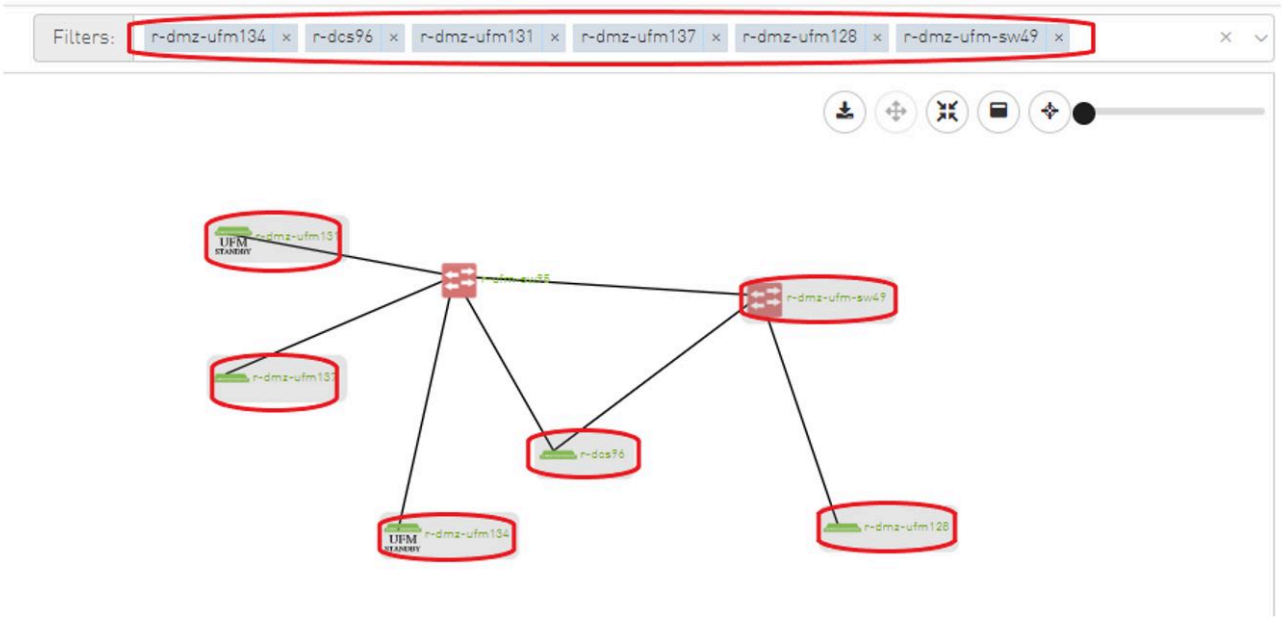
- "Default" view is a predefined view where nodes are positioned randomly, all filters are enabled, and all racks are collapsed. Changes made to this view cannot be saved unless under a new view name using the "Save As" icon.

- Saved views can be deleted using the "x" button.



You can select a node from the dropdown menu located above the Network Map view in order to highlight/display them in the "Zoom In" tab.





Map View Tab

The Network Map "View" tab displays the fabric containing all nodes (e.g. switches, racks including the hosts, etc).

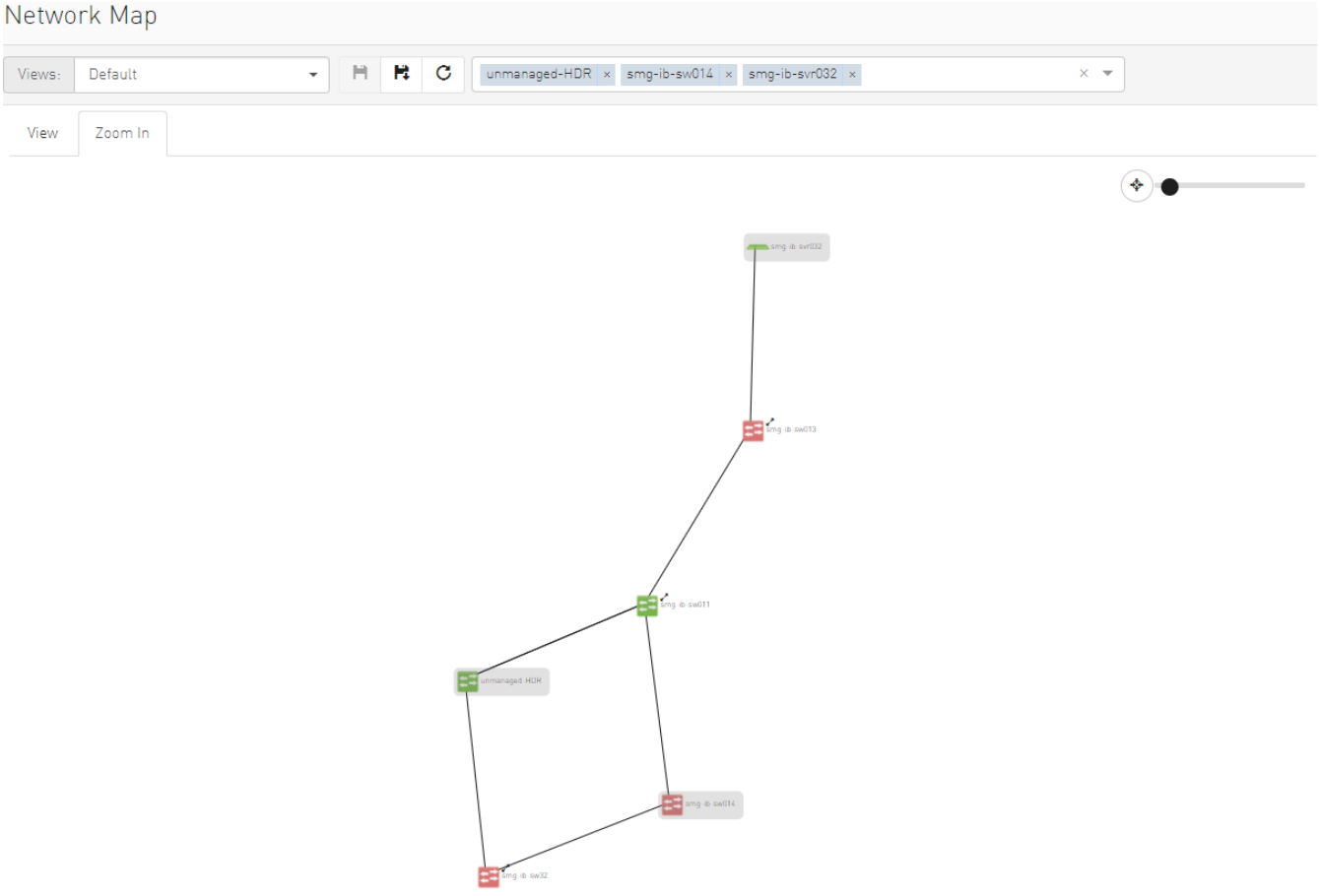
If your fabric consists of more than 500 nodes, please note that:

- The "View" tab will show only the switches in your fabric. Therefore, "Expand all racks" and "Rack filter" functions will be disabled.
- Link analysis will be disabled.

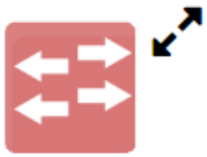
To have a better experience in this instance, you can switch to the "Zoom In" tab.

Map Zoom In Tab

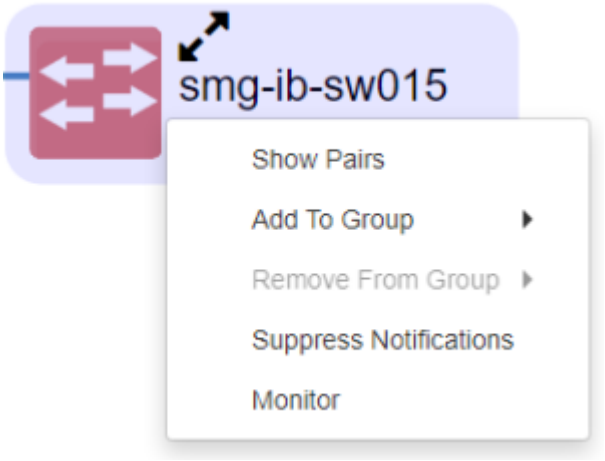
The Network Map "Zoom In" tab displays only the selected nodes from the dropdown menu above the map view and the nodes directly connected to the selected nodes.



If some switches still have hidden connected nodes, you will see the following icon:



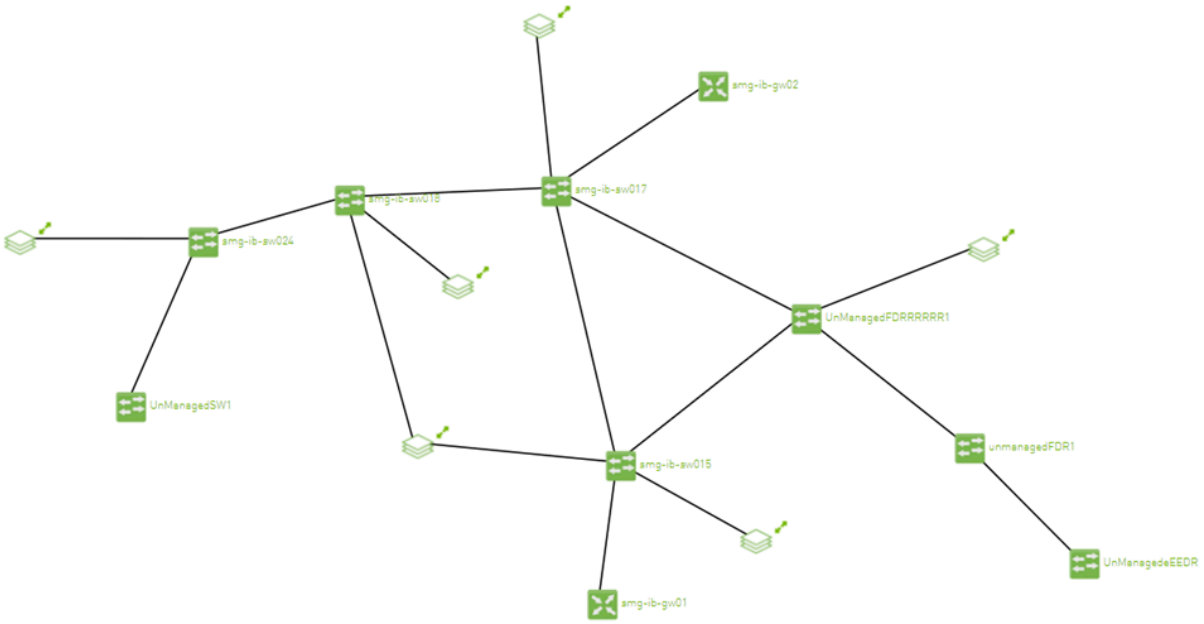
To reveal the hidden nodes connected to this switch, you can right-click it and select "Show Pairs" which adds this switch to the selected nodes list and shows the direct connected nodes to this switch.



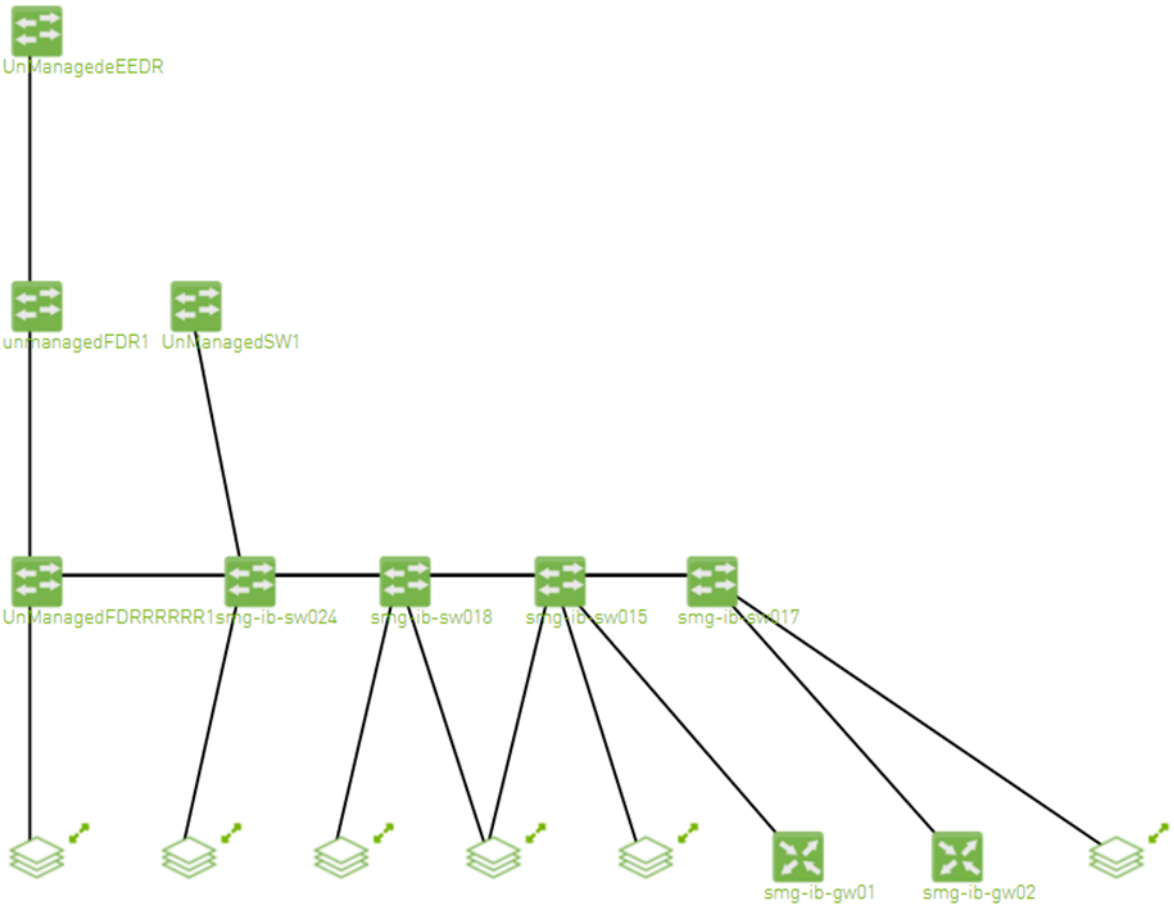
Map Layouts

Layout controls nodes positions in the map. UFM network map supports two types of layouts:

- Directed layout: the nodes are distributed depending on the connections between them so that the connected nodes will be near each other without conflict.



- Hierarchical layout: the nodes are distributed as layers; each layer will contain nodes that have the same level value.



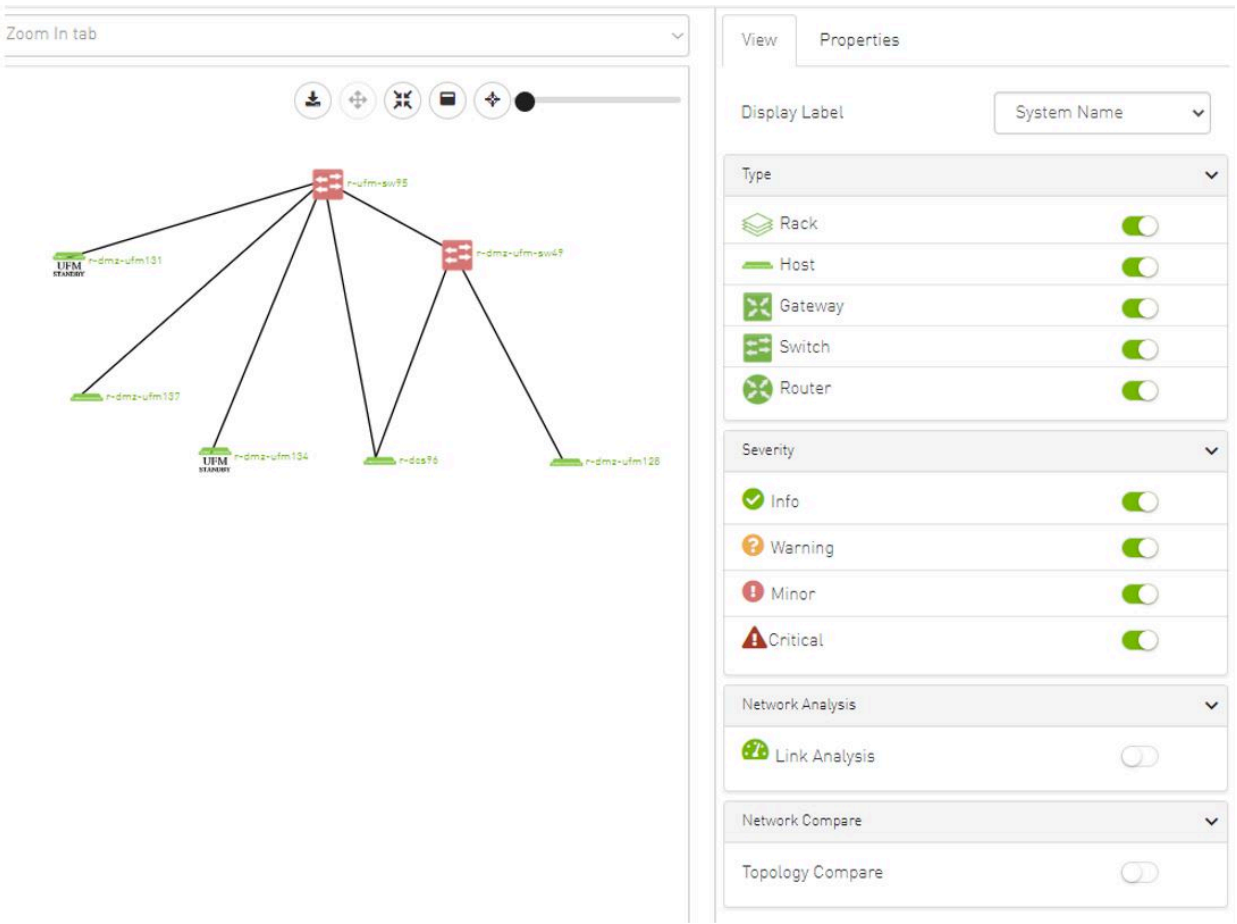
You can switch between layouts from the dropdown menu located above the Network Map view.

Information View Tab

- Enables searching for one or more elements in the map, by typing either their name or their GUID in the Search field. Note that the search mechanism is **not** case-sensitive.
- Enables displaying the elements either by their name, GUID, or IP.
- Enables viewing all hosts of all racks in the fabric using the "Expand All Racks" button.

- Enables customizing the view of the map by filtering for certain elements to appear in the map using the Type (see table "Network Map Components") and Severity (see table "Device Severity Levels") filters. Example:



Device Severity Levels

Component	Description
	Info
	Critical
	Minor
	Warning

Link Analysis

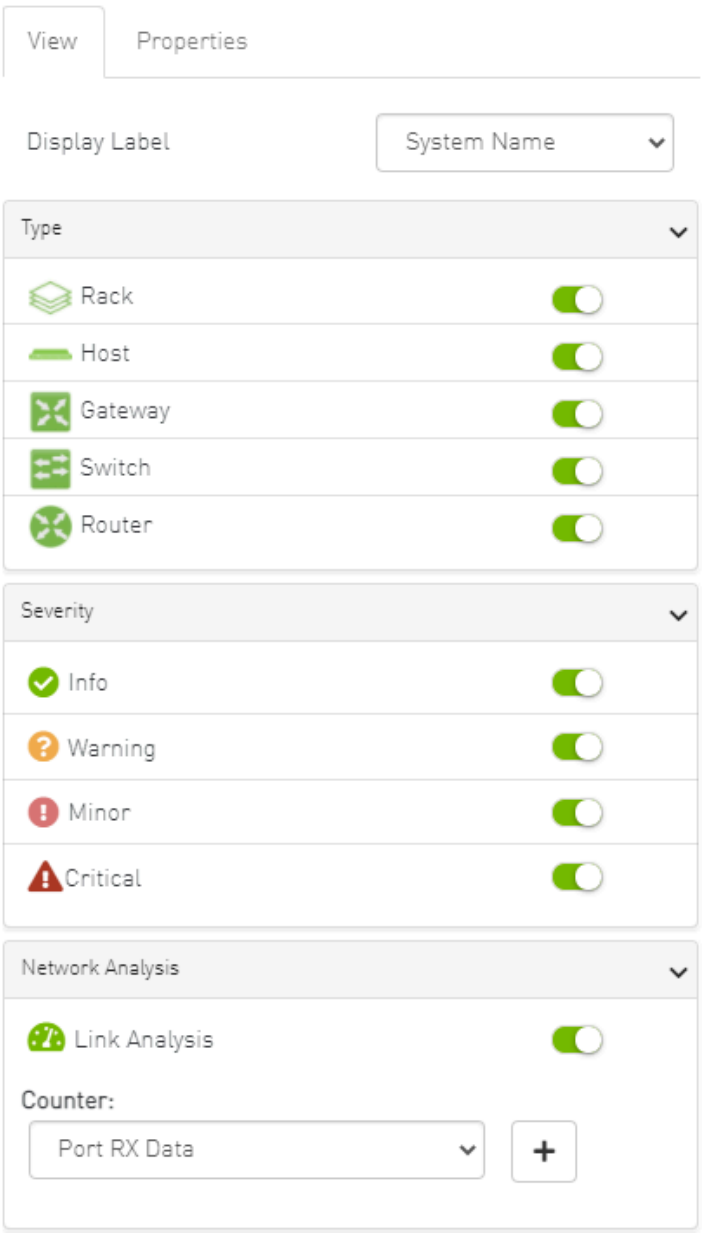
Link analysis allows the user to display the link analytics according to a selected static counter, and define the conditions on which the analysis is based. The links are colored according to the specified conditions. It is possible to define up to five conditions per counter.

The counter's conditions are applied on four values:

- The source values of the selected counter
- The destination value of the selected counter
- The source value of the opposite of the selected counter
- The destination value of the opposite of the selected counter

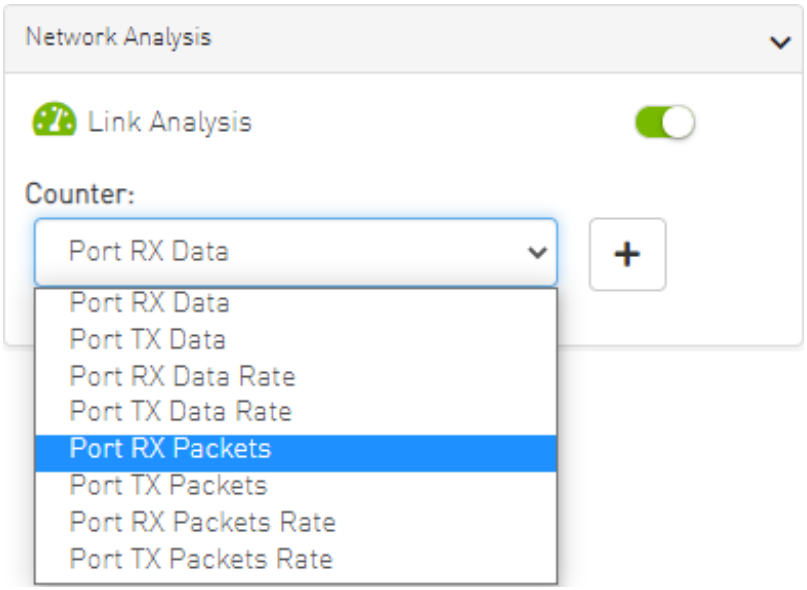
The worst matched value between these four is taken into consideration.

The "Network Analysis" section on the right side under the View tab contains a radio button to enable/disable the link analysis.

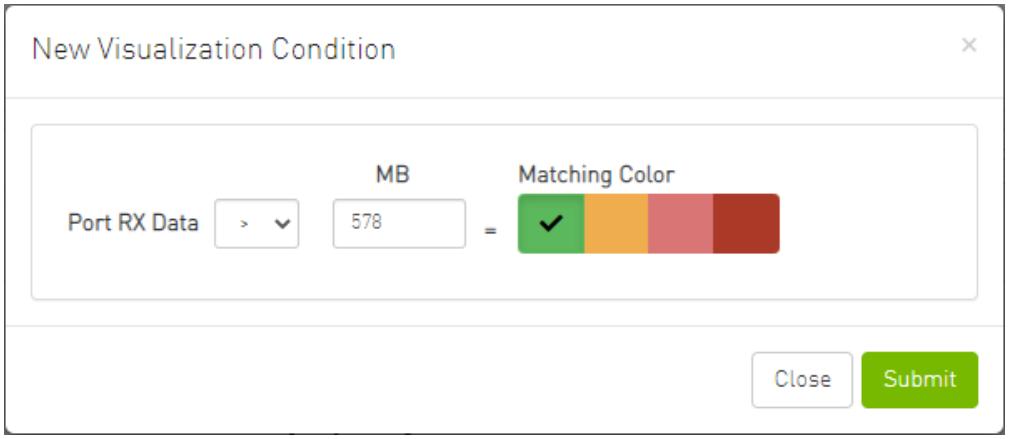


To define a condition:

1. Select the desired counter, and click the + button.



2. Select the appropriate operator, and define the desired threshold and color on the form that pops up. This color is applied on the link if the link monitoring value matches the respective condition.

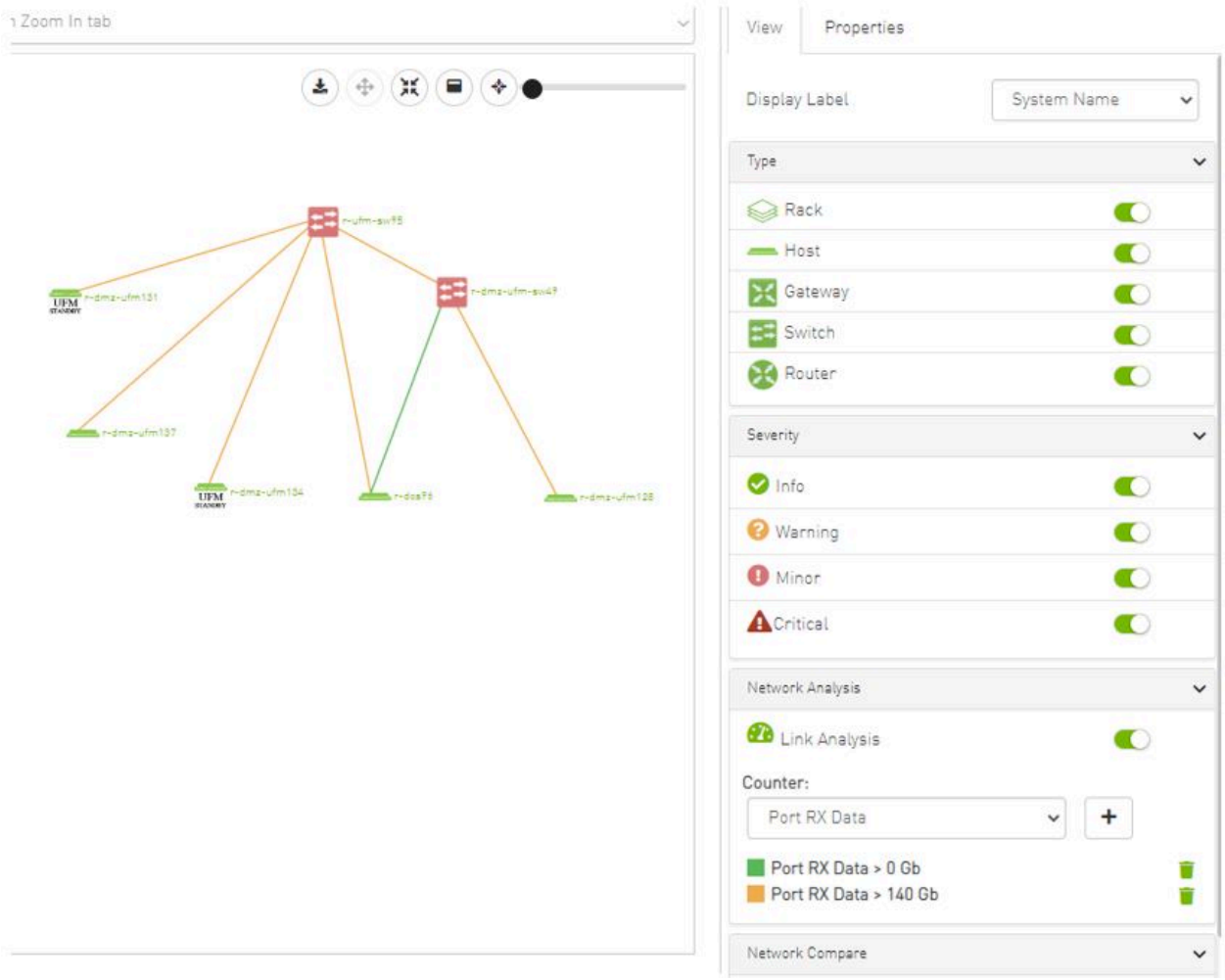


Note

The colors are sorted from the lowest to the highest priority (i.e from left to right, green to red).

> **Note**
>
> The counter's conditions are sorted based on the threshold values:
>
> - Ascending if the operator is greater than (>)
> - Descending if the operator is smaller than (<)
>
> Last matched condition's color are taken into consideration in the link coloring.

3. Once the condition is set, the network map lights up the links that meet your condition.



Note

Note how the added conditions are listed in the Network Analysis section, if Link Analysis is enabled, and they are colored accordingly.

View | Properties

Link 1

Link/Port Properties

Property	Source	Destination
System GUID	0x0002c903007b78b0	0xb8599f0300fc6de4
Port	1	3
MTU	4096	4096
Width	4X	4X
Speed	FDR	FDR
Port RX Data	20379.85 Gb	5.9 Gb
Port TX Data	18.05 Gb	6134.55 Gb
Port RX Data Rate	0 Gb/s	0 Gb/s
Port TX Data Rate	0 Gb/s	0 Gb/s
Port RX Packets	1285841763 Packets	7796207 Packets
Port TX Packets	22720574 Packets	386937725 Packets
Port RX Packets Rate	2.9 Packets/s	2.9 Packets/s
Port TX Packets Rate	2.9 Packets/s	2.9 Packets/s

Cable Info

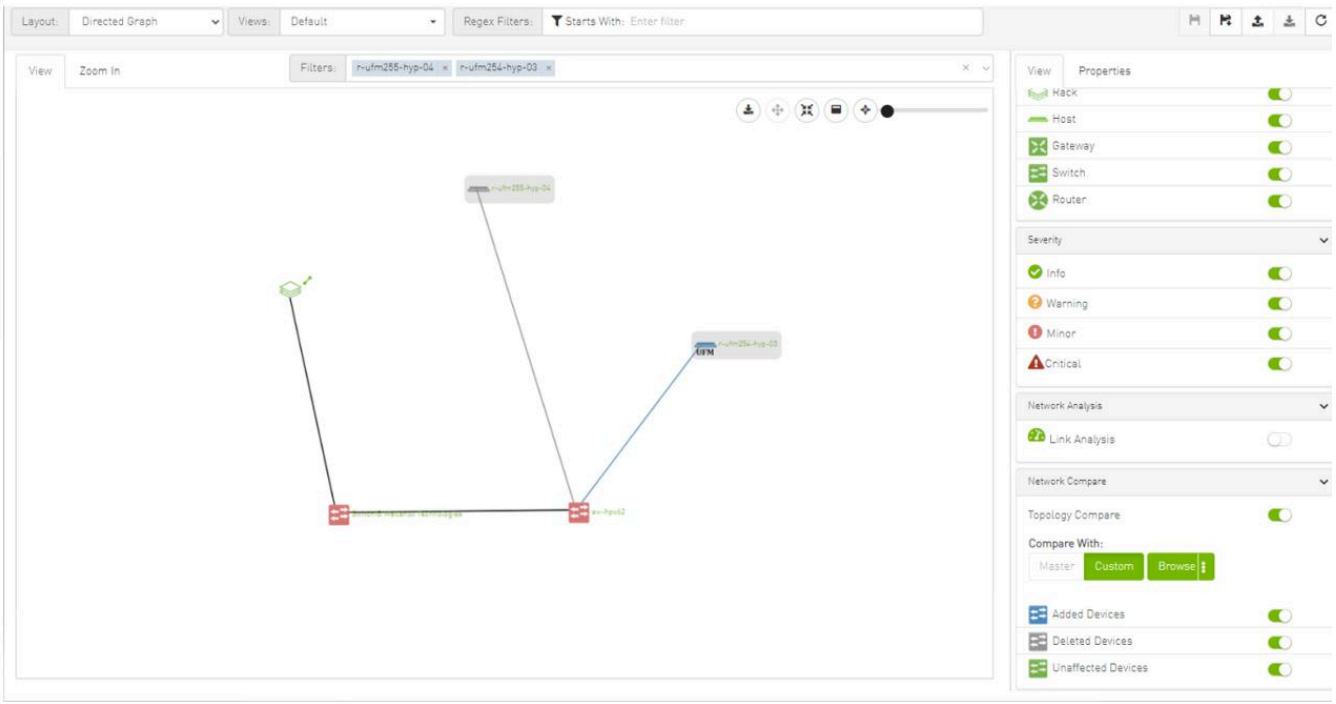
Property	Value
Part Number	MCP1600-E001
Length	1 m
Serial Number	MT1625VS05686
Identifier	QSFP+
Technology	Copper cable- unequalized
Revision	A2

> **Note**
>
> Notice how the monitored counter is presented in boldface, and the background color is presented with the worst matched condition.

Please note that if the current layout and view are saved, the defined conditions are saved inside the view being saved.

Topology Compare

It is possible to enable the Topology Compare feature from the View tab in the right-hand pane. When the radio button is enabled, it is possible to compare the current topology with the master topology or with a custom topology whose `.topo` file you may upload.



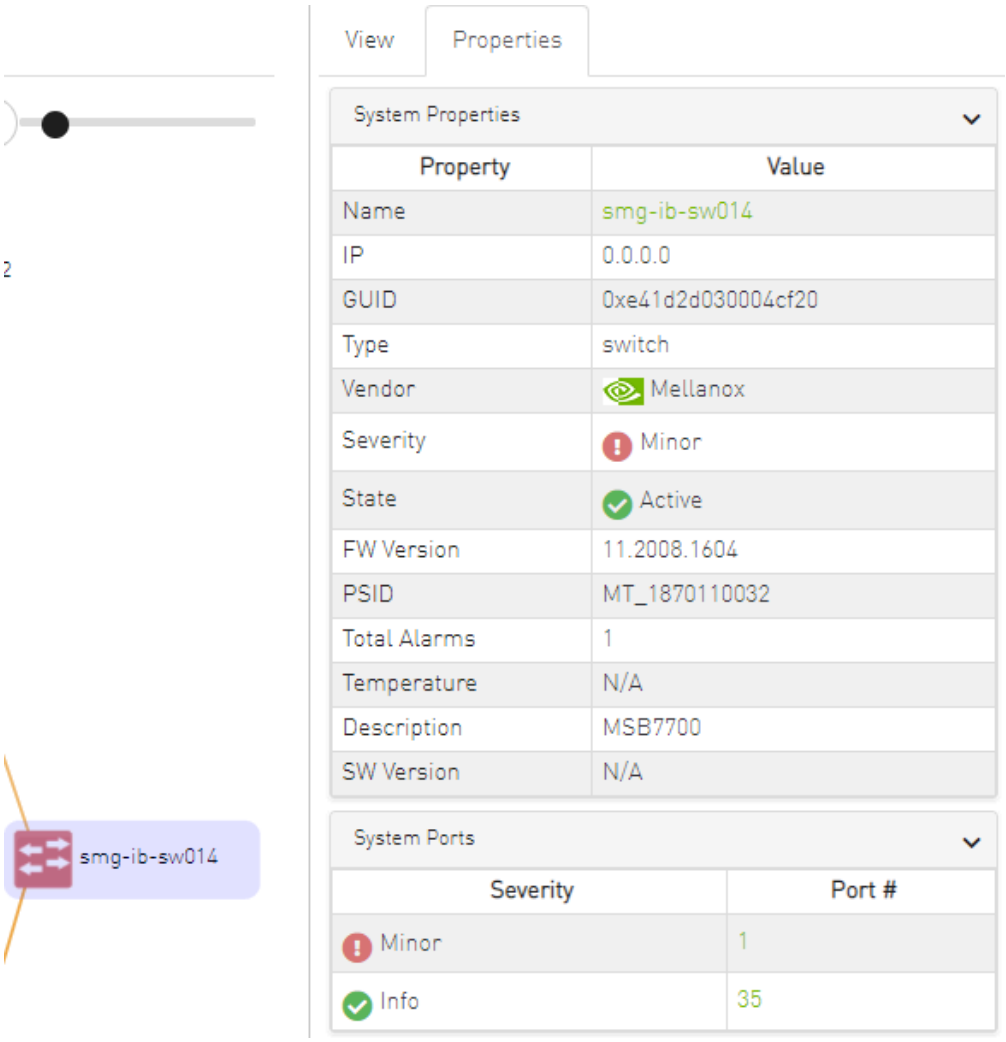
Topology compare key:

- A blue node signifies an added node
- A gray host signifies a deleted node

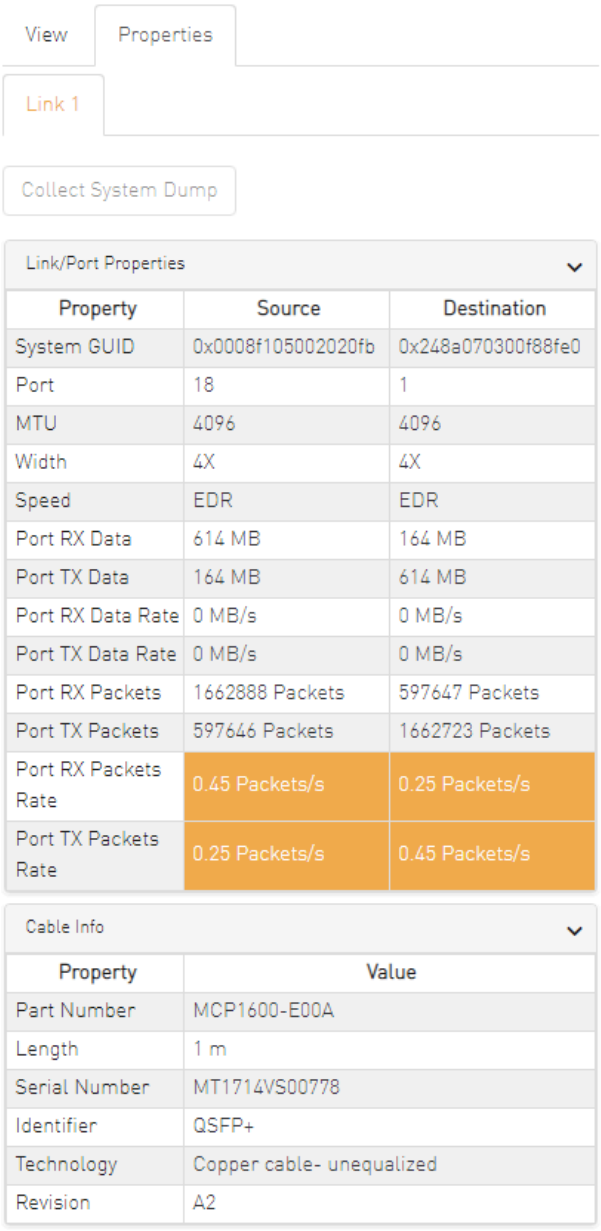
- A gray and black line signifies that some links were deleted and others were unchanged
- A gray and blue line signifies that some links were deleted, and others were added
- A gray, blue, and black line signifies that some links were deleted, some were added, and some were unchanged
- A blue and black line signifies that some links were added, and some were unchanged

Properties Tab

- Provides details on a specific system selected from the map, as shown in the following example:



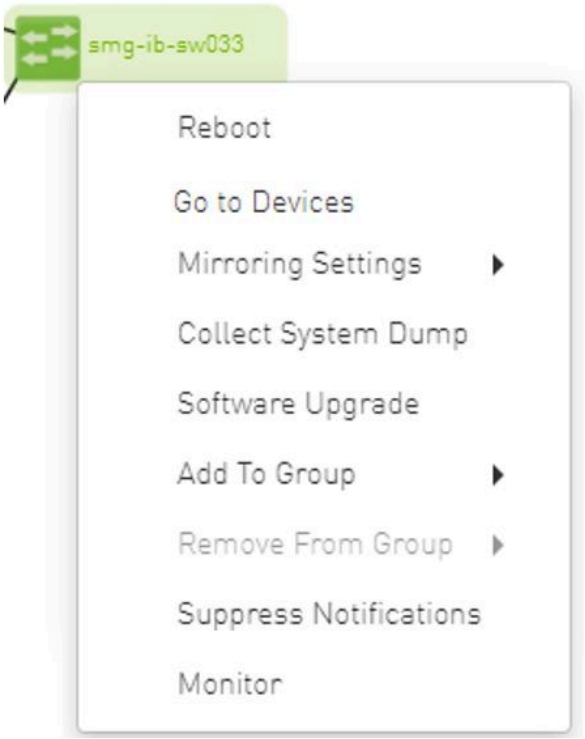
- Provides link/port properties and cable info on a specific link selected from the map, including destination and source ports, as shown in the following example:



Network Map Elements Actions

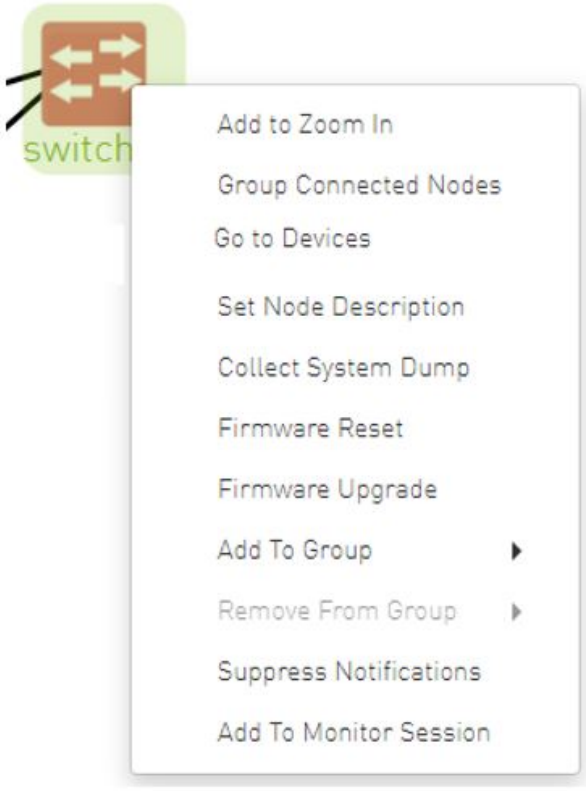
In the Network Map, a right-click on any of the elements enables performing a set of actions depending on the element type and its capabilities. See the list of available actions for each element type in the tables below.

Supported Actions for Internally Managed Switches



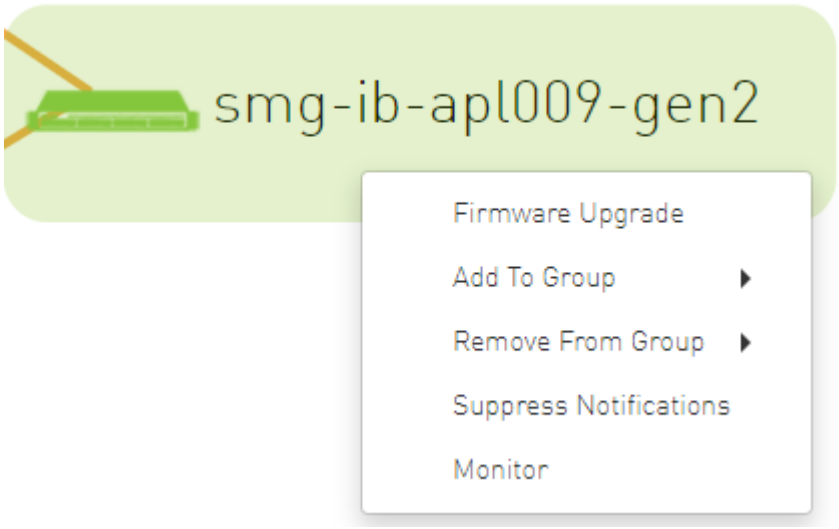
Element Type	Supported Actions	Description
Managed Switch	Reboot	Reboot the switch software
	Mirroring Settings	Set the mirroring configuration for the switch
	Collect System Dump	Collect system dump from the device
	Software Upgrade	Perform switch software upgrade
	Add to Group	Add switch to logical group
	Remove from Group	Remove switch from logical group
	Suppress Notification	Suppress all event notifications for the switch
	Monitor	Configure and activate switch monitoring
	Go to Devices	Go to devices page and select the device

Supported Actions for Externally Managed Switches



Element Type	Supported Actions	Description
Externally Managed Switch	Set Node Description	Sets description for specific node
	Firmware Reset	Perform switch firmware reset
	Firmware Upgrade	Perform switch firmware upgrade
	Add to Group	Add switch to logical group
	Remove from Group	Remove switch from logical group
	Suppress Notification	Suppress all event notifications for the switch
	Monitor	Configure and activate switch monitoring
	Go To Devices	Go to devices page and select the device

Supported Actions for Hosts



Element Type	Supported Actions	Description
Hosts	Firmware Upgrade	Perform switch firmware upgrade
	Add to Group	Add host to logical group
	Remove from Group	Remove host from logical group
	Suppress Notification	Suppress all event notifications for the host
	Monitor	Configure and activate host monitoring

Managed Elements

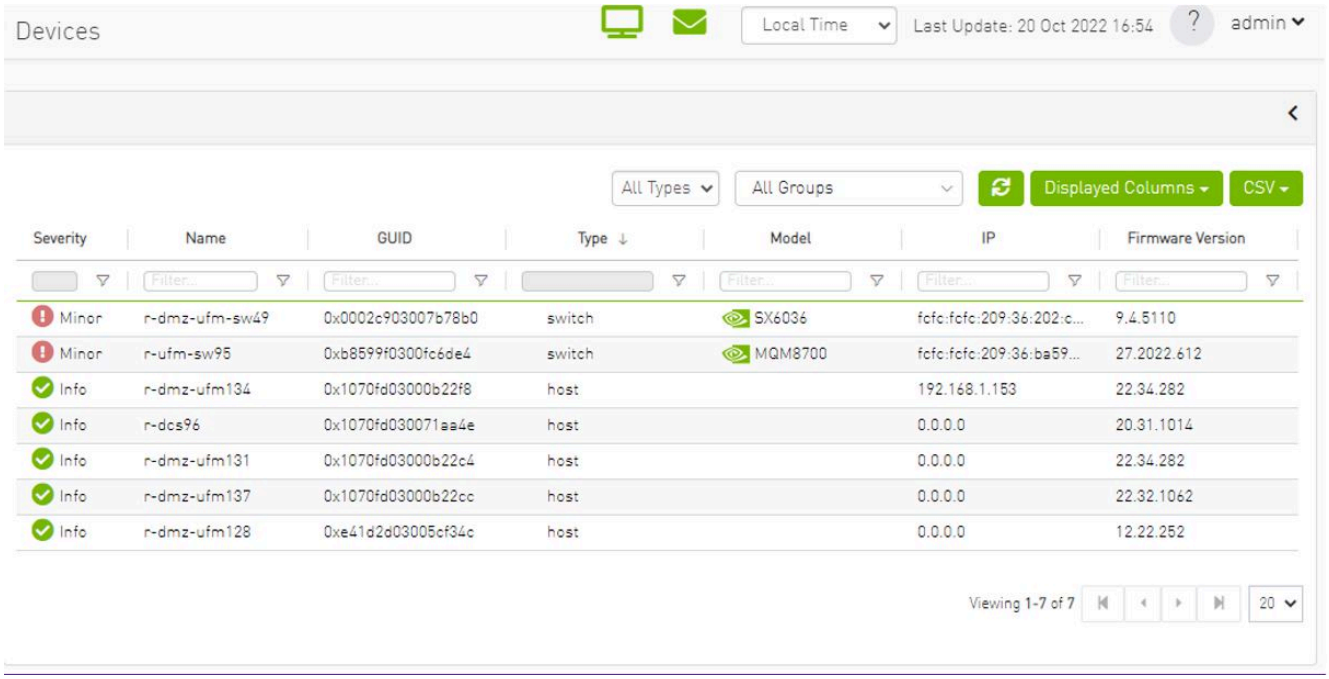
The UFM **Managed Elements** window allows you to obtain information on the fabric physical elements, such as devices, ports and cables.

> **Note**
>
> All information provided in a tabular format in UFM web UI can be exported into a CSV file.

- Devices Window
- Ports Window
- Virtual Ports Window
- Unhealthy Ports Window
- Cables Window
- Groups Window
- Inventory Window
- PKeys Window
- HCAs Window

Devices Window

The Devices window shows data pertaining to the physical devices in a tabular format.



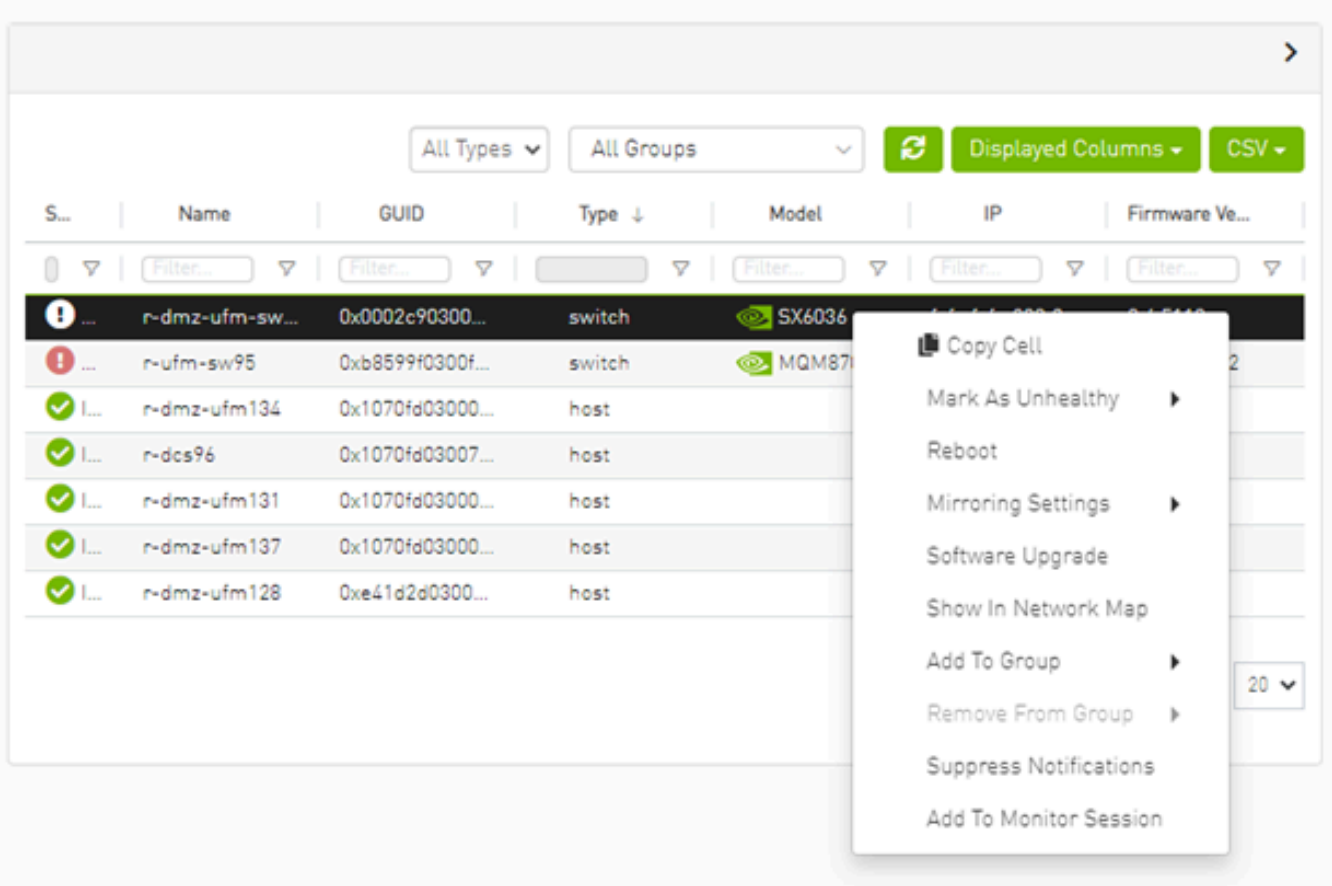
Devices Window Data

Data Type	Description
Health	Health of the device reflecting the highest alarm severity. Please refer to the Health States table.
Name	Name of the device **Note** If UFM Agent is running on a device, the following icon will appear next to the device name:
GUID	System GUID of the device
Type	Type of the device: switch, node, IB router, and getaway
IP	IP address of the device
Vendor	The vendor of the device
Firmware Version	The firmware version installed on the device

Health States

Icon	Name	Description
	Normal	Information/notification displayed during normal operating state or a normal system event.
	Critical	Critical means that the operation of the system or a system component fails.
	Minor	Minor reflects a problem in the fabric with no failure.
	Warning	Warning reflects a low priority problem in the fabric with no failure. A warning is asserted when an event exceeds a predefined threshold.

A right-click on the device name displays a list of actions that can be performed on it.



Devices Actions

Action	Description
Firmware Upgrade	Perform a firmware upgrade on the selected device
Firmware Reset	Reboot the device. This action is only applicable to unmanaged hosts (servers).
Set Node Description	Configure a description to this node
Collect System Dump	Collect the system dump log for a specific device
Add to Group	Add the selected device to a devices group
Remove from Group	Remove the selected device from a devices group
Suppress Notifications	Suppress all event notifications for the device
Add to Monitor Session	Configure and activate host monitoring

Action	Description
Show in Network Map	Move to Zoom In tab in network map and add the selected device to filter list

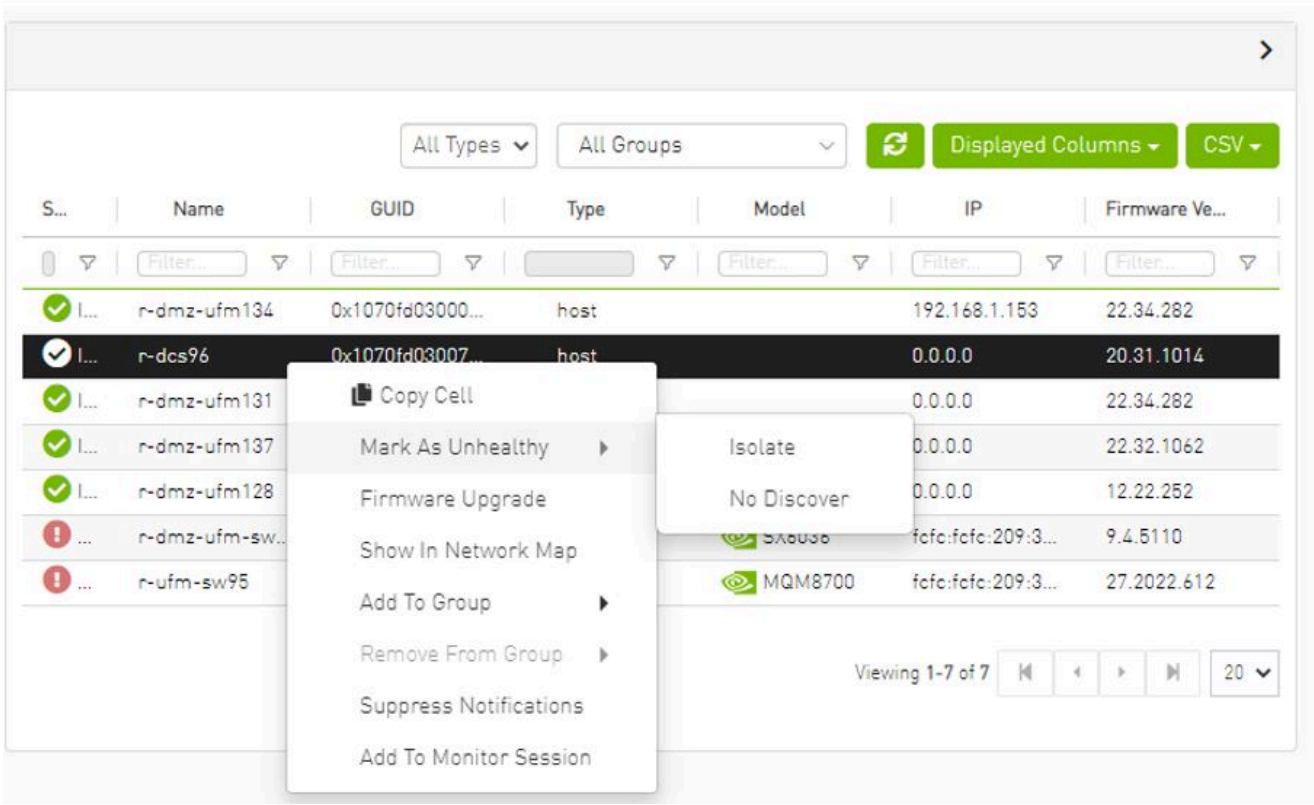
> **Note**
>
> Collecting system dump for hosts, managed by UFM, is available only for hosts which are set with a valid IPv4 address and installed with MLNX_OFED.

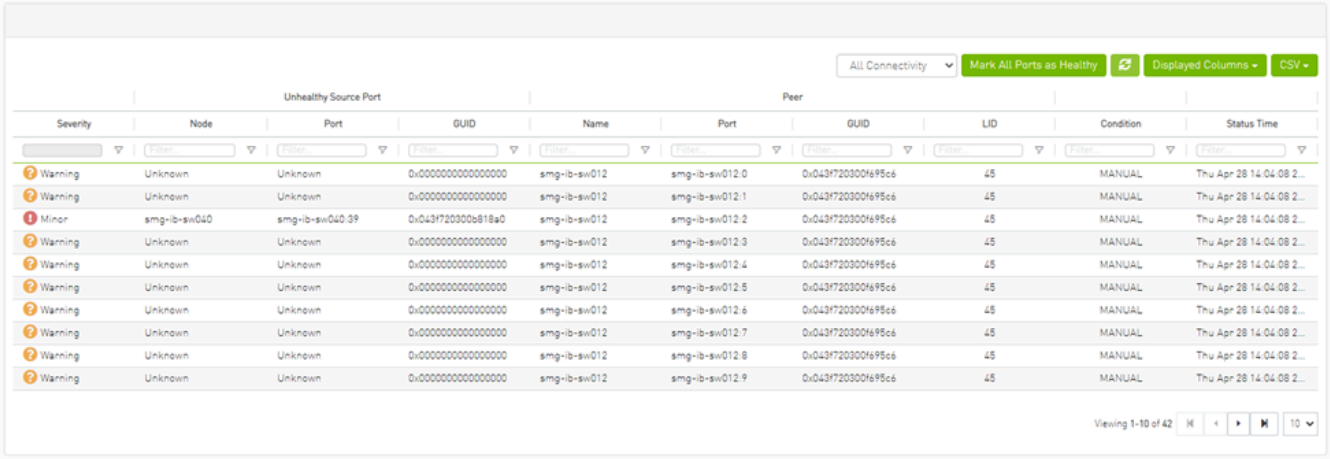
Mark Device as Unhealthy

From the Devices table, it is possible to mark devices as healthy or unhealthy using the context menu (right-click).

There are two options for marking a device as unhealthy:

- Isolate
- No Discover





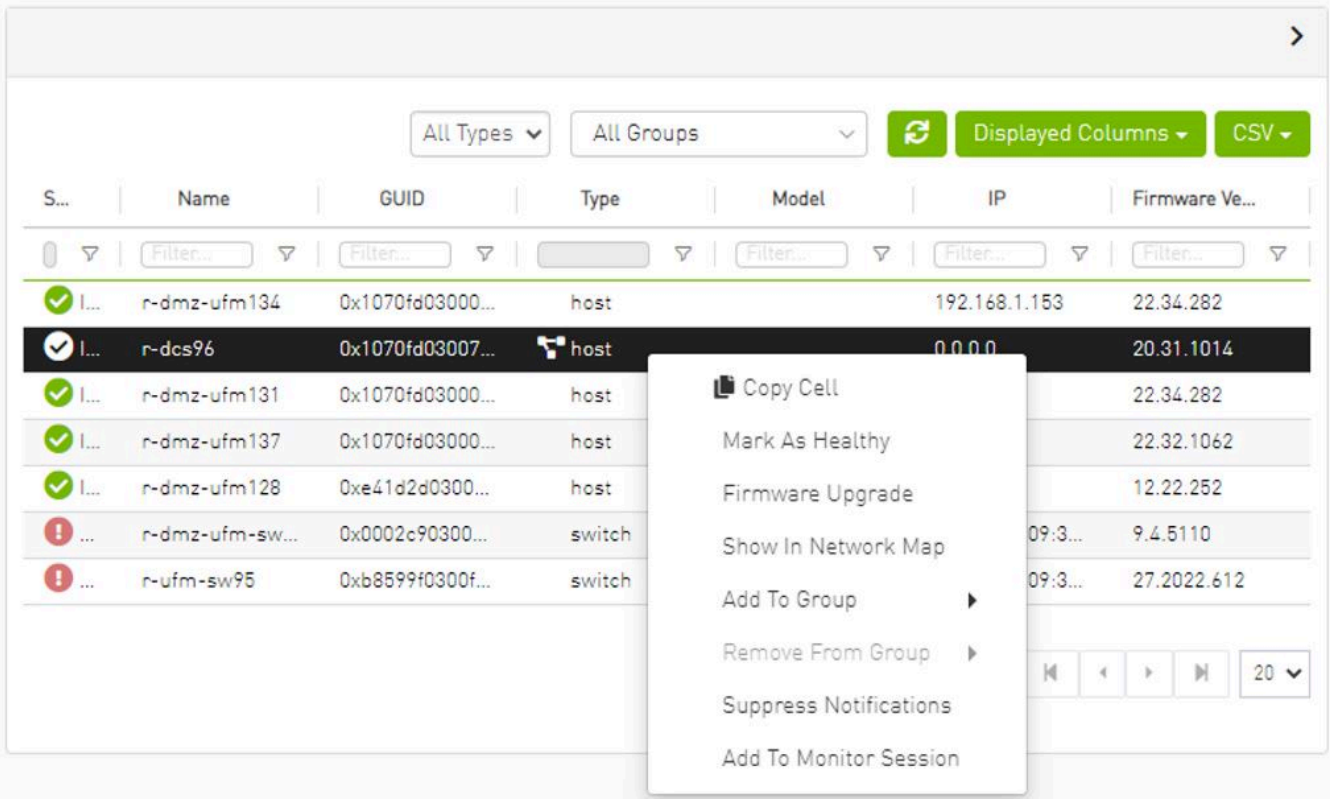
Server: `conf/opensm/opensm-health-policy.conf` content:

```
0xe41d2d030003e3b0 34 UNHEALTHY isolate
0xe41d2d030003e3b0 19 UNHEALTHY isolate
0xe41d2d030003e3b0 3 UNHEALTHY isolate
0xe41d2d030003e3b0 26 UNHEALTHY isolate
0xe41d2d030003e3b0 0 UNHEALTHY isolate
0xe41d2d030003e3b0 27 UNHEALTHY isolate
0xe41d2d030003e3b0 7 UNHEALTHY isolate
0xe41d2d030003e3b0 10 UNHEALTHY isolate
0xe41d2d030003e3b0 11 UNHEALTHY isolate
0xe41d2d030003e3b0 22 UNHEALTHY isolate
0xe41d2d030003e3b0 18 UNHEALTHY isolate
0xe41d2d030003e3b0 29 UNHEALTHY isolate
0xe41d2d030003e3b0 8 UNHEALTHY isolate
0xe41d2d030003e3b0 5 UNHEALTHY isolate
0xe41d2d030003e3b0 17 UNHEALTHY isolate
0xe41d2d030003e3b0 23 UNHEALTHY isolate
0xe41d2d030003e3b0 15 UNHEALTHY isolate
0xe41d2d030003e3b0 24 UNHEALTHY isolate
0xe41d2d030003e3b0 2 UNHEALTHY isolate
0xe41d2d030003e3b0 16 UNHEALTHY isolate
0xe41d2d030003e3b0 13 UNHEALTHY isolate
0xe41d2d030003e3b0 14 UNHEALTHY isolate
0xe41d2d030003e3b0 32 UNHEALTHY isolate
0xe41d2d030003e3b0 33 UNHEALTHY isolate
0xe41d2d030003e3b0 35 UNHEALTHY isolate
0xe41d2d030003e3b0 20 UNHEALTHY isolate
0xe41d2d030003e3b0 21 UNHEALTHY isolate
0xe41d2d030003e3b0 28 UNHEALTHY isolate
0xe41d2d030003e3b0 1 UNHEALTHY isolate
0xe41d2d030003e3b0 9 UNHEALTHY isolate
0xe41d2d030003e3b0 4 UNHEALTHY isolate
0xe41d2d030003e3b0 31 UNHEALTHY isolate
0xe41d2d030003e3b0 30 UNHEALTHY isolate
0xe41d2d030003e3b0 36 UNHEALTHY isolate
0xe41d2d030003e3b0 12 UNHEALTHY isolate
```

```
0xe41d2d030003e3b0 25 UNHEALTHY isolate
0xe41d2d030003e3b0 6 UNHEALTHY isolate
```

`/opt/ufm/files/log/opensm-unhealthy-ports.dump` content:

Mark Device as Healthy



Server `/opt/ufm/files/conf/opensm/opensm-health-policy.conf` content:

```
0xe41d2d030003e3b0 15 HEALTHY
0xe41d2d030003e3b0 25 HEALTHY
0xe41d2d030003e3b0 35 HEALTHY
0xe41d2d030003e3b0 0 HEALTHY
0xe41d2d030003e3b0 11 HEALTHY
0xe41d2d030003e3b0 21 HEALTHY
0xe41d2d030003e3b0 28 HEALTHY
0xe41d2d030003e3b0 7 HEALTHY
0xe41d2d030003e3b0 17 HEALTHY
0xe41d2d030003e3b0 14 HEALTHY
0xe41d2d030003e3b0 24 HEALTHY
0xe41d2d030003e3b0 34 HEALTHY
0xe41d2d030003e3b0 3 HEALTHY
0xe41d2d030003e3b0 10 HEALTHY
0xe41d2d030003e3b0 20 HEALTHY
0xe41d2d030003e3b0 31 HEALTHY
0xe41d2d030003e3b0 6 HEALTHY
0xe41d2d030003e3b0 16 HEALTHY
0xe41d2d030003e3b0 27 HEALTHY
0xe41d2d030003e3b0 2 HEALTHY
0xe41d2d030003e3b0 13 HEALTHY
0xe41d2d030003e3b0 23 HEALTHY
0xe41d2d030003e3b0 33 HEALTHY
0xe41d2d030003e3b0 30 HEALTHY
0xe41d2d030003e3b0 9 HEALTHY
0xe41d2d030003e3b0 19 HEALTHY
0xe41d2d030003e3b0 26 HEALTHY
0xe41d2d030003e3b0 36 HEALTHY
0xe41d2d030003e3b0 5 HEALTHY
0xe41d2d030003e3b0 12 HEALTHY
0xe41d2d030003e3b0 22 HEALTHY
0xe41d2d030003e3b0 32 HEALTHY
0xe41d2d030003e3b0 1 HEALTHY
0xe41d2d030003e3b0 8 HEALTHY
0xe41d2d030003e3b0 18 HEALTHY
```

```
0xe41d2d030003e3b0 29 HEALTHY
0xe41d2d030003e3b0 4 HEALTHY
```

`/opt/ufm/files/log/opensm-unhealthy-ports.dump` content:

```
# NodeGUID, PortNum, NodeDesc, PeerNodeGUID, PeerPortNum,
PeerNodeDesc, {BadCond1, BadCond2, ...}, timestamp
```

Upgrading Software and Firmware for Hosts and Externally Managed Switches

Software/Firmware Upgrade via FTP

Software and firmware upgrade over FTP is enabled by the UFM Agent. UFM invokes the Software/Firmware Upgrade procedure locally on switches or on hosts. The procedure copies the new software/firmware file from the defined storage location and performs the operation on the device. UFM sends the set of attributes required for performing the software/firmware upgrade to the agent.

The attributes are:

- File Transfer Protocol – default FTP
    - The Software/Firmware upgrade on InfiniScale III ASIC-based switches supports FTP protocol for transmitting files to the local machine.
    - The Software/Firmware upgrade on InfiniScale IV-based switches and hosts supports TFTP and protocols for transmitting files to the local machine.
- IP address of file-storage server
- Path to the software/firmware image location

  The software/firmware image files should be placed according to the required structure under the defined image storage location. Please refer to section Devices Window.
- File-storage server access credentials (User/Password)

In-Band Firmware Upgrade

You can perform in-band firmware upgrades for externally managed switches and HCAs. This upgrade procedure does not require the UFM Agent or IP connectivity, but it does require current PSID recognition. Please refer to section PSID and Firmware Version In-Band Discovery. This feature requires that the Mellanox Firmware Toolkit (MFT), which is included in the UFM package, is installed on the UFM server. UFM uses flint from the MFT for in-band firmware burning.

Before upgrading, you must create the firmware repository on the UFM server under the directory `/opt/ufm/files/userdata/fw/`. The subdirectory should be created for each PSID and one firmware image should be placed under it. For example:

```
/opt/ufm/files/userdata/fw/
         MT_0D80110009
                        fw-ConnectX2-rel-2_9_1000-MHQH29B-
XTR_A1.bin
         MT_0F90110002
                        fw-IS4-rel-7_4_2040-MIS5023Q_A1-A5.bin
```

Directory Structure for Software or Firmware Upgrade Over FTP

Before performing a software or firmware upgrade, you must create the following directory structure for the upgrade image. The path to the `<ftp user home>/<path>/` directory should be specified in the upgrade dialog box.

```
<ftp user home>/<path>/
         InfiniScale3 - For anafa based switches
Software/Firmware upgrade images
                        voltaire_fw_images.tar – firmware image
file
                        ibswmpr-<s/w version>.tar – software
image file
        InfiniScale4 - For InfiniScale IV based switches
Software/Firmware upgrade images
                        firmware_2036_4036.tar – Firmware image
file
                        upgrade_2036_4036.tgz – Software image
file
        OFED /* For host SW upgrade*/
                        OFED-<OS label>.tar.bz2
        <PSID>* – For host FW upgrade
                        fw_update.img
```

The <PSID> value is extracted from the `mstflint` command:

```
mstflint -d <device> q
```

The device is extracted from the `lspci` command. For example:

```
# lspci
06:00.0 InfiniBand: Mellanox Technologies MT25208 InfiniHost III
Ex
# mstflint -d 06:00.0 q | grep PSID
PSID: VLT0040010001
```

PSID and Firmware Version In-Band Discovery

The device PSID and device firmware version are required for in-band firmware upgrade and for the correct functioning of Subnet Manager plugins, such as Congestion Control Manager and Lossy Configuration Management. For most devices, UFM discovers this information and displays it in the Device Properties pane. The PSID and the firmware version are discovered by the Vendor-specific MAD.

By default, the gv.cfg file value for event_plugin_option is set to (null). This means that the plugin is disabled and opensm does not send MADs to discover devices' PSID and FW version. Therefore, values for devices' PSID and FW version are taken from `ibdiagnet` output (section NODES_INFO).

The below is an example of the default value:

```
event_plugin_options = (null)
```

To enable the vendor-specific discovery by opemsm, in the gv.cfg configuration file, change the value of event_plugin_option to (--vendinfo –m 1), as shown below:

```
event_plugin_options = --vendinfo –m 1
```

If the value is set to –vendinfo –m 1, the data should be supplied by opensm, and in this case the `ibdiagnet` output is ignored.

> **Note**
>
> In some firmware versions, the information above is currently not available.

Switch Management IP Address Discovery

From NVIDIA switch FM version 27.2010.3942 and up, NVIDIA switches support switch management IP address discovery using MADs. This information can be retrieved as part of `ibdiagnet` run (`ibdiagnet` output), and assigned to discover switches in UFM.

There is an option to choose the IP address of which IP protocol version that is assigned to the switch: IPv4 or IPv6.

The discovered_switch_ip_protocol key, located in the gv.cfg file in section [FabricAnalysys], is set to 4 by default. This means that the IP address of type IPv4 is assigned to the switch as its management IP address. In case this value is set to 6, the IP address of type IPv6 is assigned to the switch as its management IP address.

After changing the discover_switch_ip_protocol value in gv.cfg, the UFM Main Model needs to be restarted for the update to take effect. The discovered IP addresses for switches are not persistent in UFM – every UFM Main Model restarts the values of management IP address which is assigned from the `ibdiagnet` output.

Upgrading Server Software

The ability to update the server software is applicable only for hosts (servers) with the UFM Agent.

To upgrade the software:

1. Select a device.
2. From the right-click menu, select Software Update.
3. Enter the parameters listed in the following table.

Parameter	Description
Protocol	Update is performed via FTP protocol
IP	Enter the host IP
Path	Enter the parent directory of the FTP directory structure for the Upgrade image. The path should not be an absolute path and should not contain the first slash (/) or trailer slash.
User	Name of the host username

Parameter	Description
Password	Enter the host password

4. Click Submit to save your changes.

Upgrading Firmware

You can upgrade firmware over FTP for hosts and switches that are running the UFM Agent, or you can perform an in-band upgrade for externally managed switches and HCAs.

Before you begin the upgrade ensure that the new firmware version is in the correct location. For more information, please refer to section In-Band Firmware Upgrade.

To upgrade the firmware:

1. Select a host or server.
2. From the right-click menu, select Firmware Upgrade.
3. Select protocol In Band.
4. For upgrade over FTP, enter the parameters listed in the following table.

Parameter	Description
IP	Enter device IP
Path	Enter the parent directory of the FTP directory structure for the Upgrade image. The path should not be an absolute path and should not contain the first slash (/) or trailer slash.
Username	Name of the host username
Password	Enter the host password

5. Click submit to save your changes.

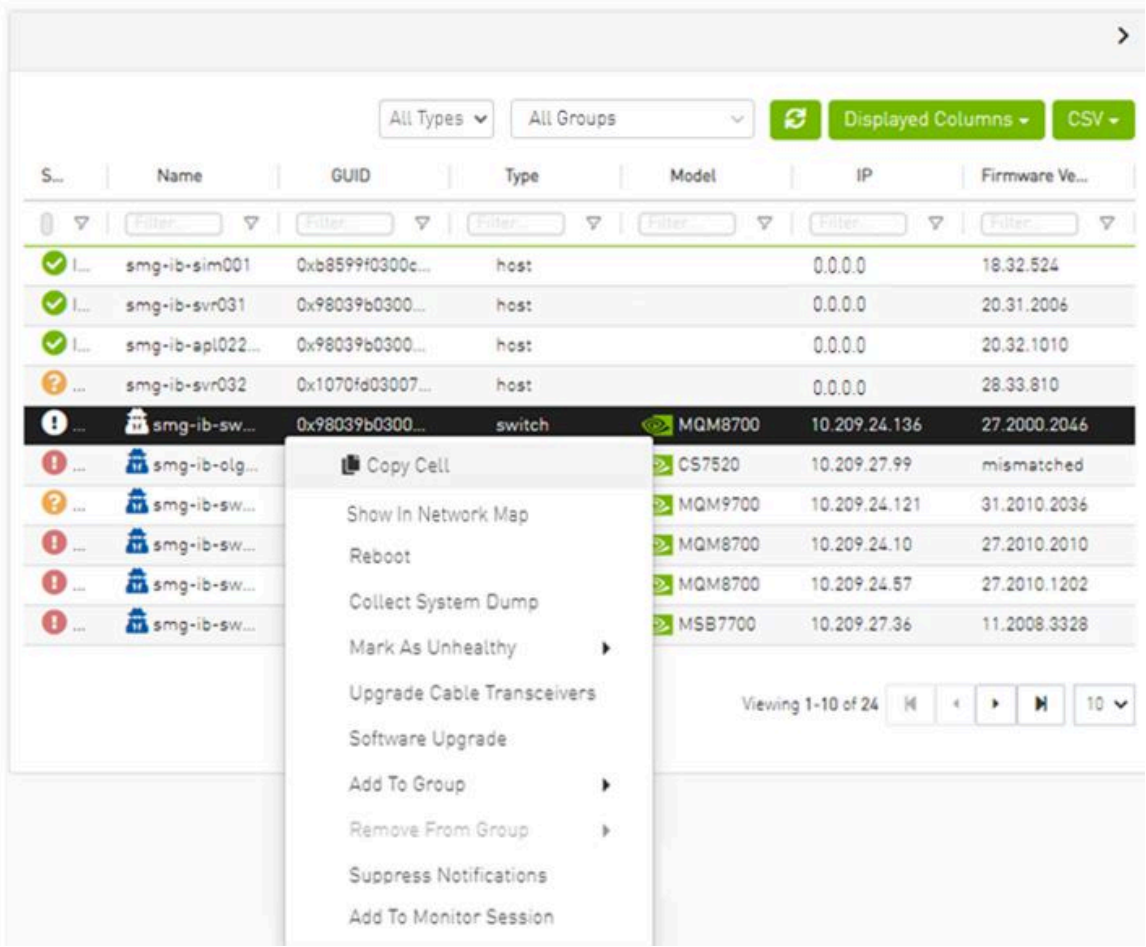
> **Note**
>
> The firmware upgrade takes effect only after the host or externally managed switch is restarted.

Upgrade Cables Transceivers Firmware Version

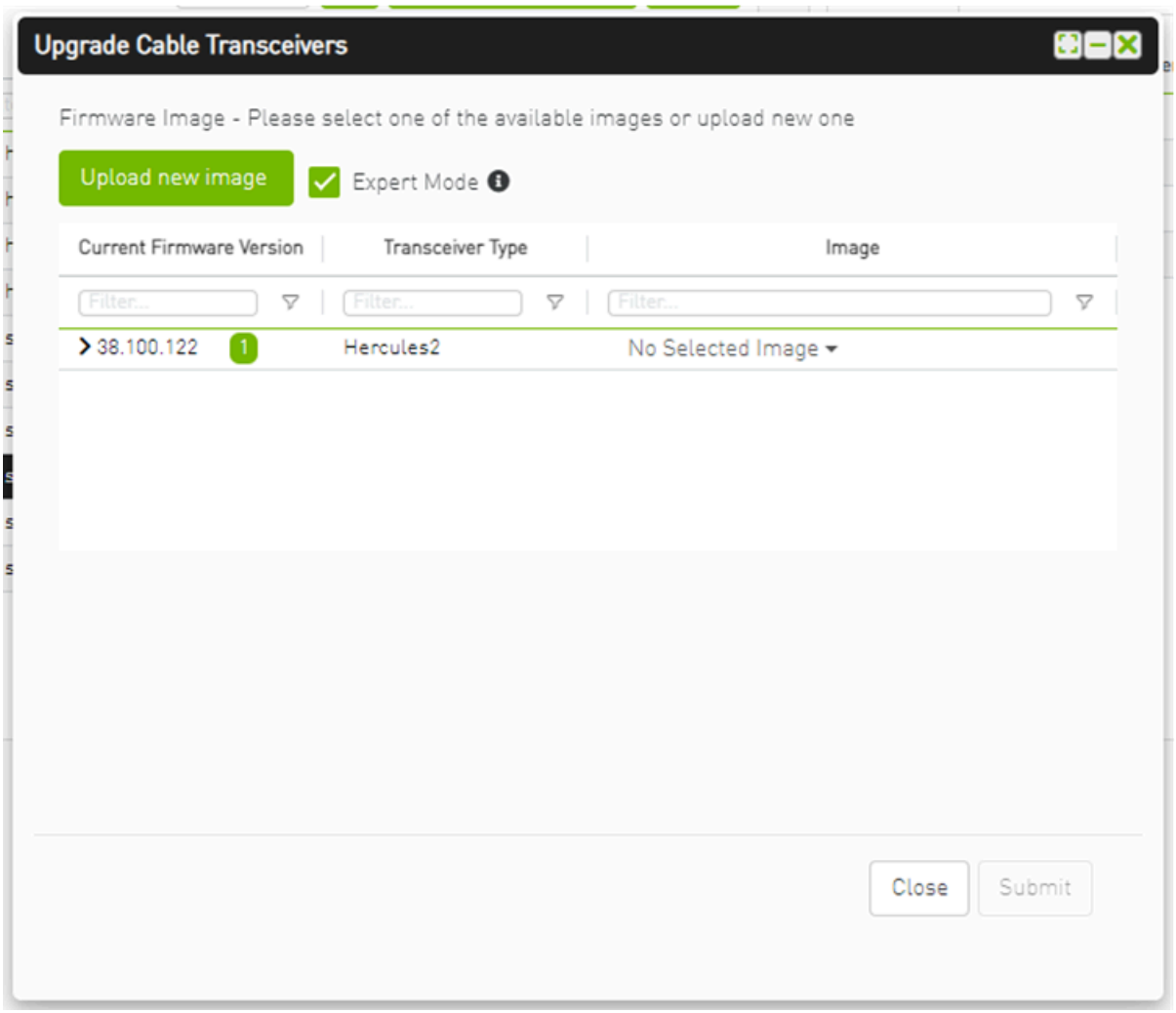
The main purpose of this feature is to add support for burning of multiple cables transceiver types on multiple devices using linkx tool which is part of flint. This needs to be done from both ends of the cable (switch and HCA/switch).

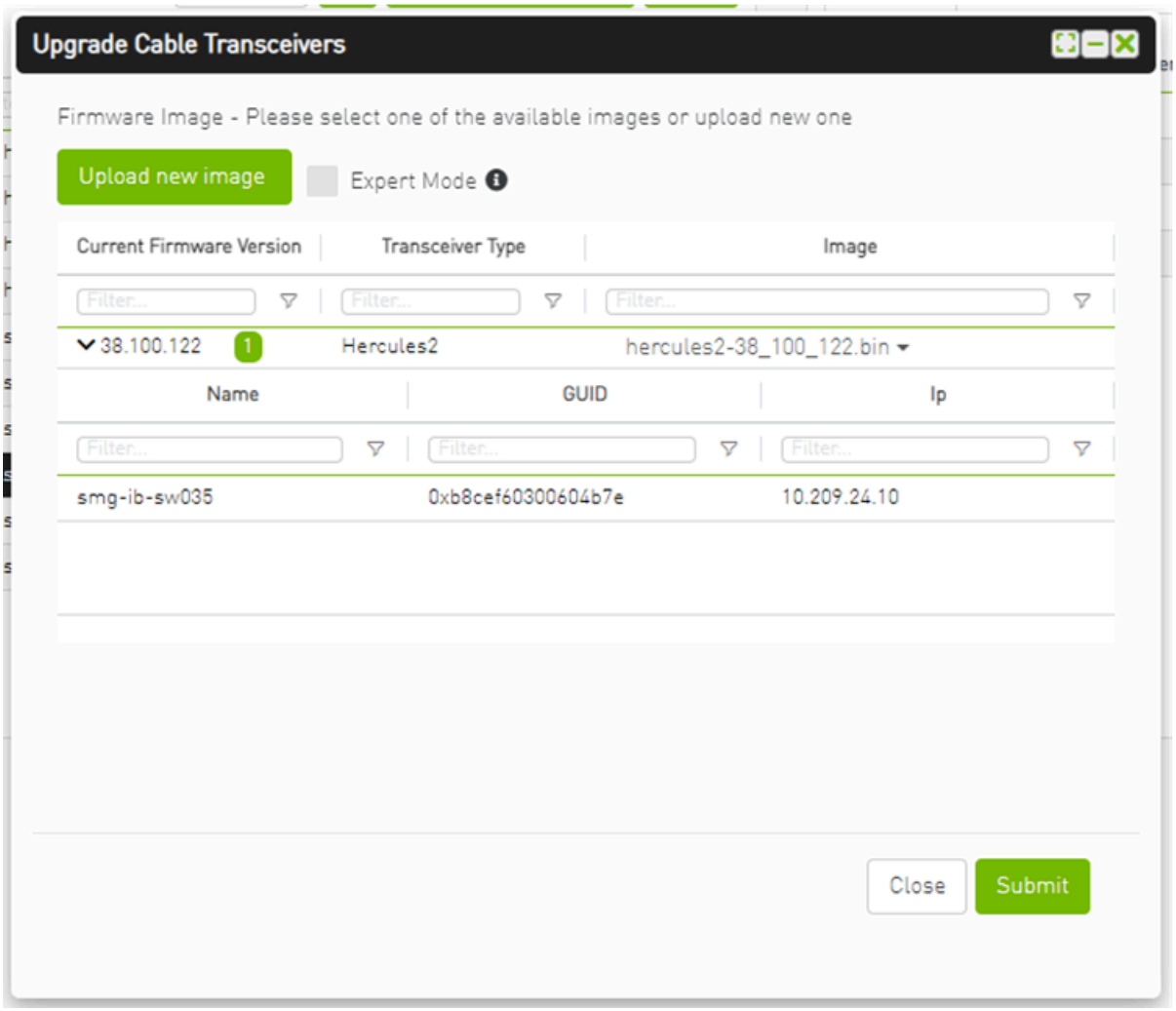
To upgrade cables transceivers FW version:

1. Navigate to managed elements page
2. select the target switches and click on Upgrade Cable Transceivers option

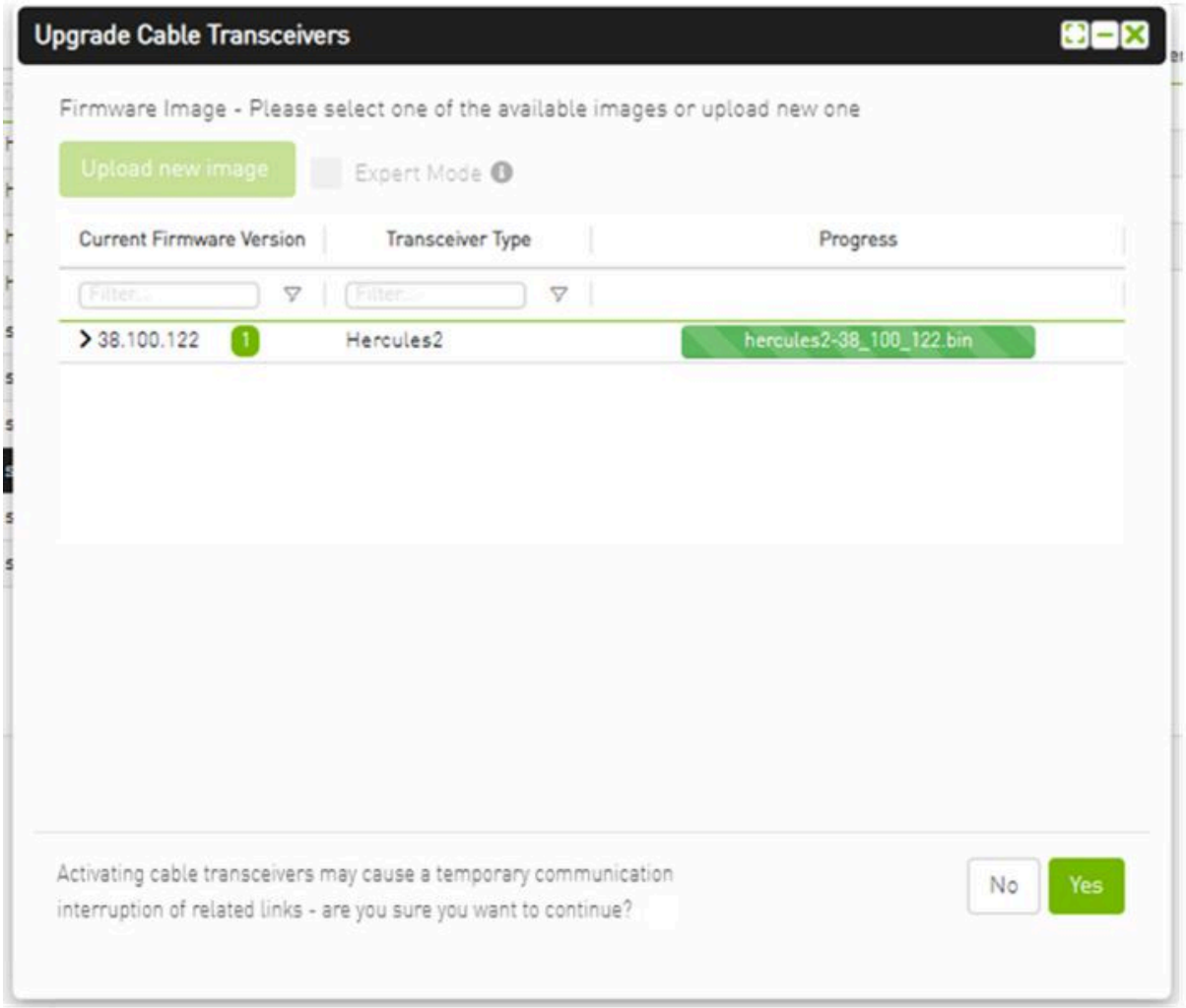


3. A model will be shown containing list of the active firmware versions for the cables of the selected switches, besides the version number, a badge will show the number of matched switches:


Upgrade Cable Transceivers
Firmware Image - Please select one of the available images or upload new one
Upload new image
Expert Mode
Current Firmware Version
Transceiver Type
Image
Filter...
38.100.122
1
Hercules2
No Selected Image
Close
Submit

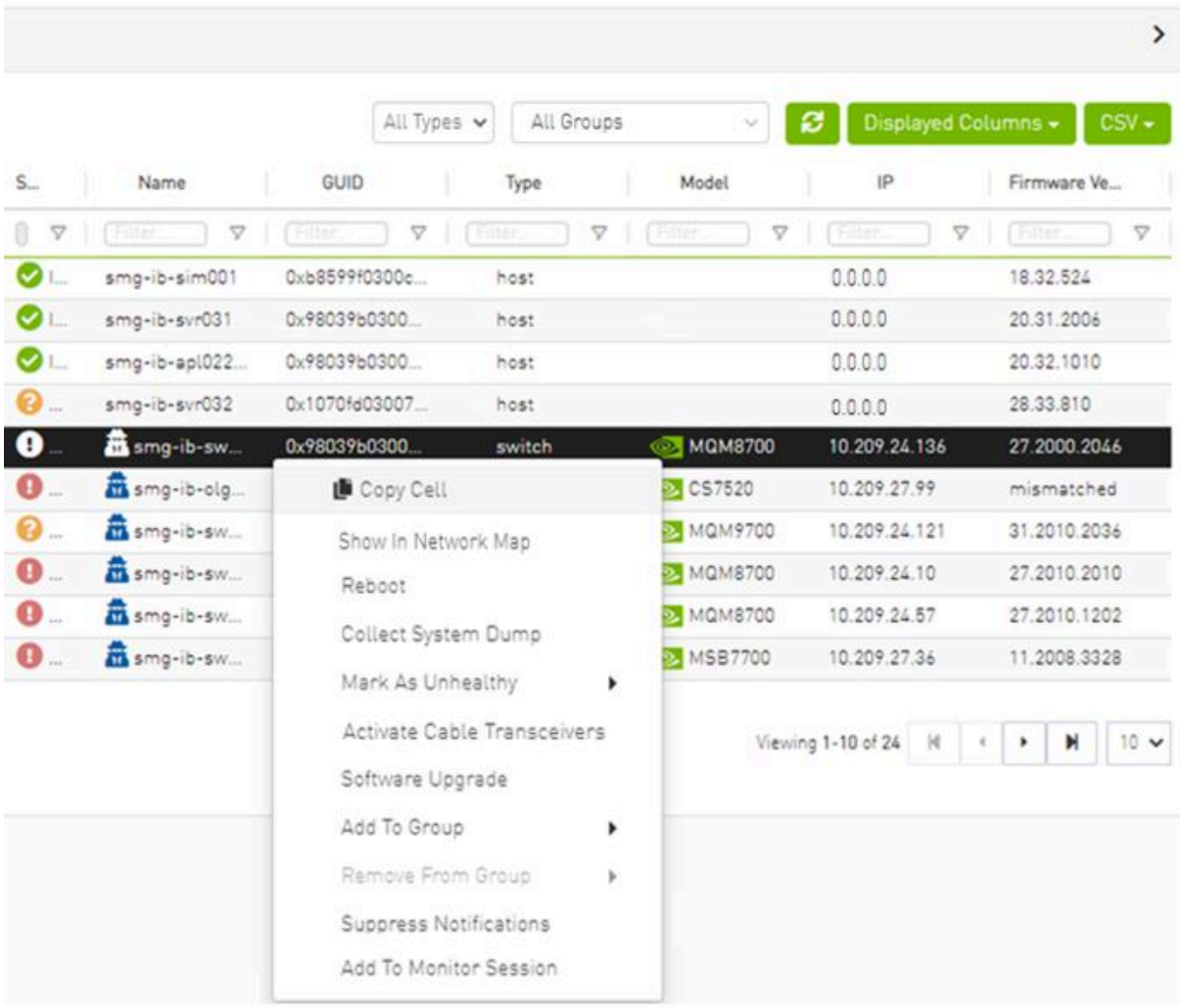


4. After the user clicks Submit, the GUI will start sending the selected binaries with the relevant switches sequentially, and a model with a progress bar will be shown (this model can be minimized):



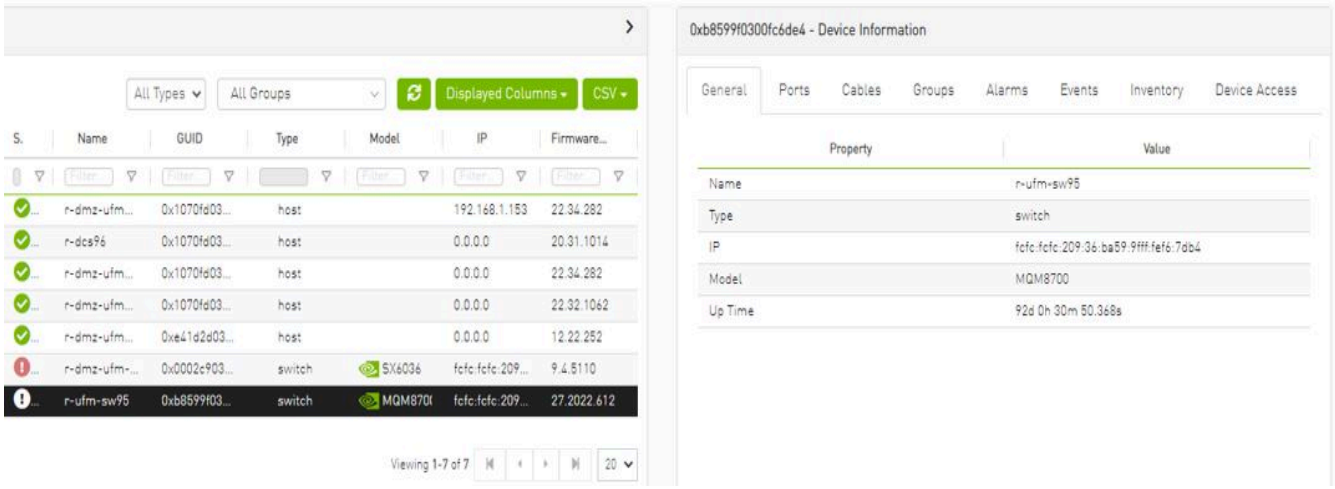
5. After the whole action is completed successfully, you will be able to see the following message at the model bottom The upgrade cable transceivers completed successfully, do you want to activate it? by clicking the yes button it will run a new action on all the burned devices to activate the new uploaded binary image.

6. Another option to activate burned cables transceivers you can go to the Groups page and right click on the predefined Group named **Devices Pending FW Transceivers Reset** or you can right click on the upgraded device from managed element page and select Activate cable Transceivers action.



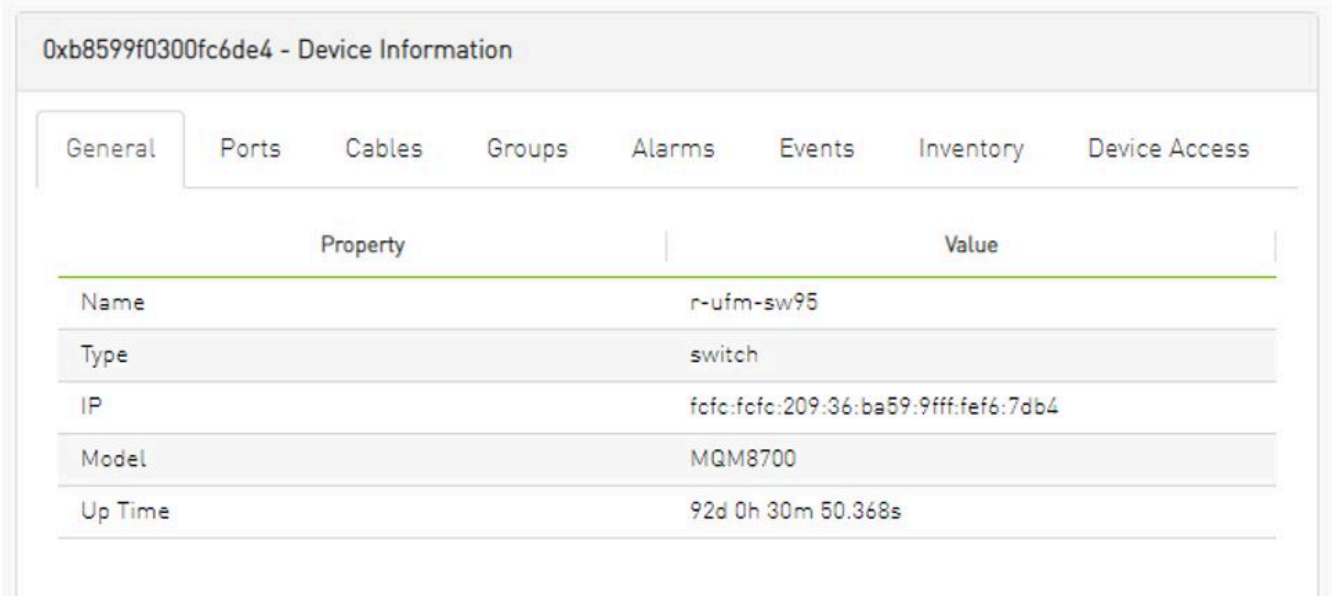
Device Information Tabs

Selecting a device from the Devices table reveals the **Device Information** table on the right side of the screen. This table provides information on the device's ports, cables, groups, events, alarms, inventory, and device access.



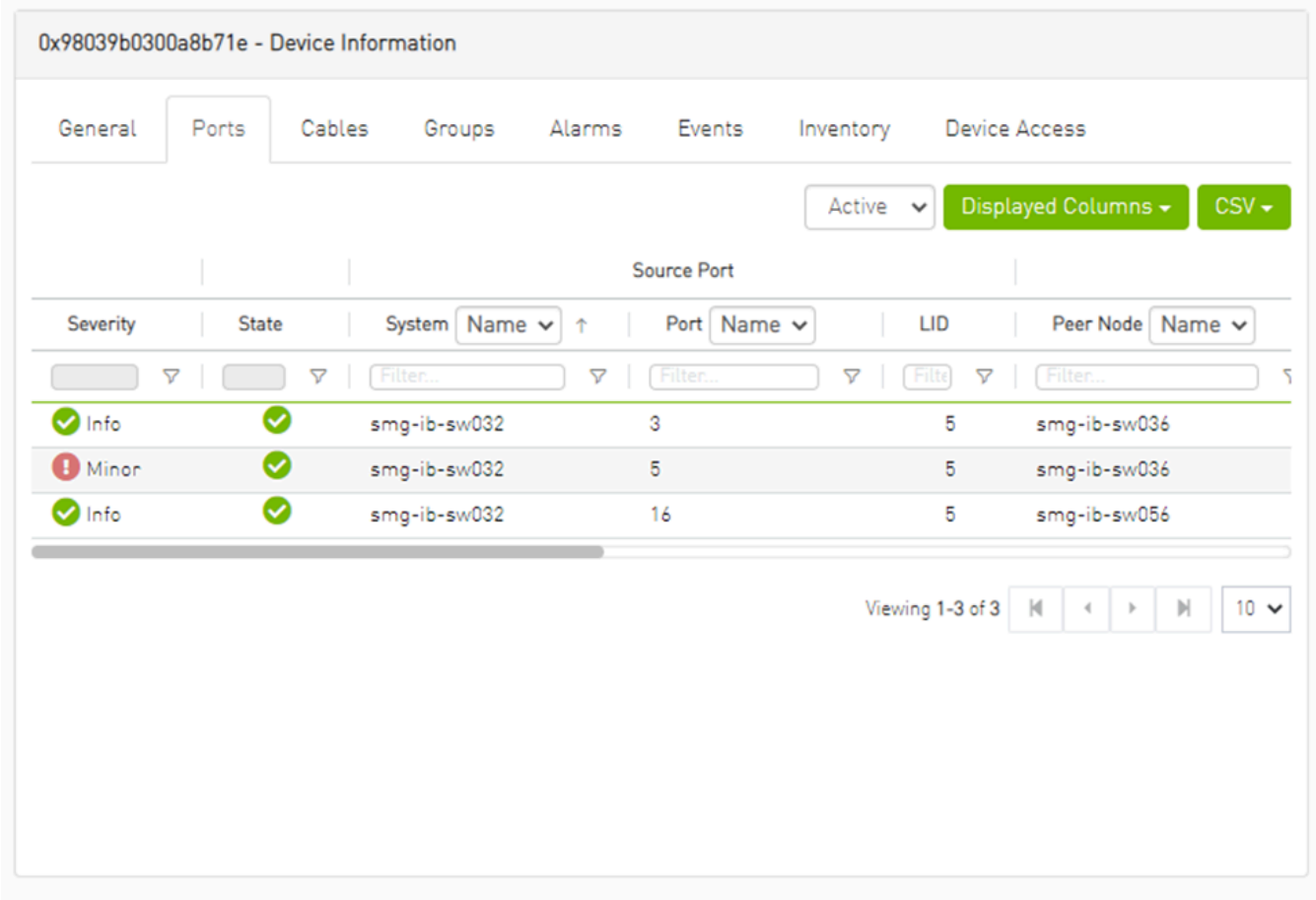
General Tab

Provides general information on the selected device.



Ports Tab

This tab provides a list of the ports connected to this device in a tabular format.



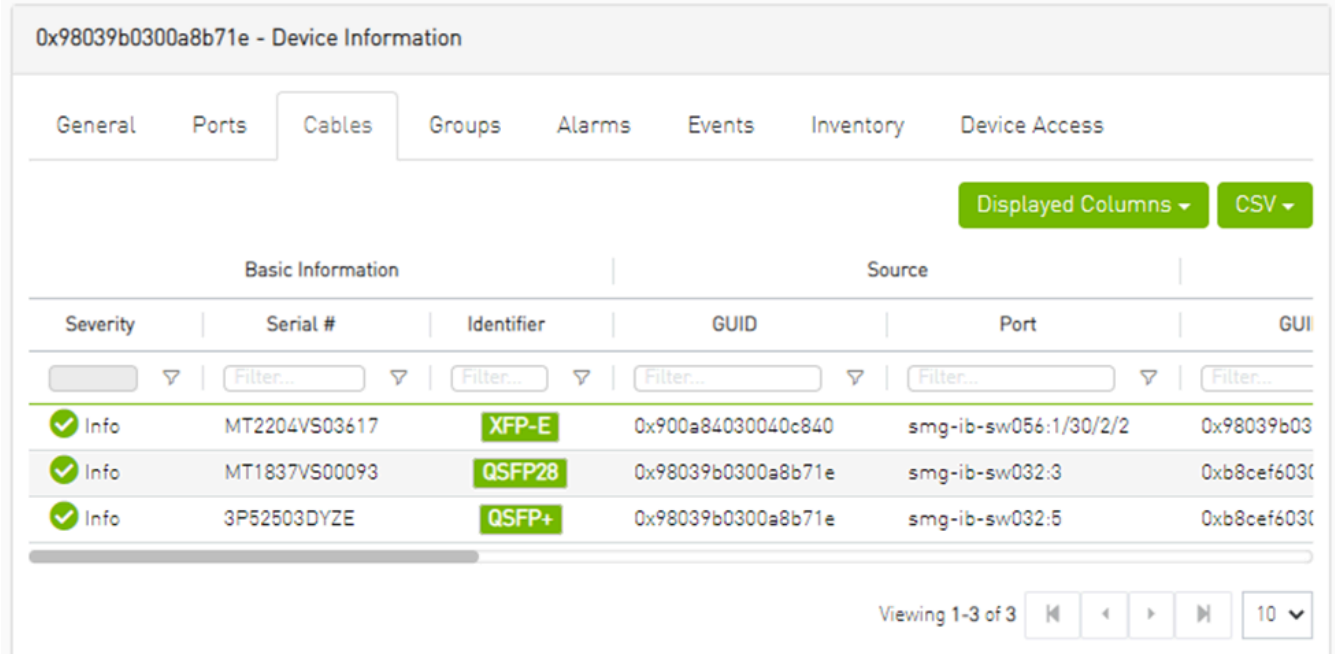
Ports Data

Data Type	Description
Port Number	The number of ports on device.
Node	The node name/GUID/IP that the port belongs to. Note that you can choose the node label (name/GUID/IP) using the drop-down menu available above the Ports data table.
Health	Health of the port reflecting the highest alarm severity. Please refer to the Health States table.
State	Indicates whether the port is connected (active or inactive).
LID	The local identifier (LID) of the port.
MTU	Maximum Transmission Unit of the port.

Data Type	Description
Speed QDR FDR EDR	Lists the highest value of active, enabled and supported speeds in icons indicating their status: • Dark green – active speed • Light green – enabled speed • Grey – supported yet disabled speed
Width 1X 2X 4X	Lists the highest value of active, enabled and supported widths in icons indicating their status: • Dark green – active width • Light green – enabled width • Grey – supported yet disabled width
Peer	The GUID of the device the port is connected to.
Peer Port	The name of the port that is connected to this port.

Cables Tab

This tab provides a list of the cables connected to this device in a tabular format.

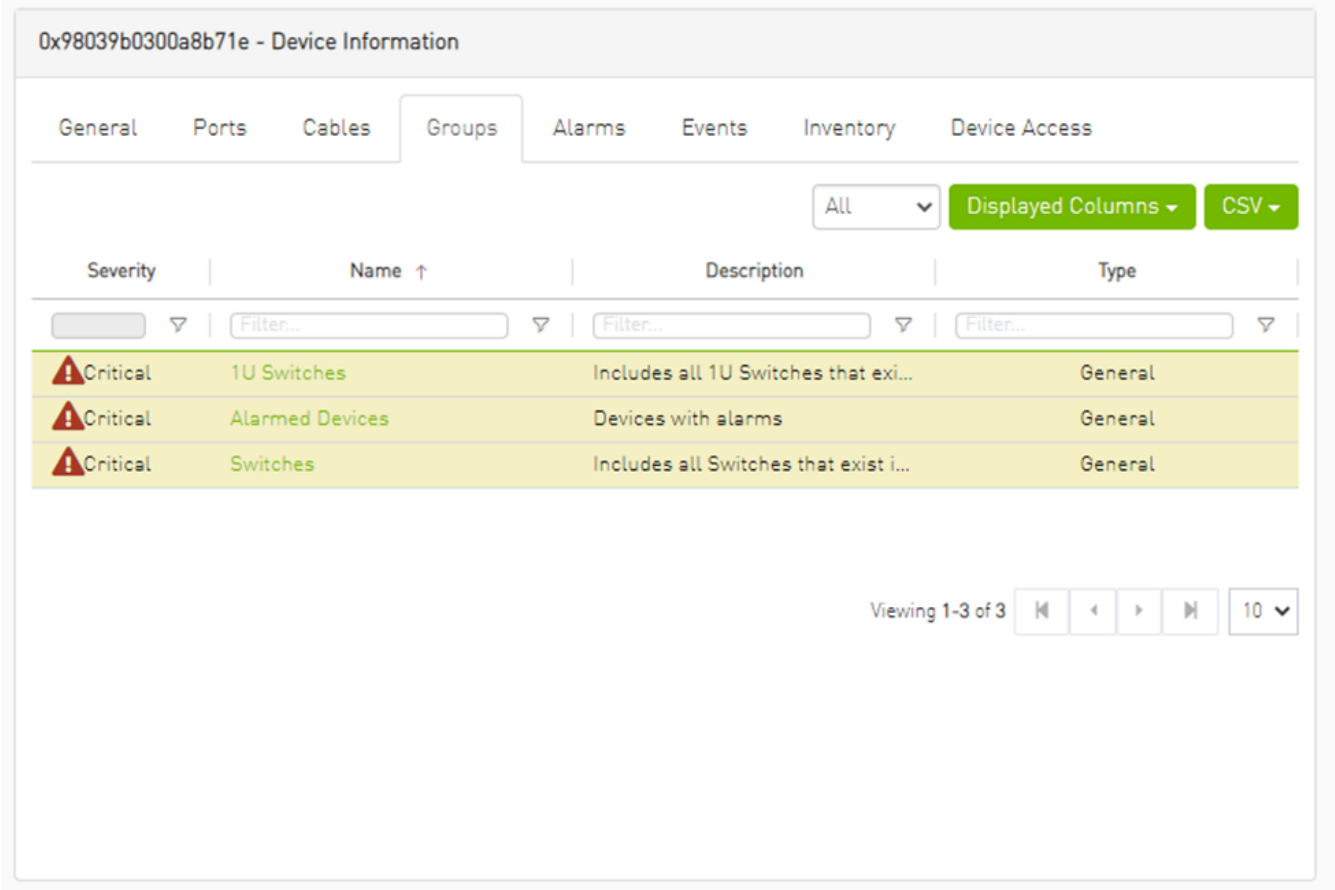


Cables Data

Data Type	Description
Basic Information	
Health	Health of the cable reflecting the highest alarm severity. Please refer to the Health States table.
Serial Number	Serial number of the cable.
Identifier	Identifier of the cable.
Source Port Information	
Source GUID	GUID of the source port the cable is connected to.
Source Port	The number of the source port the cable is connected to.
Destination Port Information	
Destination GUID	GUID of the destination port the cable is connected to.
Destination Port	The number of the destination port the cable is connected to.
Advanced Information	
Revision	Revision of the cable.
Link Width	The maximum link width of the cable.
Part Number	Part number of the cable.
Technology	The transmitting medium of the cable: copper/optical/etc.
Length	The cable length in meters.

Groups Tab

This tab provides a list of the groups to which the selected device belongs.

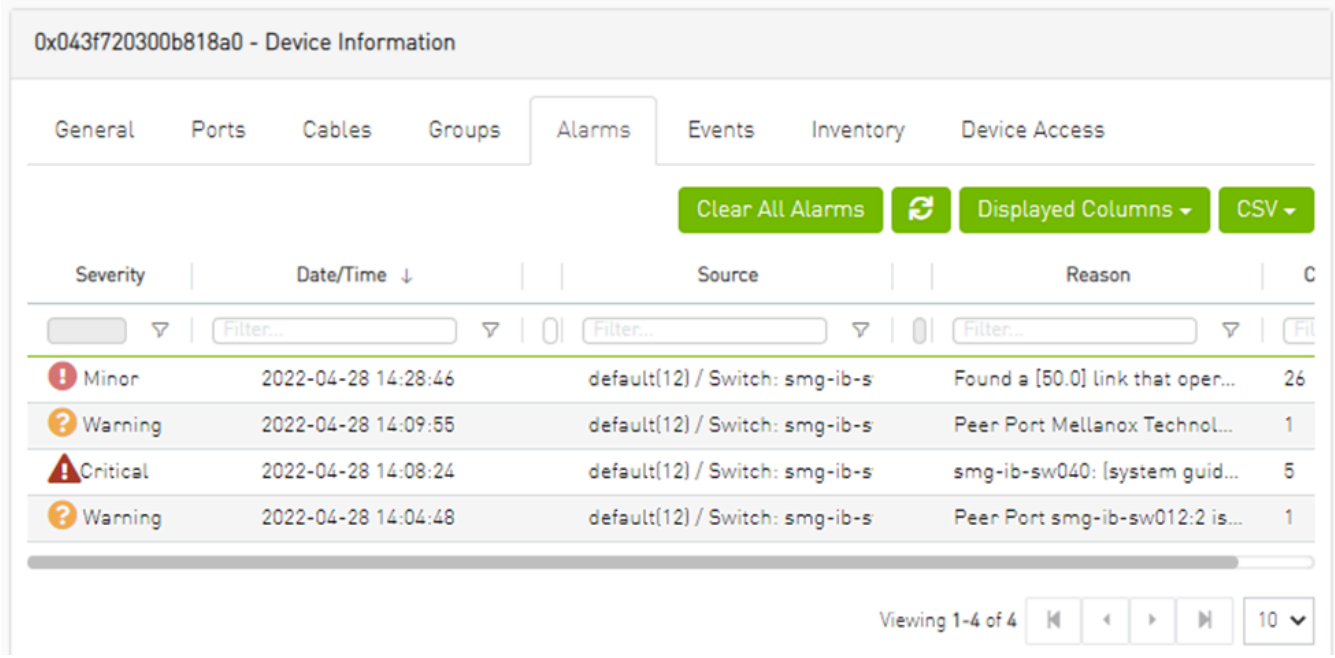


Groups Data

Data Type	Description
Severity	Aggregated severity level of the group (the highest severity level of all group members).
Name	Name of the group.
Description	Description of the group.
Type	Type of the group: General/Rack.

Alarms Tab

This tab provides a list of all UFM alarms related to the selected device.

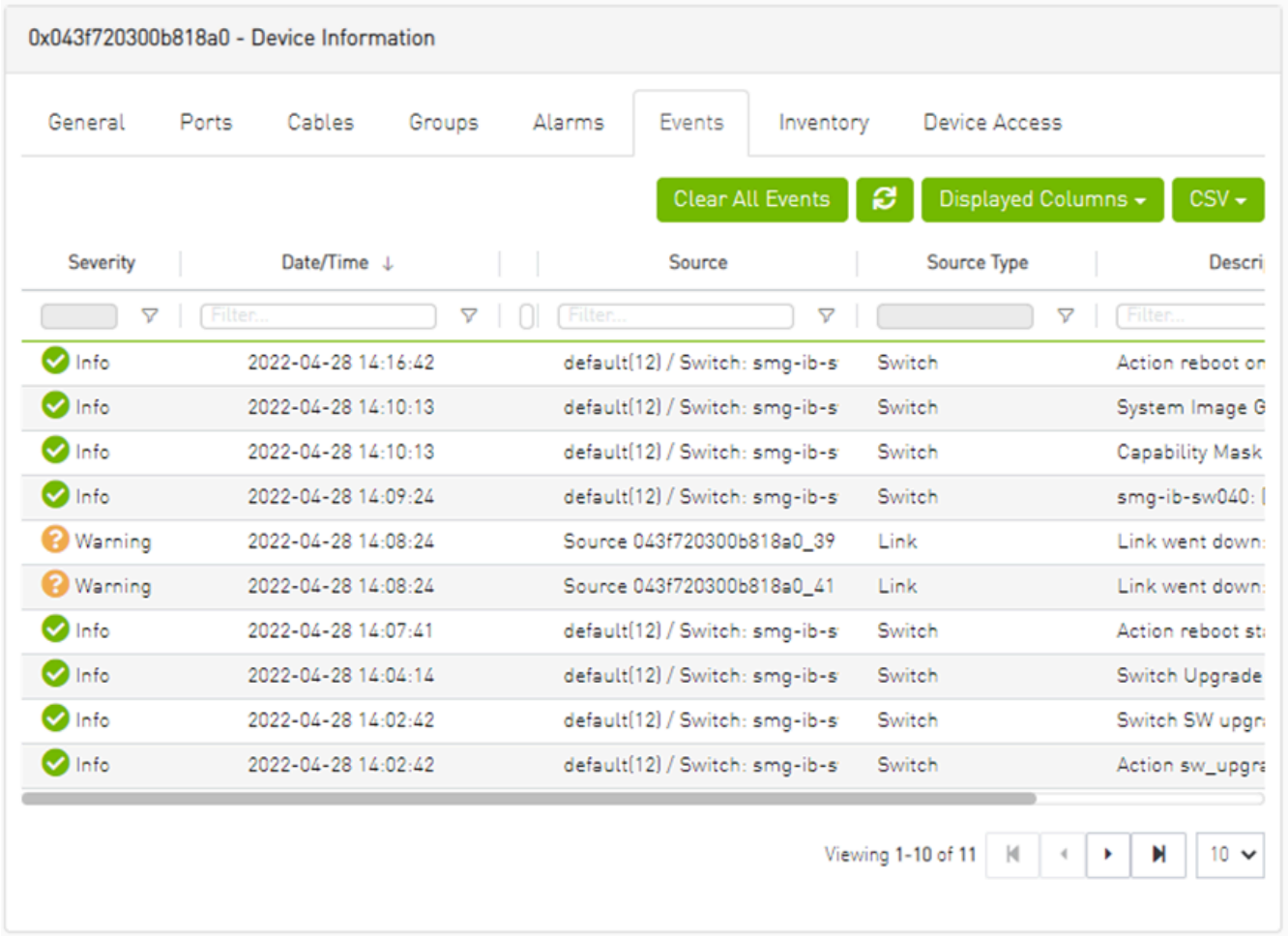


Alarms Data

Data Type	Description
Alarms ID	Alarm identifier.
Source	Source object (device/port) on which the alarm was triggered.
Severity	The severity of the alarm.
Description	Description of the alarm.
Date/Time	The time when the alarm was triggered.
Reason	Reason for the alarm.
Count	Number of instances that the alarm occurred on the related source object.

Events Tab

This tab provides a list of the UFM events that are related to the selected device.



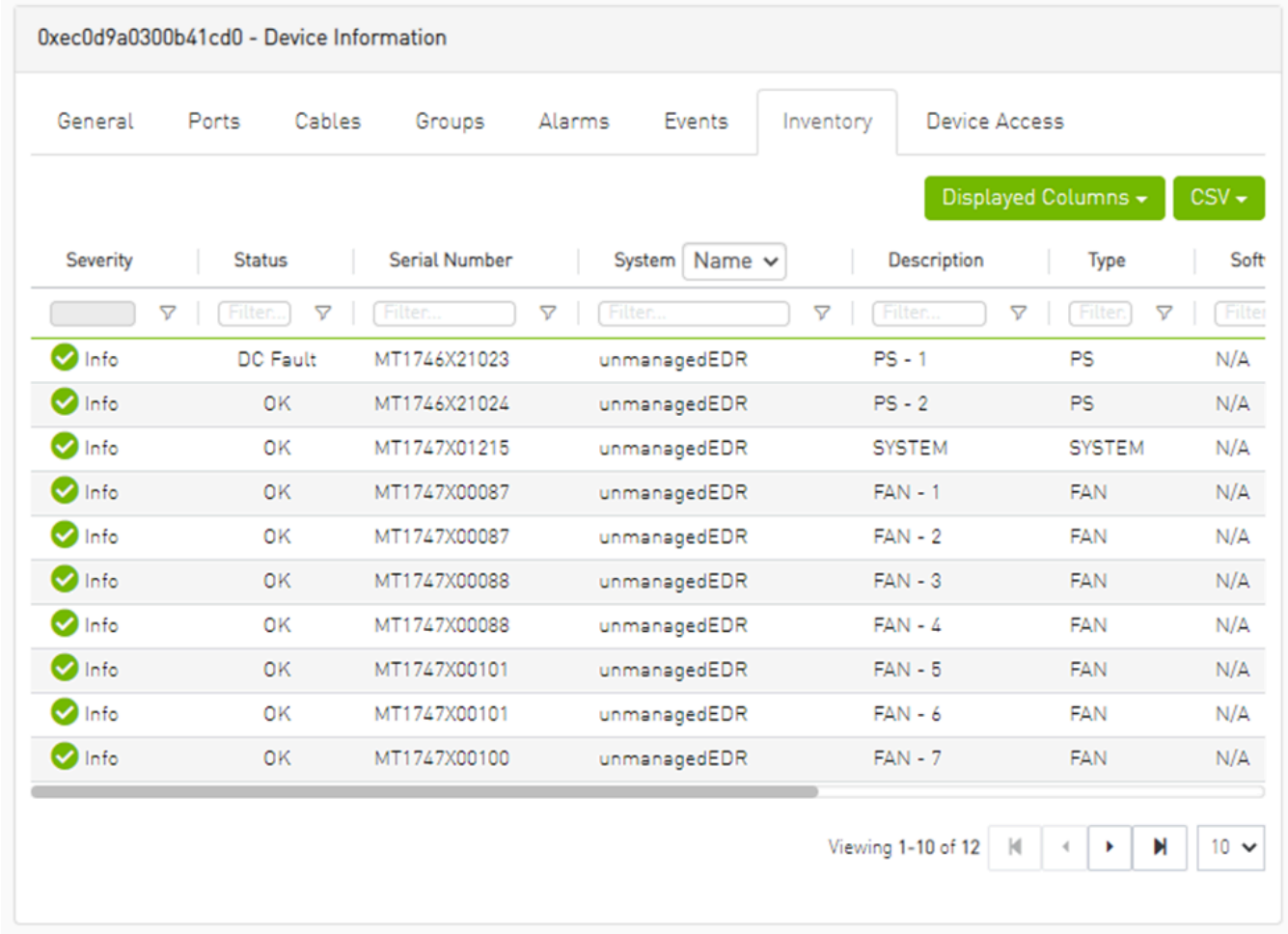
Events Data

Data Type	Description
Severity	Event severity – Info, Warning, Error, Critical or Minor.
Event Name	The name of the event.
Source	The source object (device/port) on which the event was triggered.
Date/Time	The time when the event was triggered.
Category	The category of the event indicated by icons. Hovering over the icon will display the category name.
Description	Description of the event. Full description can be displayed by hovering over the text.

Inventory Tab

This tab provides a list of the device's modules with information in a tabular format.

> **Note**
>
> This tab is available for switches only.



Inventory Data

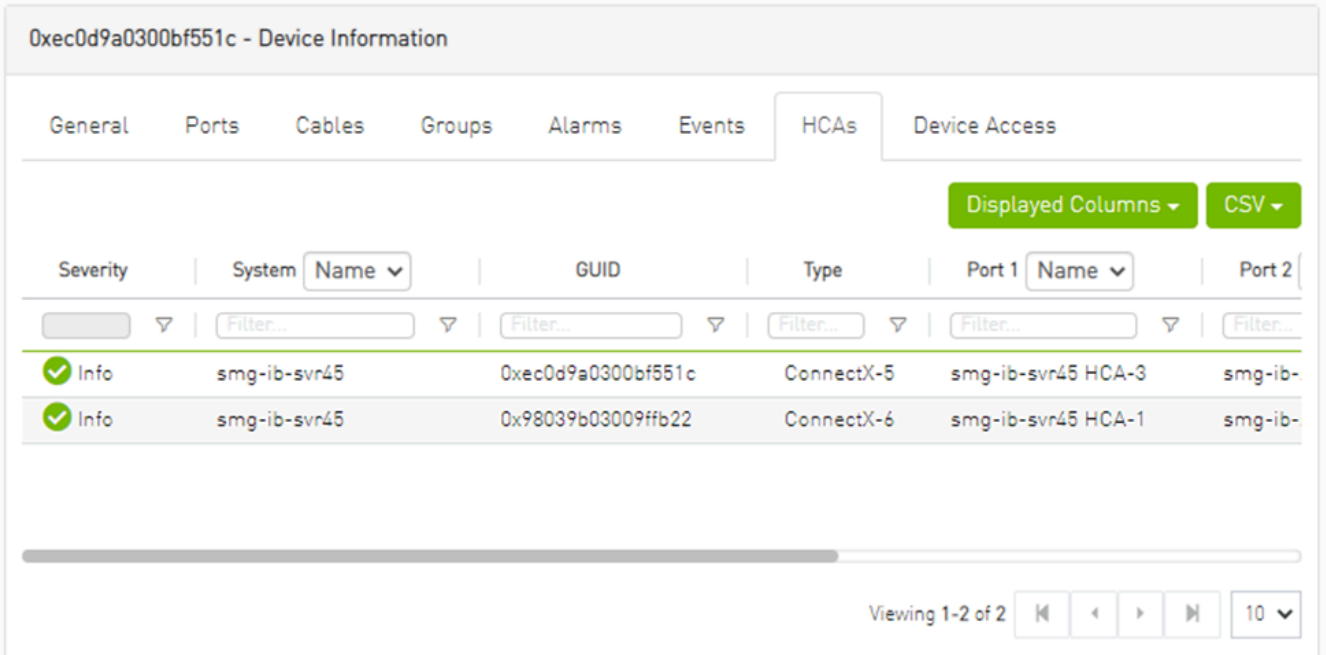
Data Type	Description
Health	Health of the module reflecting the highest alarm severity. Please refer to the Health States table.
Status	The module status.
Serial Number	Serial number of the module.

Data Type	Description
Name	Name of the device.
Description	Description of the module.
Type	Type of the module: spine/line/etc.
Firmware Version	Firmware version installed on the module.
Hardware Version	Hardware version of the module.
Temperature	Temperature of the module.

HCAs Tab

This tab provides a list of the device's HCAs with information in a tabular format.

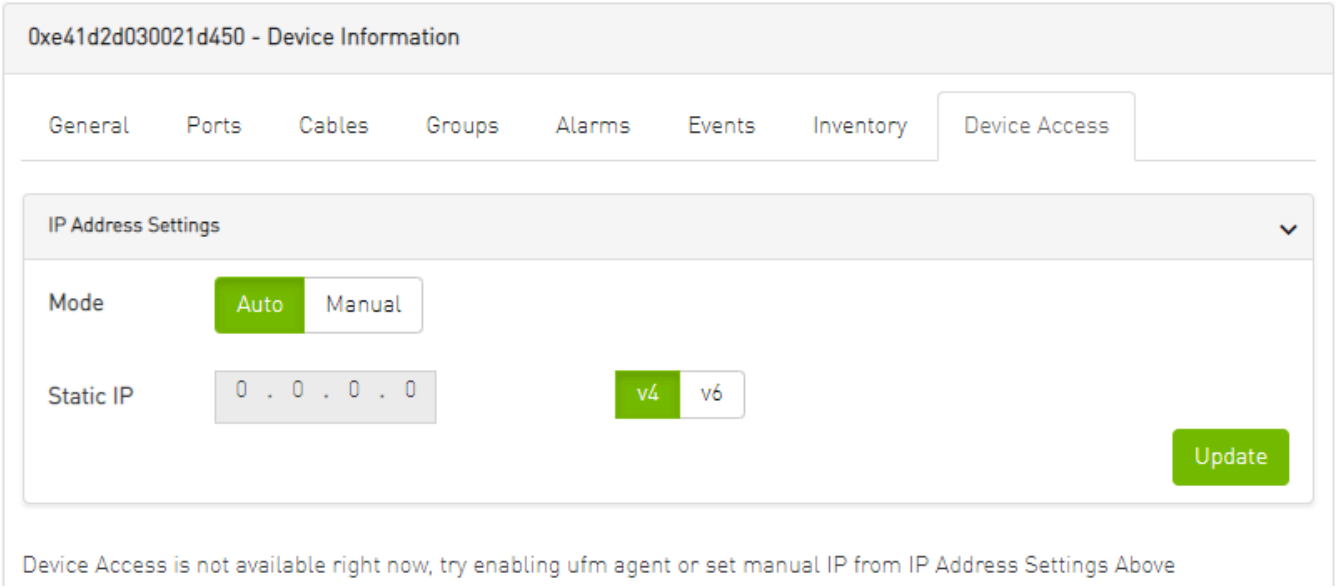
> **Note**
>
> This tab is available for hosts only.



Data Type	Description
Health	Health of the HCA reflecting the highest alarm severity. Please refer to the Health States table.
Name	HCA Index
GUID	HCA GUID
Type	HCA Type
Port GUID	HCA ports GUIDs
PSID	HCA PSID
FW Version	HCA firmware version

Device Access Tab

This tab allows for managing the access credentials of the selected device for remote accessibility. To be able to set access credentials for the device, a device IP must be set either by installing UFM Agent on the device, or by manually setting the IP under **IP Address Settings** (IP is now supported with v4 and v6).

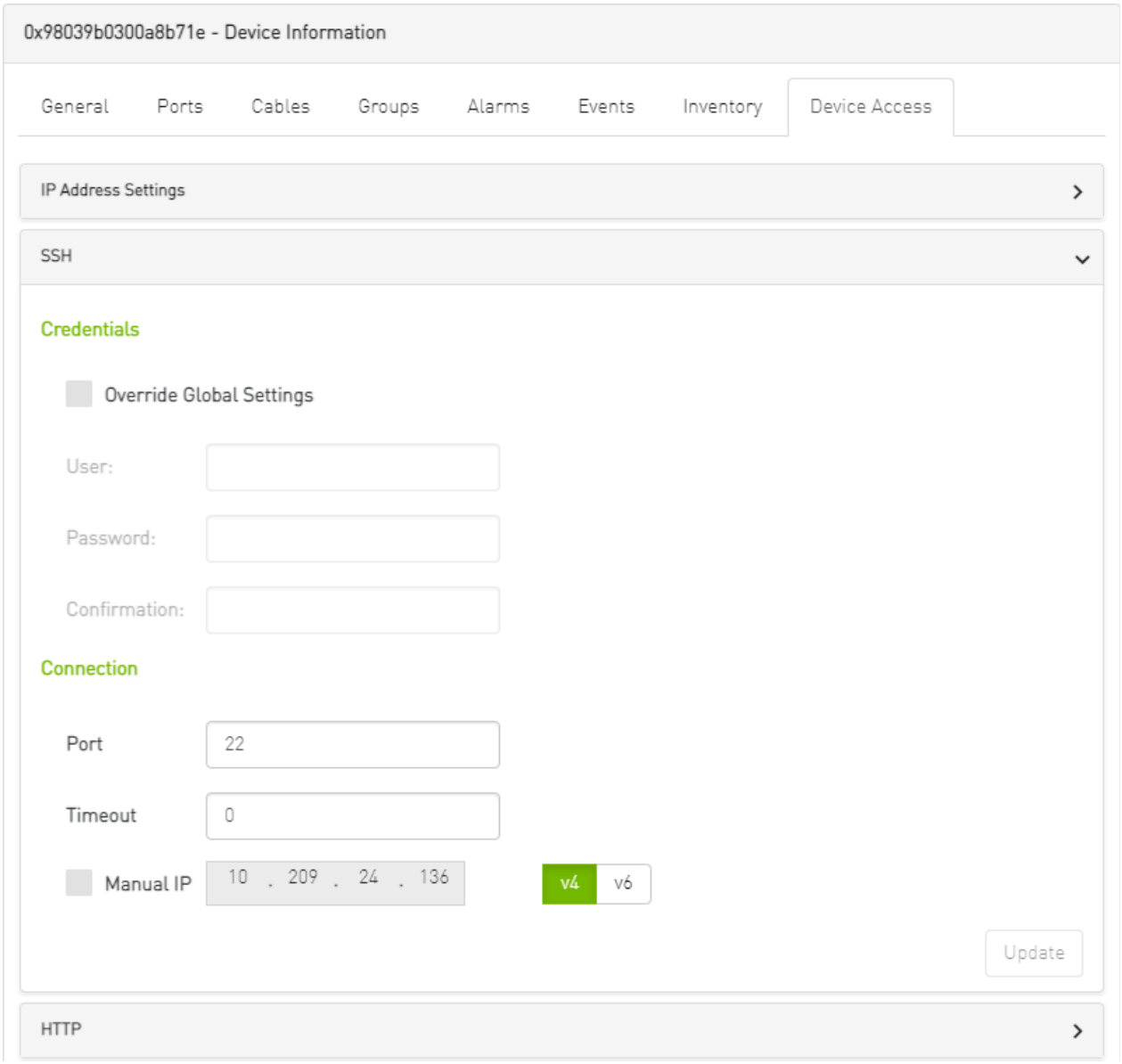


Note

After manually setting the IP address of NVIDIA® Mellanox® InfiniScale IV® and SwitchX® based switches, UFM will first validate the new IP before setting it.

To edit your device access credentials

1. Select the preferred protocol tab:
    - **SSH** – allows you to define the SSH parameters to open an SSH session on your device (available for **nodes** and **switches**)
    - **IPMI** – allows you to set the IPMI parameters to open an IPMI session on your device for remote power control (available for **nodes only**)
    - **HTTP** – allows you to define the HTTP parameters to open an HTTP session on your device (available for **switches only**)
2. Click **Update** to save your changes.

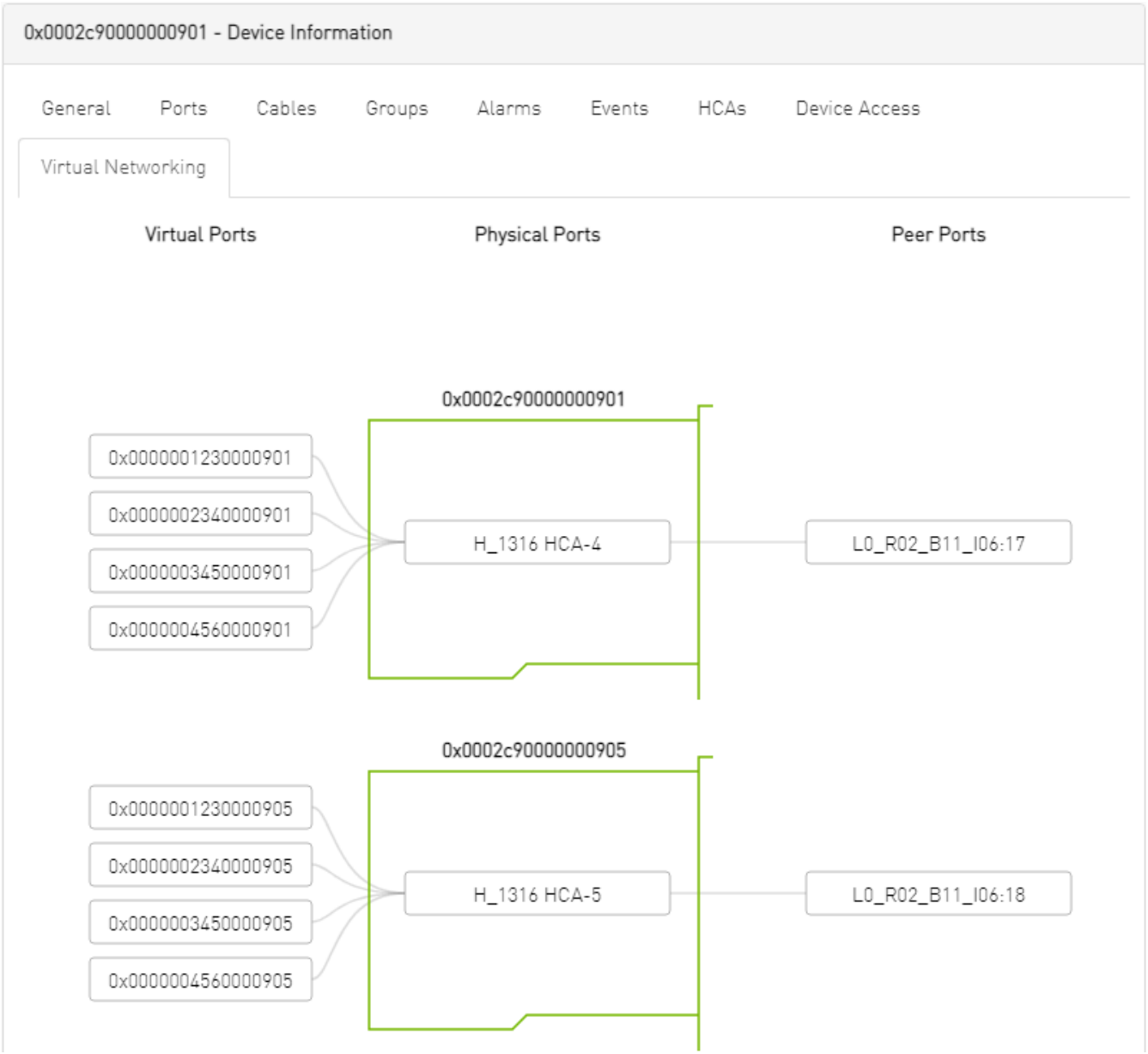


Device Access Credentials Parameters

Field	Description
User	Fill in or edit the computer user name.
Password	Enter the device password.
Confirmation	Enter the device password a second time to confirm.
Manual IP	Enter the device IP address (could be IPv4/IPv6).
Port	Enter the port number.
Timeout	Enter the connection timeout (in seconds) for the device specific protocol (SSH/HTTP/IPMI).

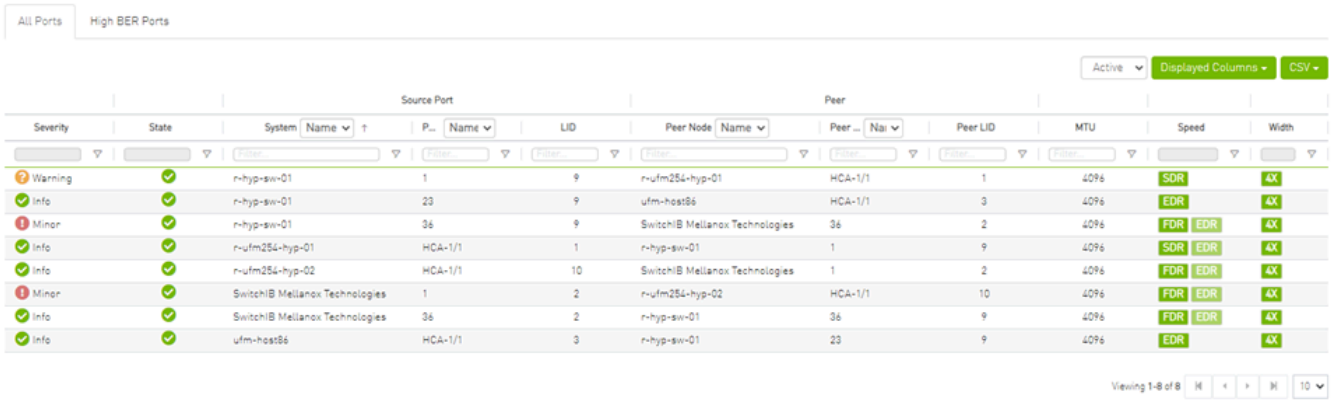
Virtual Networking Tab

This tab displays a map containing the HCAs for the selected device, and the ports and virtual ports it is connected to.



Ports Window

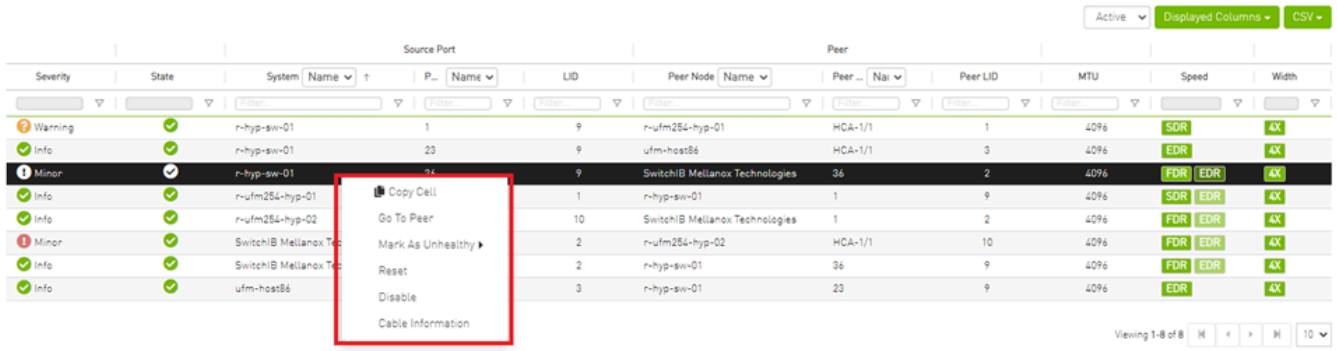
Provides a list of all ports in UFM.



The table can be filtered by port state. The filter contains two options:

- Active – only active ports
- All – all ports

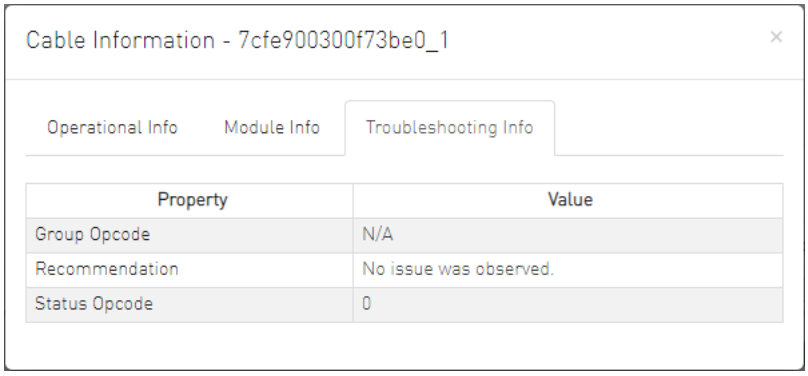
When right-clicking one of the available ports, the following actions appear:

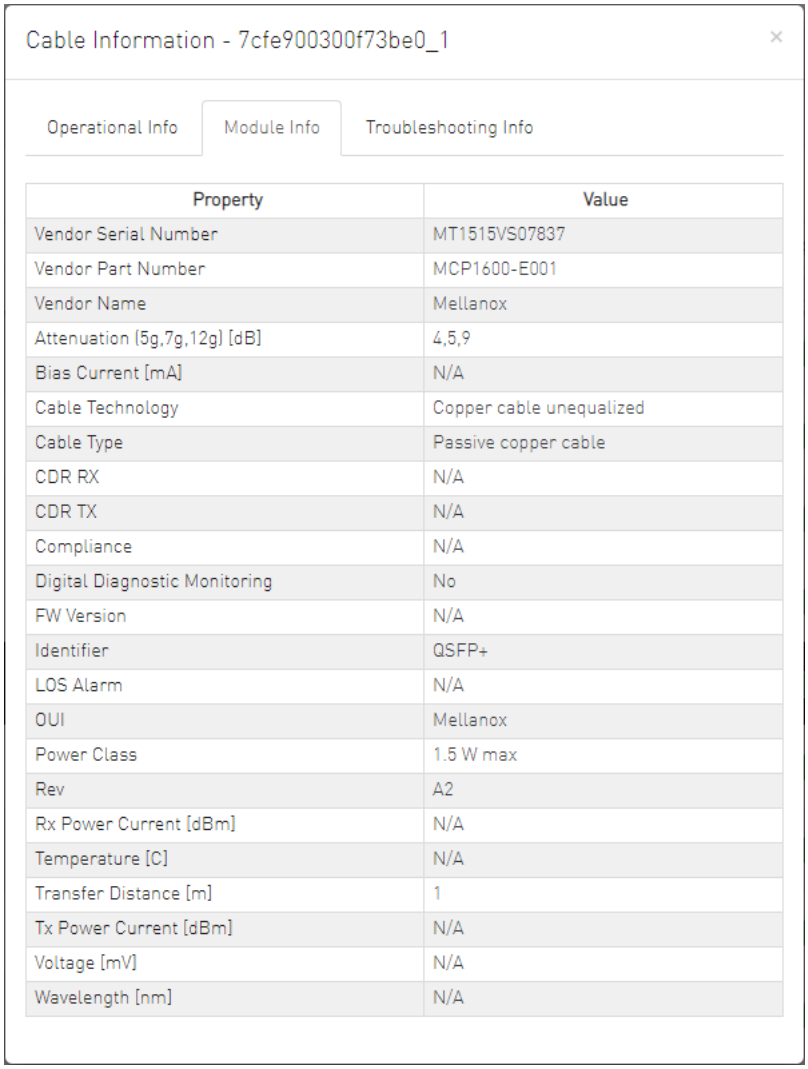


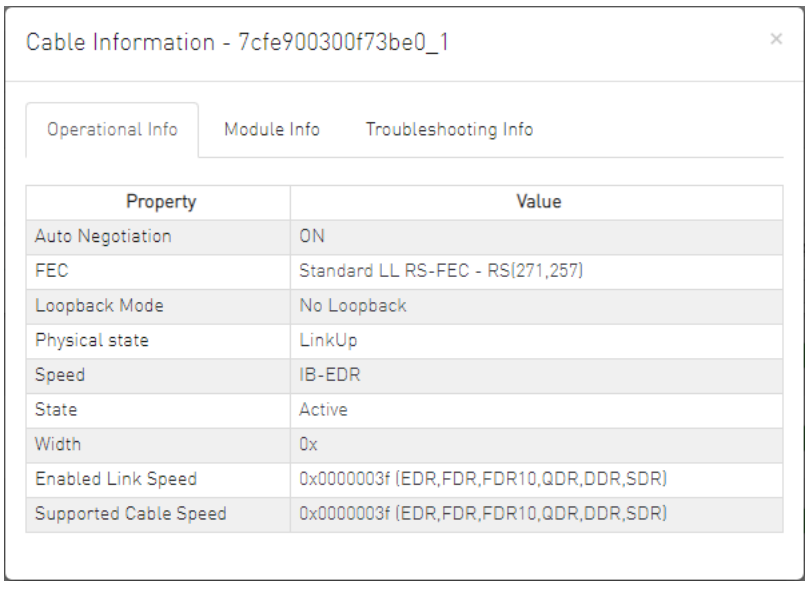
Note

All enable/disable actions on managed switches' ports are persistent. Thus, if a managed switch port is disabled, the port remains disabled even when rebooting the switch.

Clicking "Cable Information" opens up a window which provides data on operational, module, and troubleshooting information as shown in the following:





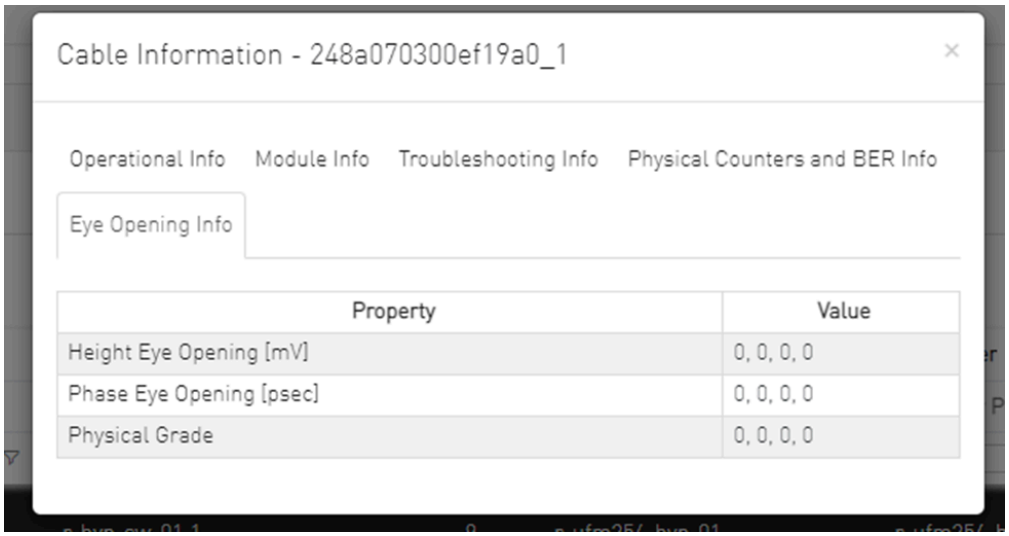


Physical Grade and Eye Opening Information

Eye opening information contains the following data:

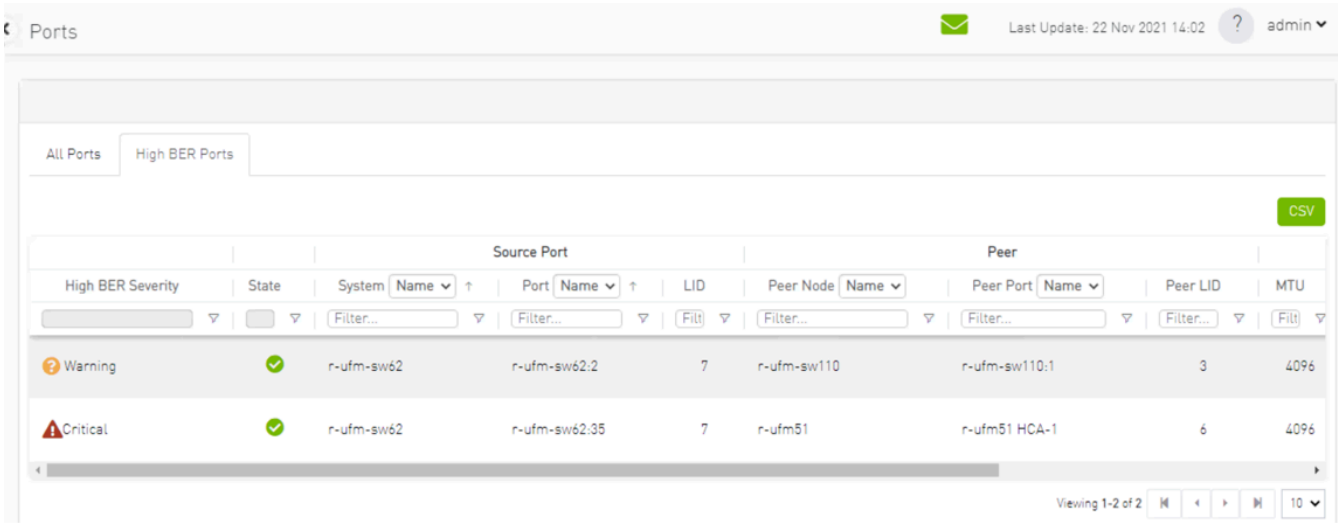
- Physical Grade: [Grade0, Grade1, Grade2, Grade3]
- Height Eye Opening [mV]: [Height0, Height1, Height2, Height3]
- Phase Eye Opening [psec]: [Phase0, Phase1, Phase2, Phase3]

A new tab called Eye Information was added under cable information modal in ports table.



Auto-isolation of High-BER Ports

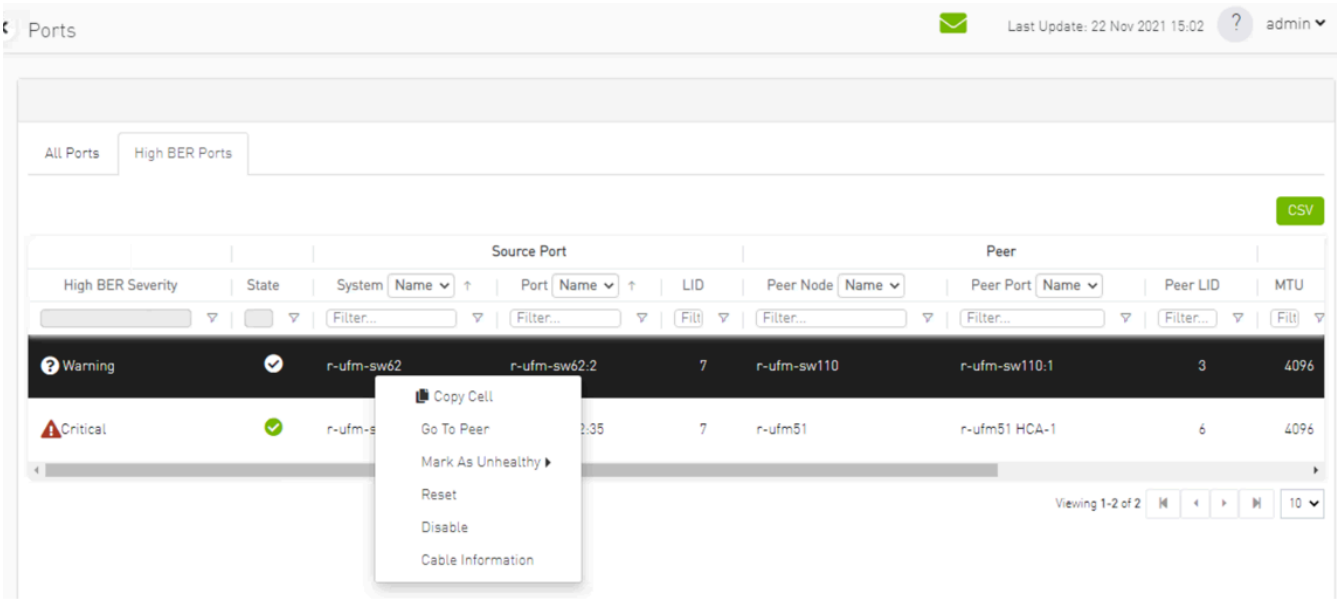
The High BER Ports tab lists all high-BER ports in the fabric.



The flags `high_ber_ports_auto_isolation` must be configured in the `gv.cfg` file to enable this feature.

For each port discovered as a high-BER port, a new event is triggered in the Events table.

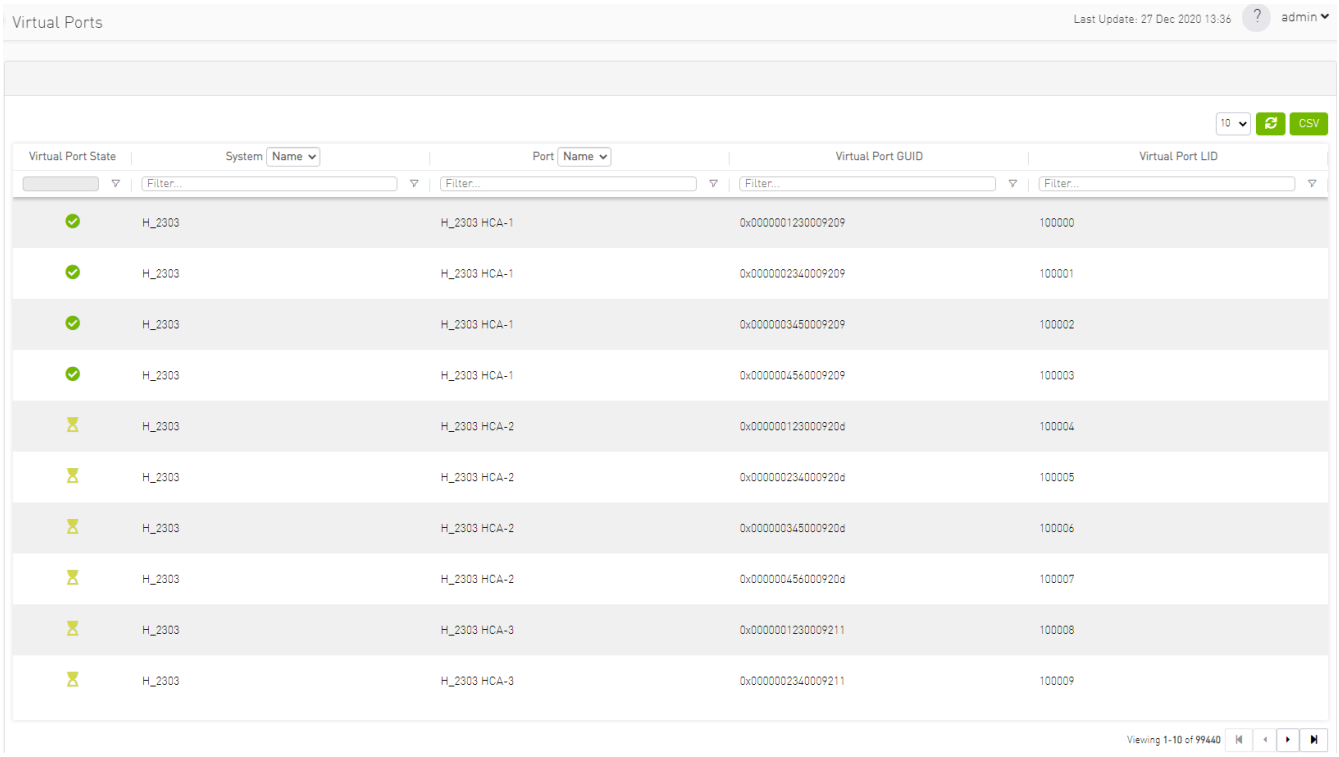
Marking the high-BER port as unhealthy suppresses all events and notifications related to the auto-isolated port.



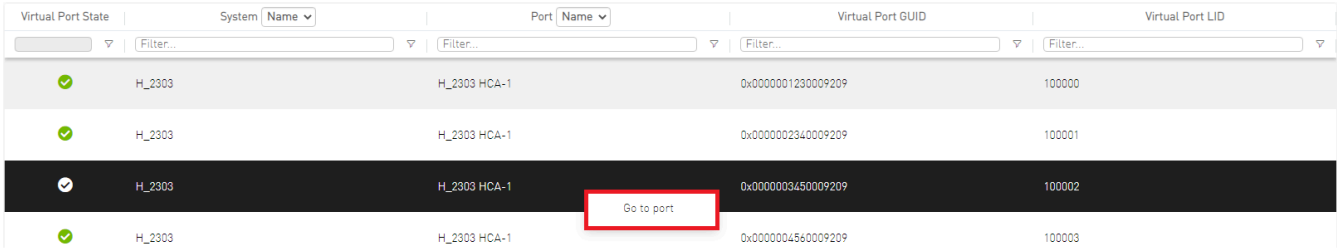
Virtual Ports Window

> **Note**
>
> This page is only available if Virtualization is enabled in gv.cfg.

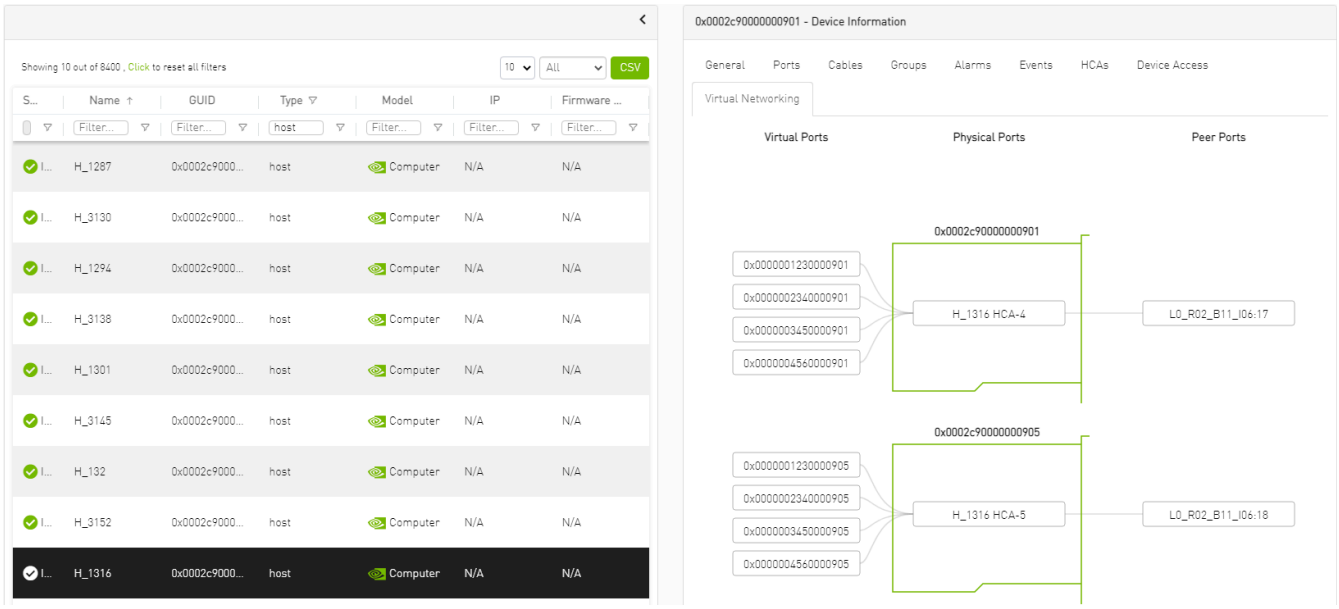
Provides a list of all virtual ports in UFM.



Right-clicking a virtual port allows navigation to the physical port mapped it is mapped to.



Clicking "Go to port" navigates to the Virtual Networking tab of the Device Information screen.

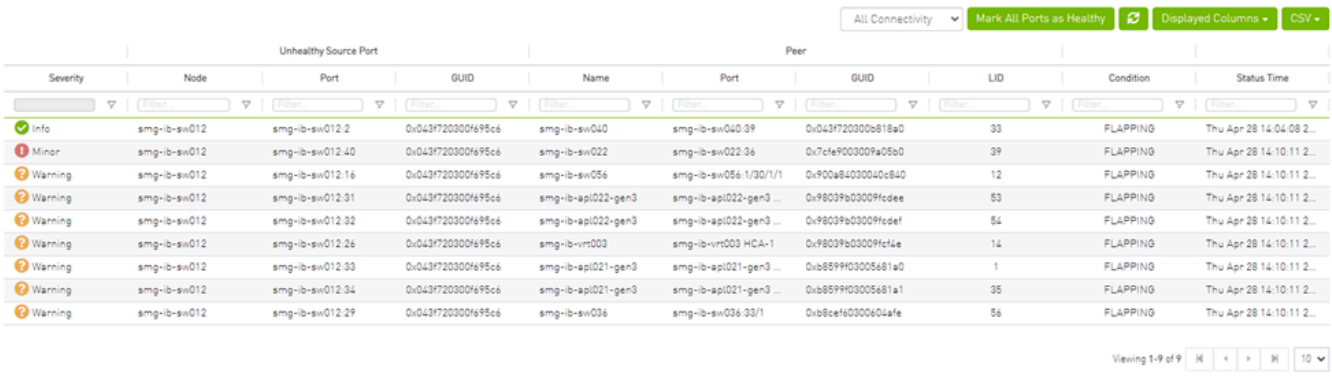


Unhealthy Ports Window

The Unhealthy Ports view shows all the unhealthy nodes in the fabric and the OpenSM health policy of the healthy/unhealthy nodes.

After the Subnet Manager examines the behavior of subnet nodes (switches and hosts) and discovers that a node is “unhealthy” according to the conditions specified below, the node is displayed in the Unhealthy Ports window. Once a node is declared as “unhealthy”, Subnet Manager can either ignore, report, isolate or disable the node. The user is provided with the ability to control the actions performed and the phenomena that declares a node “unhealthy." Moreover, the user can “clear” nodes that were previously marked as “unhealthy."

The information is displayed in a tabular form and includes the unhealthy port’s state, source node, source port, source port GUID, peer node, peer port, peer GUID, peer LID, condition, and status time.



Note

The feature requires OpenSM parameter hm_unhealthy_ports_checks to be set to TRUE (default).

Note

This feature is not available in the "Monitoring Only Mode."

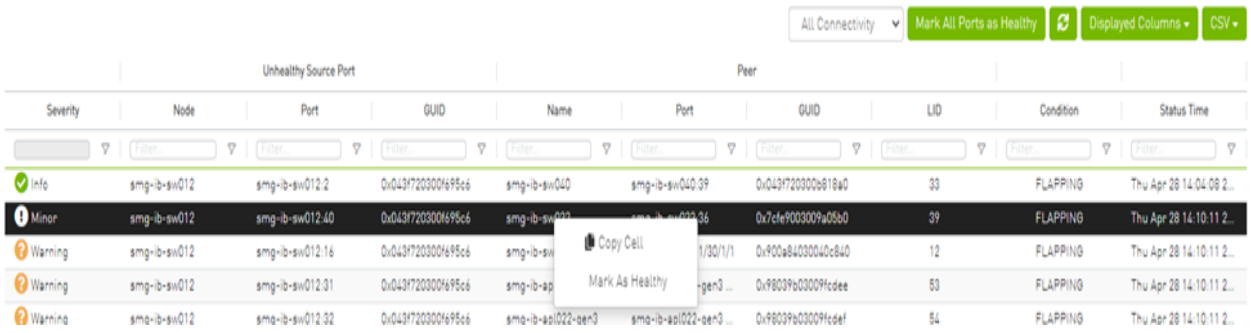
The following are the conditions that would declare a node as “unhealthy”:

- Reboot - If a node was rebooted more than 10 times during last 900 seconds
- Flapping - If several links of the node found in Initializing state in 5 out of 10 previous sweeps
- Unresponsive - A port that does not respond to one of the SMPs and the MAD status is TIMEOUT in 5 out of 7 previous SM sweeps
- Noisy Node - If a node sends traps 129, 130 or 131 more than 250 traps with interval of less than 60 seconds between each two traps
- Seterr - If a node respond with bad status upon SET SMPs (PortInfo, SwitchInfo, VLArb, SL2VL or Pkeys)
- Illegal - If illegal MAD fields are discovered after a check for MADs/fields during receive_process
- Manual - Upon user request mark the node as unhealthy/healthy
- Link Level Retransmission (LLR) – Activated when retransmission-per-second counter exceeds its threshold

All conditions except LLR generate Unhealthy port event, LLR generates a High Data retransmission event.

To clear a node from the Unhealthy Ports Tab, do the following:

1. Go to the Unhealthy Ports window under Managed Elements.
2. From the Unhealthy Ports table, right click the desired port it and mark it as healthy.



To mark a node as permanently healthy, do the following:

1. Open the /opt/ufm/files/conf/health-policy.conf.user_ext file.
2. Enter the node and the port information and set it as "Healthy."
3. Run the /opt/ufm/scripts/sync_hm_port_health_policy_conf.sh script.

Note

To control Partial Switch ASIC Failure event:

Trigger Partial Switch ASIC Failure whenever number of unhealthy ports exceed the defined percent of the total number of the switch ports.

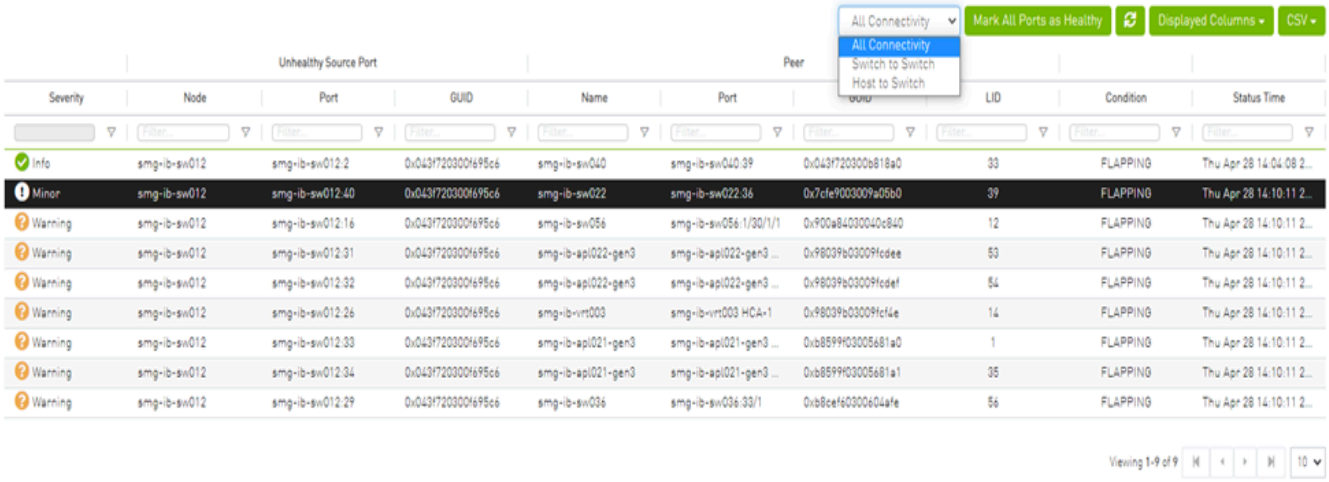
The switch_asic_fault_threshold flag (under the UnhealthyPorts section in gv.cfg file) default value is 20.

Unhealthy Port Connectivity Filter

It is possible to to filter the Unhealthy Ports table by connectivity (all, host-to-switch, or switch-to-host).

Filtering the Unhealthy Ports table is possible from the dropdown options at the top of the table which includes

- All Connectivity
- Switch to Switch
- Host to Switch

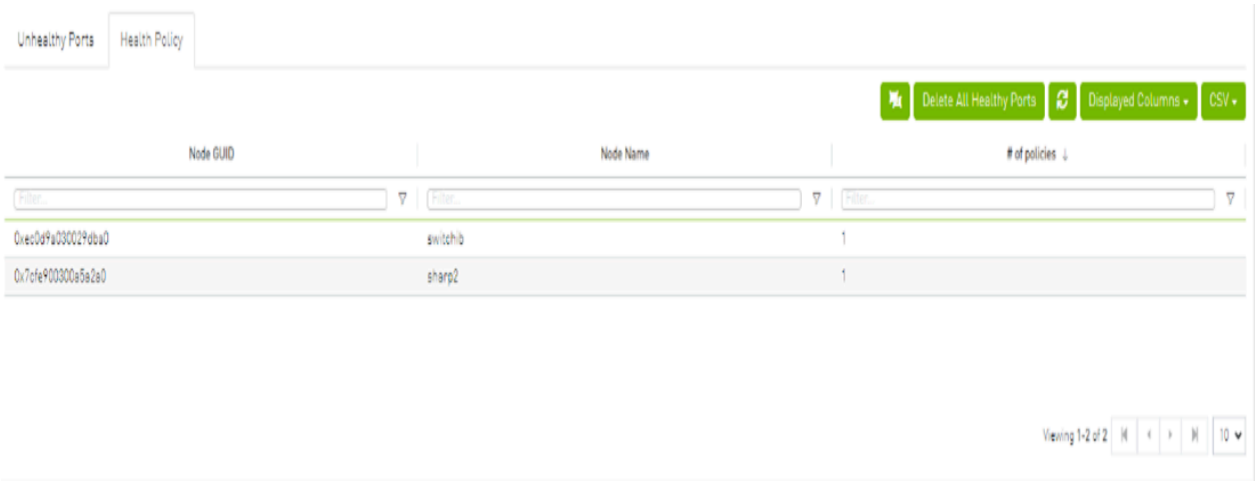


Health Policy Management

This view manages the OpenSM health policy for the healthy/unhealthy nodes and ports. The OpenSM health policy is stored in the /opt/ufm/files/conf/opensm/opensm-health-policy.conf file.

The information is displayed in a tabular form, with an option to group it either by devices or ports, and includes the health nodes/ports details (GUID, Name, policy [healthy/unhealthy])

1. Health Policy by devices:



2. Health Policy by ports:

To switch between the above views, simply click on the control button located at the top right corner of the table. By default, the devices view will be shown.

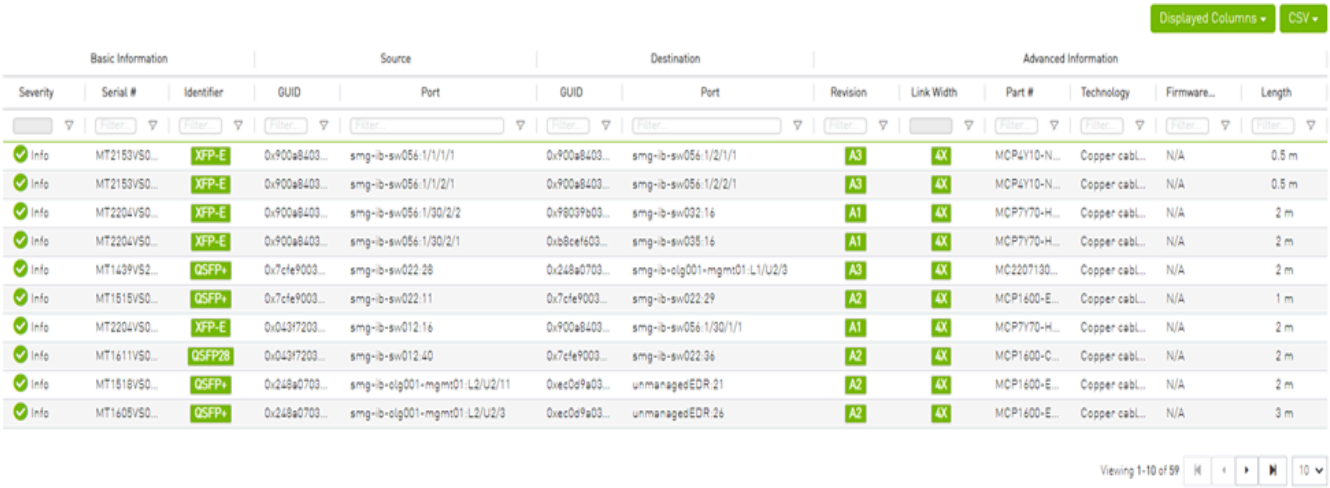
The health policy supports the following capabilities. When you select a policy and right-click, you can perform the following actions:

1. Delete the Policy
2. Mark the selected healthy policies as unhealthy (Isolate/No discover)
3. Mark the selected unhealthy policies as healthy

If you wish to delete all the healthy ports from the health policy, click on the 'Delete All Healthy Ports' option situated at the top right corner of the policy table.

Cables Window

Provides a list of all cables in UFM. For more information, see Device's Cables Tab.



Right-clicking a cable from the list allows users to Collect System Dump for the endpoints of the link.

Groups Window

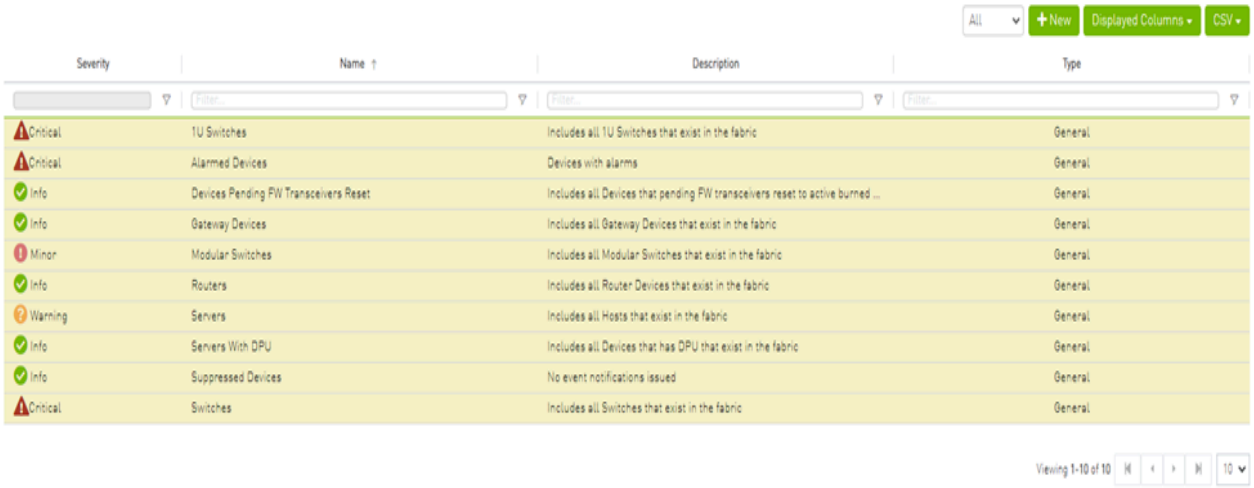
The Groups window allows users to create new groups of devices and provides information about existing groups.

> **Note**
>
> All predefined groups have Read permissions only, except Suppressed_Devices to/from which the user is also able to add/remove members or devices.

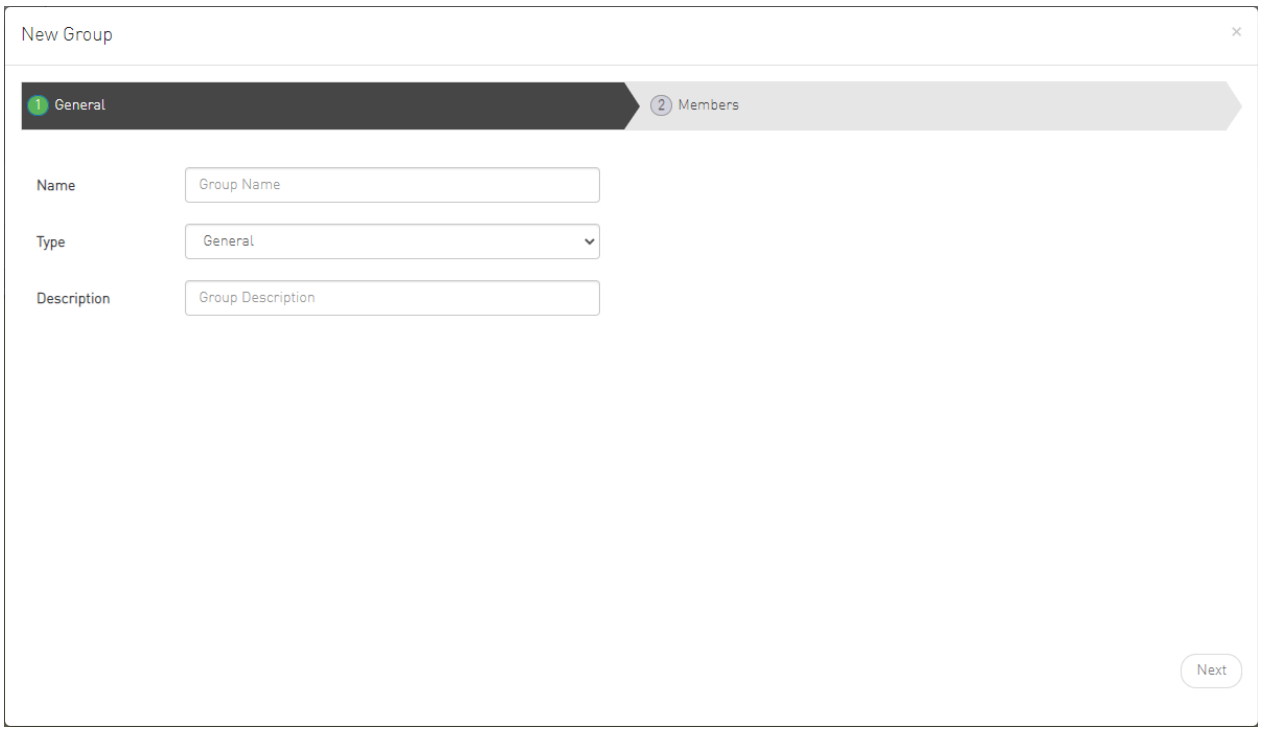
> **Note**
>
> The following predefined groups auto-populate upon UFM startup: Switches, 1U_Switches, Modular_Switches, Gateway_Devices, and Hosts.

To create a group of devices, do the following:

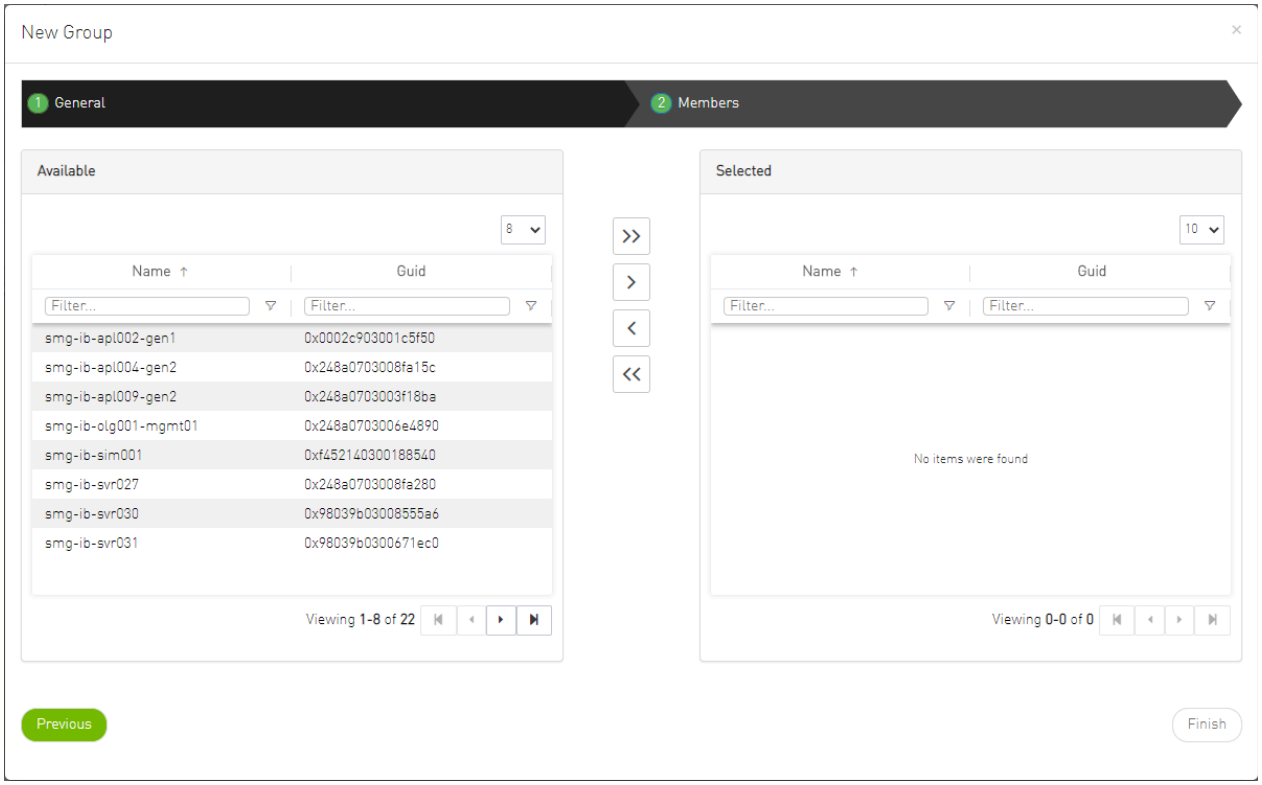
1. Click “New” under “Groups.”



2. In the New Group wizard, fill in the required information under the General tab: Name (must be between 4-20 characters), Type (General/Rack/Port), and Description (optional), and click **Next**.

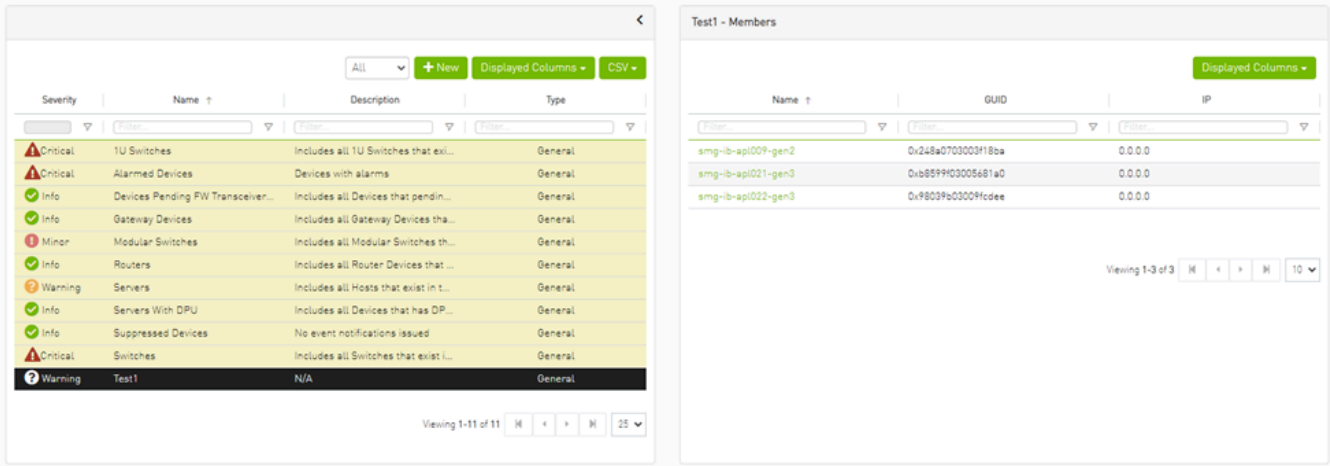


3. Under Members tab, move the members of the new group from the **Available** list to the **Selected** list.



4. Click “Finish” and the new group will appear under the Groups window.

Group members details – port’s hostname, port’s GUID, and device’s IP address – can be viewed when selecting the group from the list of all groups available.



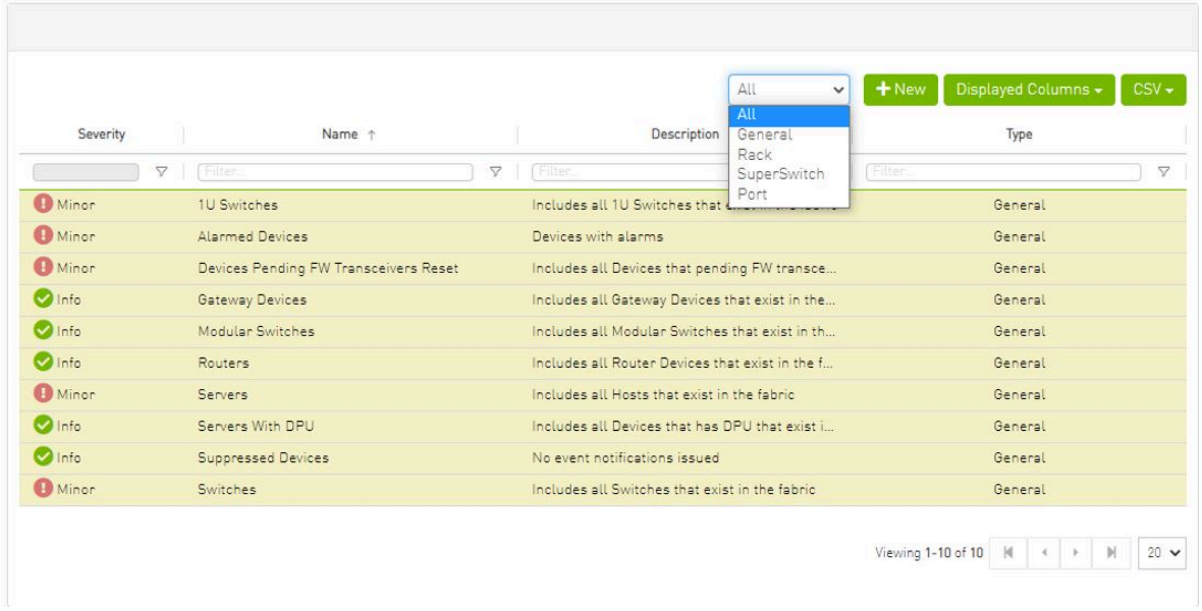
Group Actions

Right-clicking a group enables performing the following actions:

- **Edit** – groups can be modified either by editing the group description under **General** tab, or substituting group members under **Members** tab

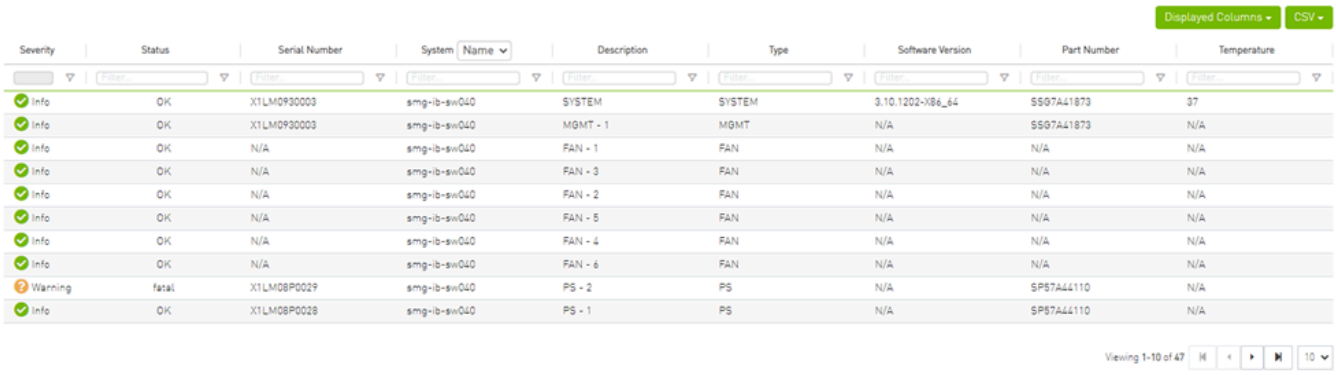
- **Delete** – existing groups can be deleted from the list
- **Remove All Members** – all members of an existing group can be removed at once
- **Collect System Dump** – sysdump may be generated for all members of an existing group

  The user can filter group by type (General, Rack, Super Switch and Port)



Inventory Window

Provides a list of all modules in UFM. For more information, see Device's Inventory Tab.



PKeys Window

The PKeys window allows users to create new groups of ports and provides information about existing PKeys.

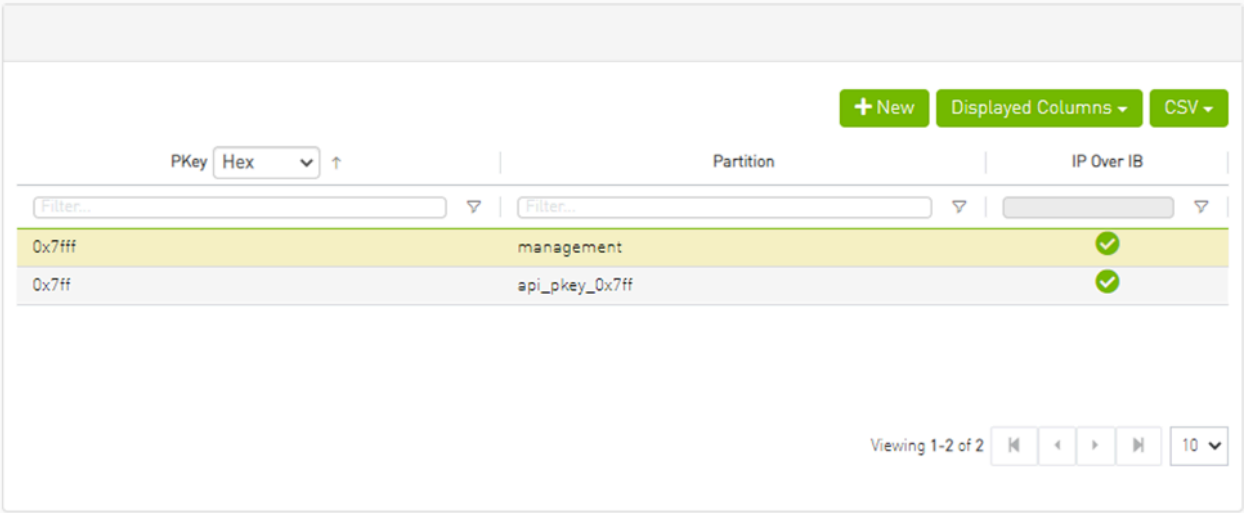
> **Note**
>
> This window offers one predefined PKey (highlighted in the list of PKeys): Management key 0x7fff with Read permissions only.

For further information about InfiniBand partitioning (Pkeys management), please refer to the Partitioning Appendix.

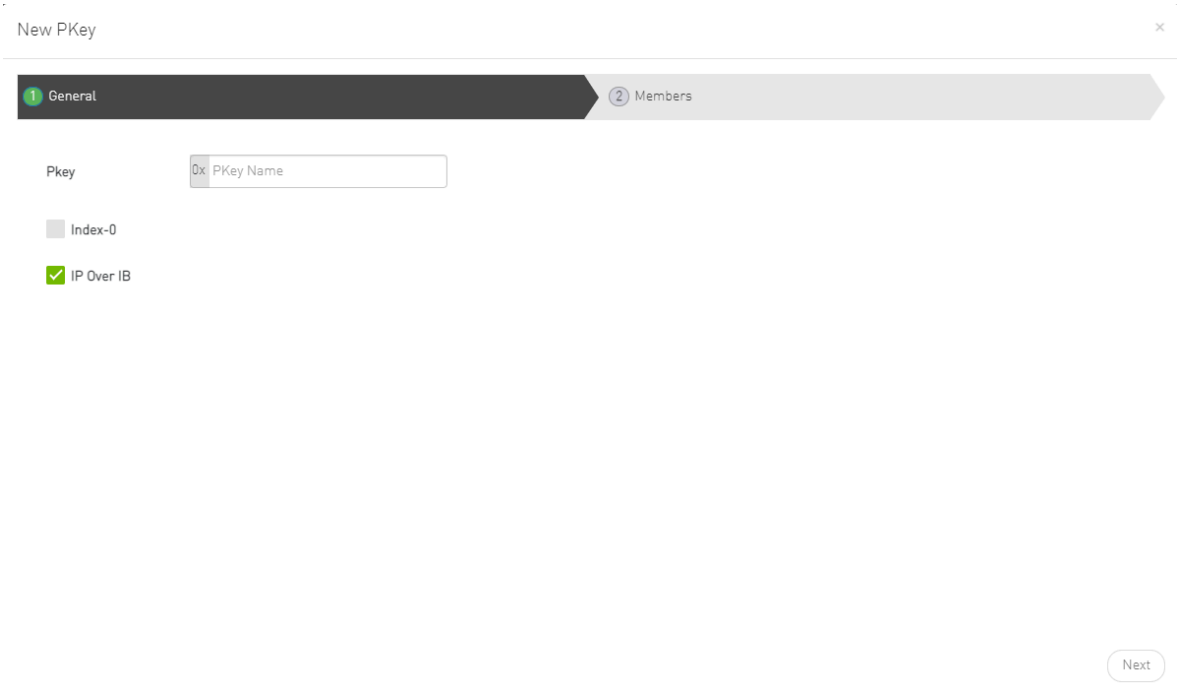
Creating New PKey

1. Click the “New” button under “PKeys".

   Please note that the yellow highlighted PKeys are predefined ones.

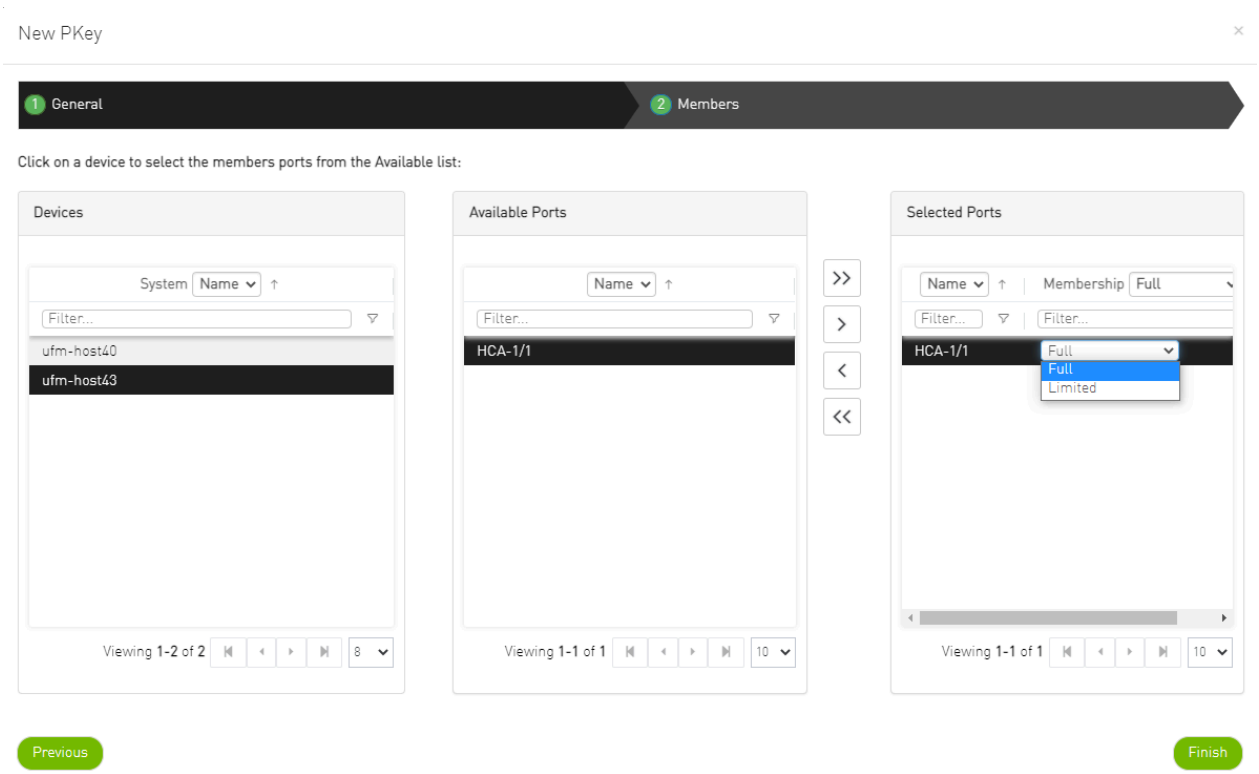


2. In the New PKey wizard, fill in the required information under the General tab:
   - Name—must be between 0x1 and 0x7fff, inclusive
   - Index-0 attribute—True/False
   - IP Over IB attribute—True/False



3. Click "Next."

4. Under Members tab, select the device of which ports you would like to group in one PKey, and move the members (ports) from the **Available** list to the **Selected** list. For each member (port) you may specify a membership type (Full/limited).

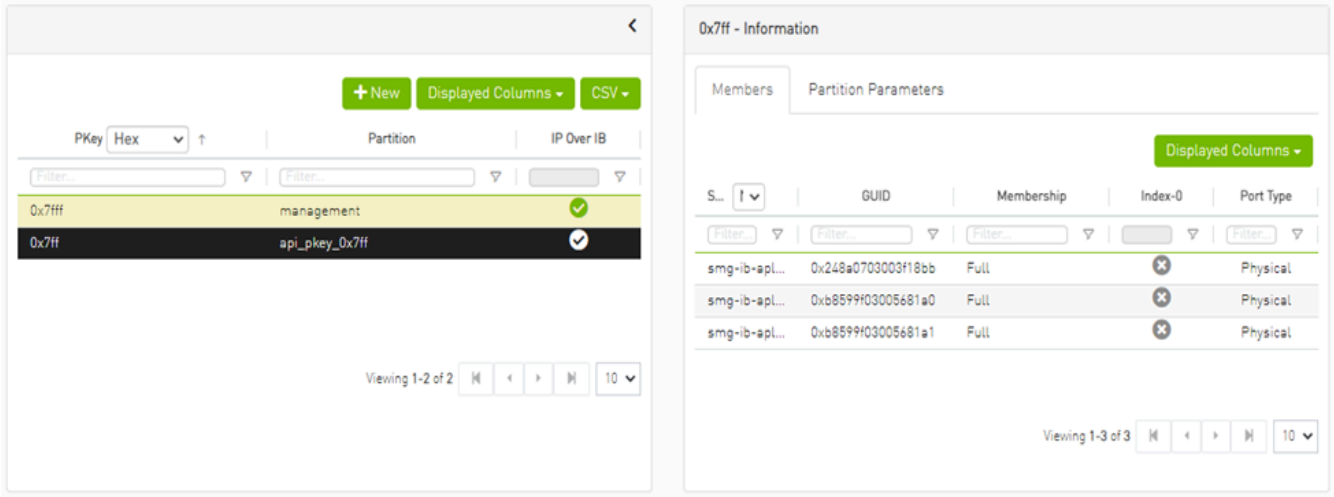


5. Click “Finish”. The new PKey will become available under the PKey window.

When selecting a PKey from the PKeys table, **PKey Information** table will appear on the right side of the screen. This table provides information on the PKey's members and QoS settings.

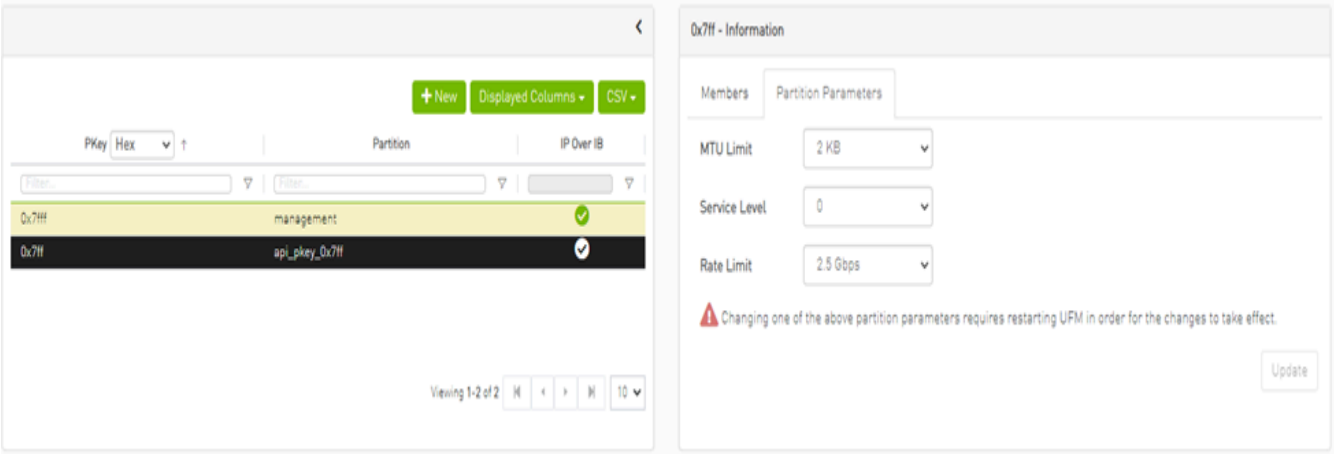
PKey Members Tab

Provides details on the PKey members: port’s hostname (node), device's IP address, port GUID, port number, membership and index-0 attributes values.



PKey QoS Tab

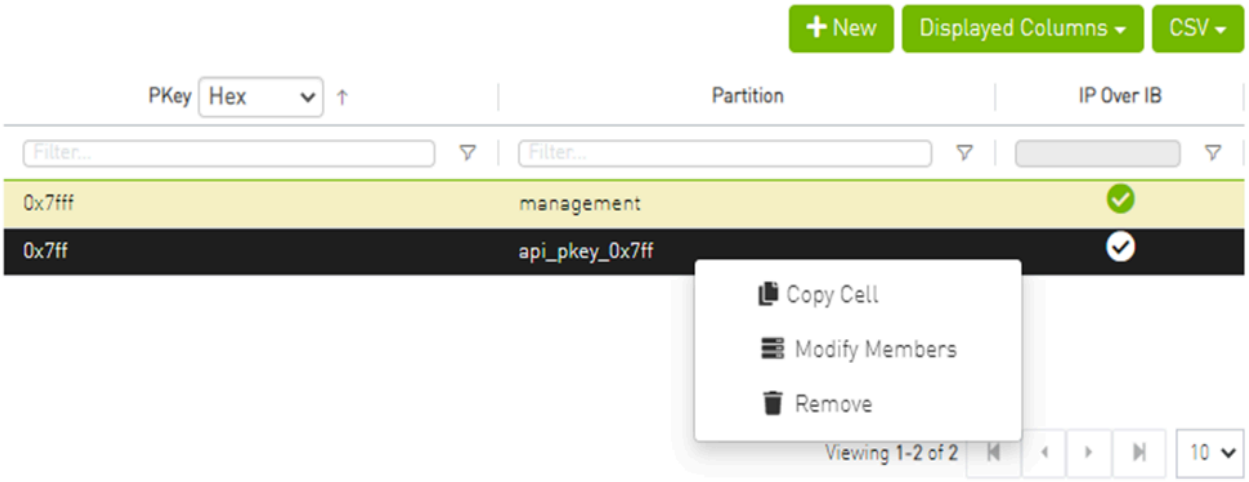
Displays the current partitioning parameter settings of the selected PKey: MTU Limit, Service Level and Rate limit. These settings can be modified by the user.



PKey Actions

Right-clicking one PKey from the list enables performing the following actions:

- **Modify Members** – PKeys can be modified either by editing the attributes under **General** tab, or updating the members under **Members** tab. Including updating ports memberships.
- **Remove** – existing PKeys can be deleted from the list.



> **Note**
>
> For information on partitioning, refer to Appendix – Partitioning.

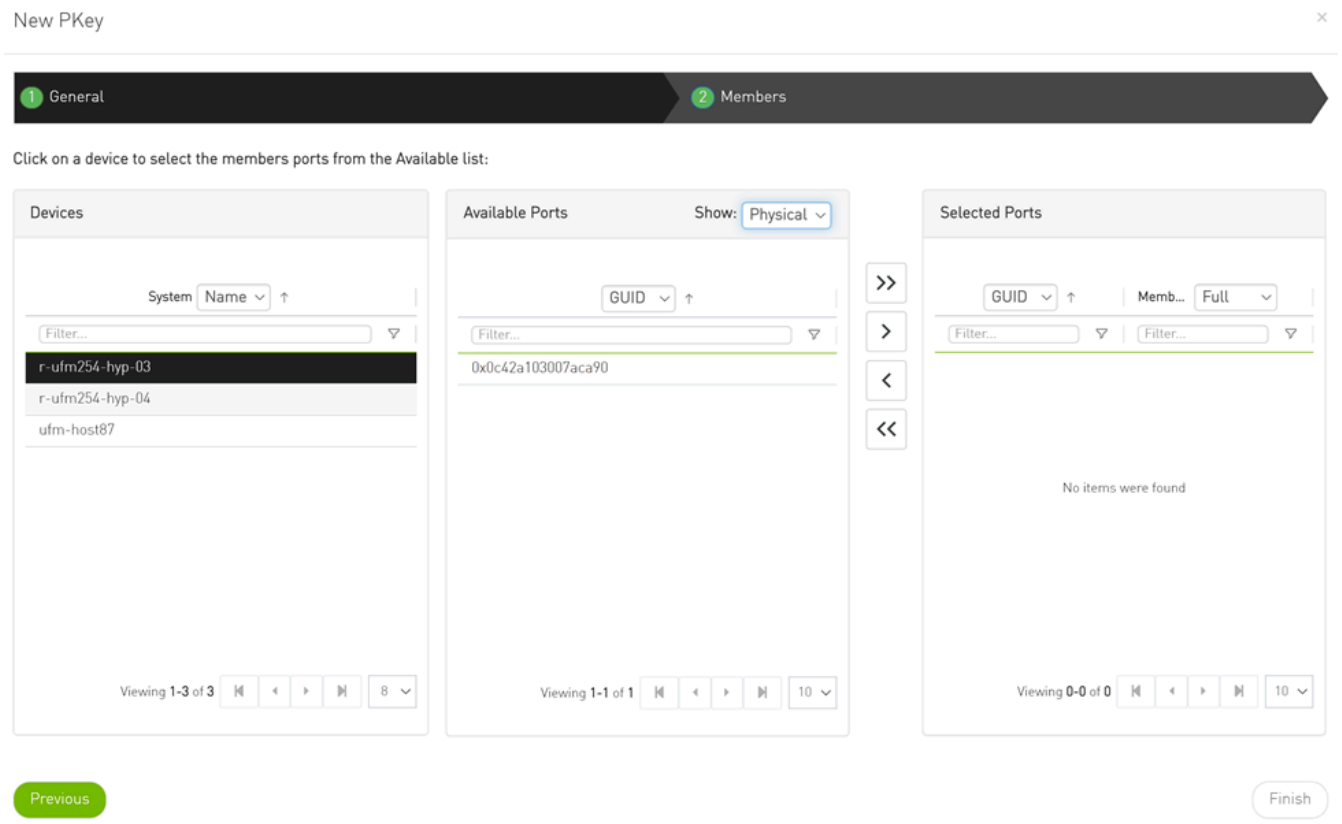
> **Note**
>
> Note that restarting OpenSM is required for the QoS parameters change to take effect.

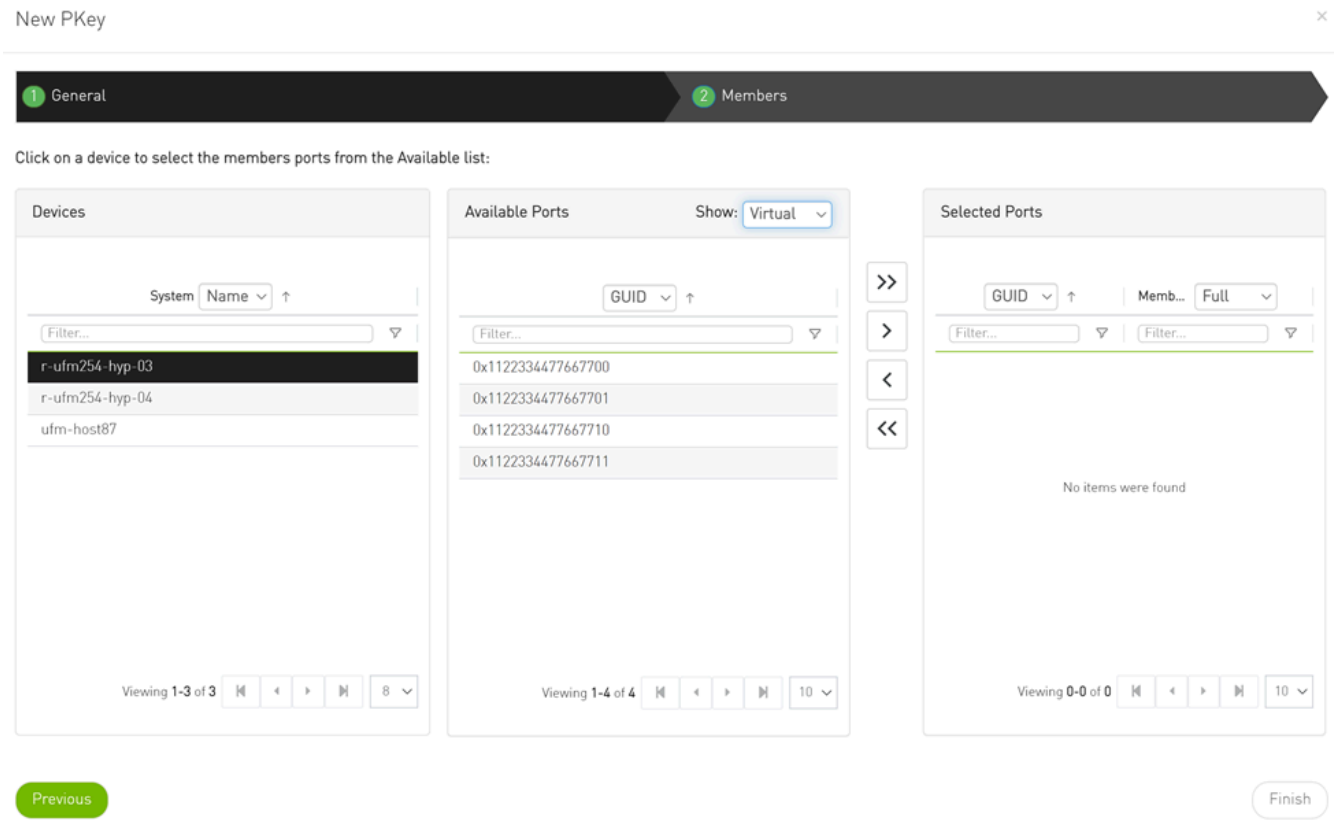
Support Pkey with Virtual Ports

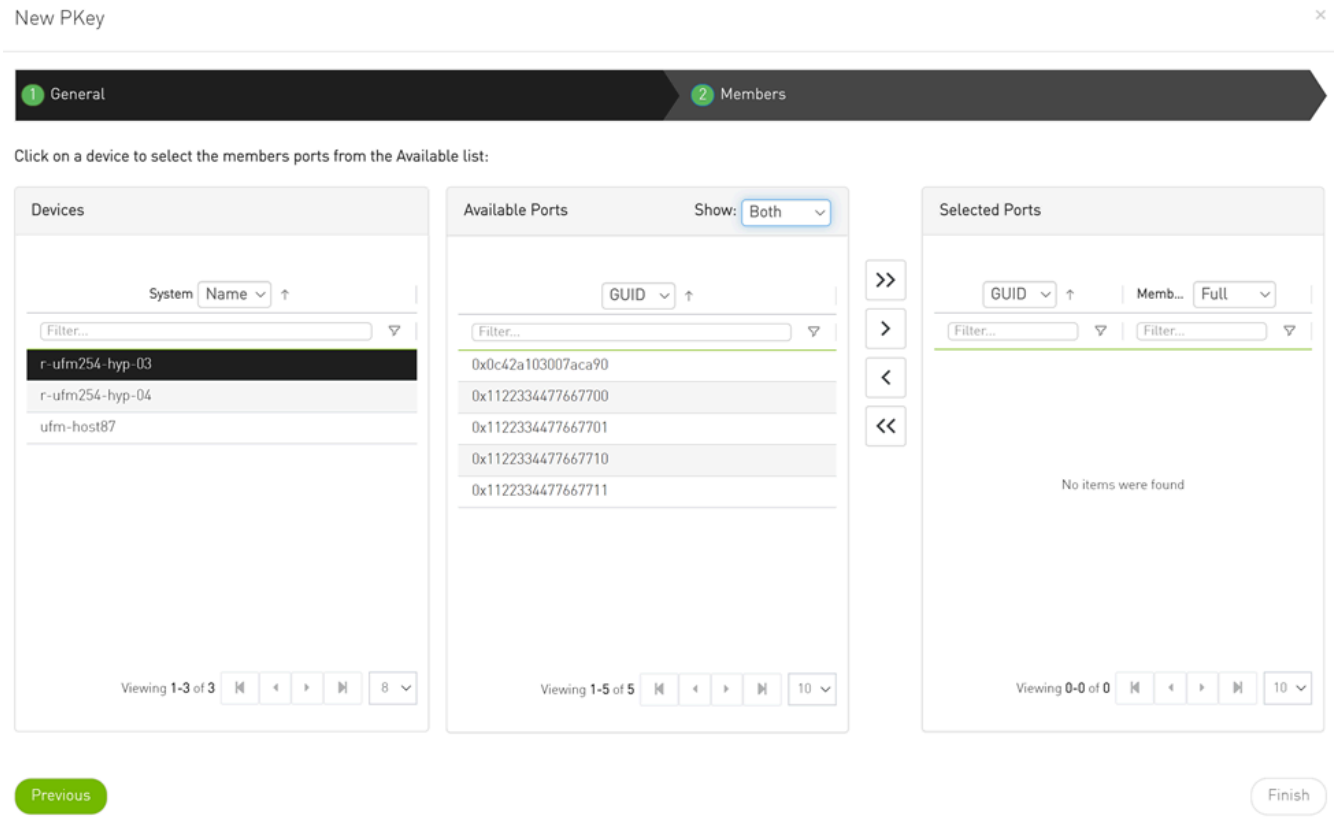
Creating a pkey with virtual ports is supported, so pkey can contain the following types of port:

- Physical
- Virtual
- Both physical and virtual

The create new pkey wizard dropdown includes port types.

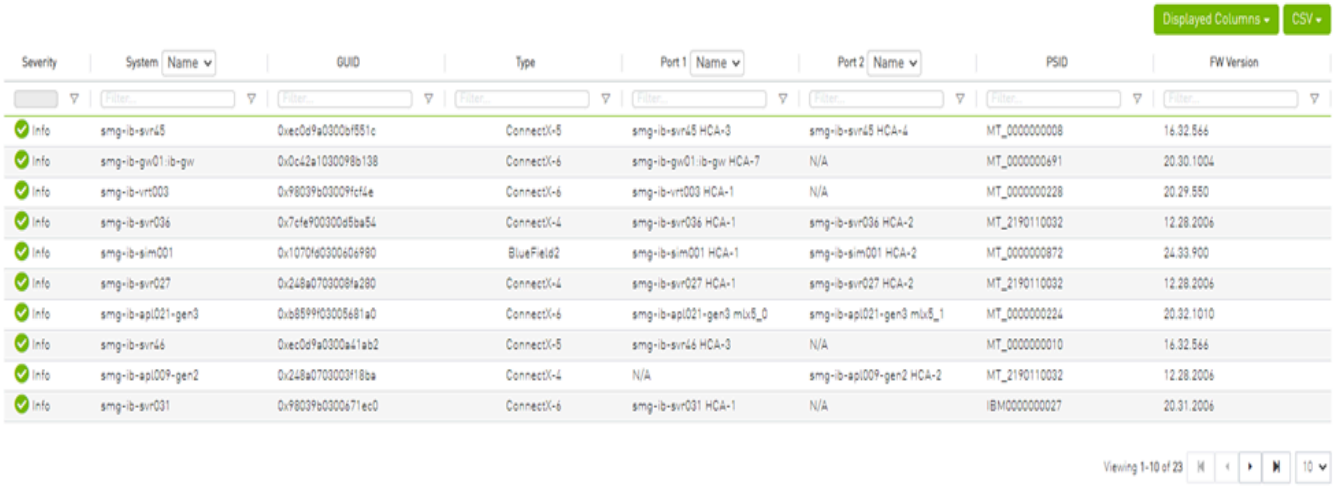



New PKey
1 General
2 Members
Click on a device to select the members ports from the Available list:
Devices
System Name
Filter...
r-ufm254-hyp-03
r-ufm254-hyp-04
ufm-host87
Viewing 1-3 of 3
Available Ports
Show: Virtual
GUID
Filter...
0x1122334477667700
0x1122334477667701
0x1122334477667710
0x1122334477667711
Viewing 1-4 of 4
Selected Ports
GUID
Memb... Full
No items were found
Viewing 0-0 of 0
Previous
Finish


New PKey
1 General
2 Members
Click on a device to select the members ports from the Available list:
Devices
System Name
Filter...
r-ufm254-hyp-03
r-ufm254-hyp-04
ufm-host87
Viewing 1-3 of 3
Available Ports
Show: Both
GUID
Filter...
0x0c42a103007aca90
0x1122334477667700
0x1122334477667701
0x1122334477667710
0x1122334477667711
Viewing 1-5 of 5
Selected Ports
GUID
Memb... Full
No items were found
Viewing 0-0 of 0
Previous
Finish

HCAs Window

Provides a list of all the HCAs of the hosts in UFM. For more information, see section "HCAs Tab".

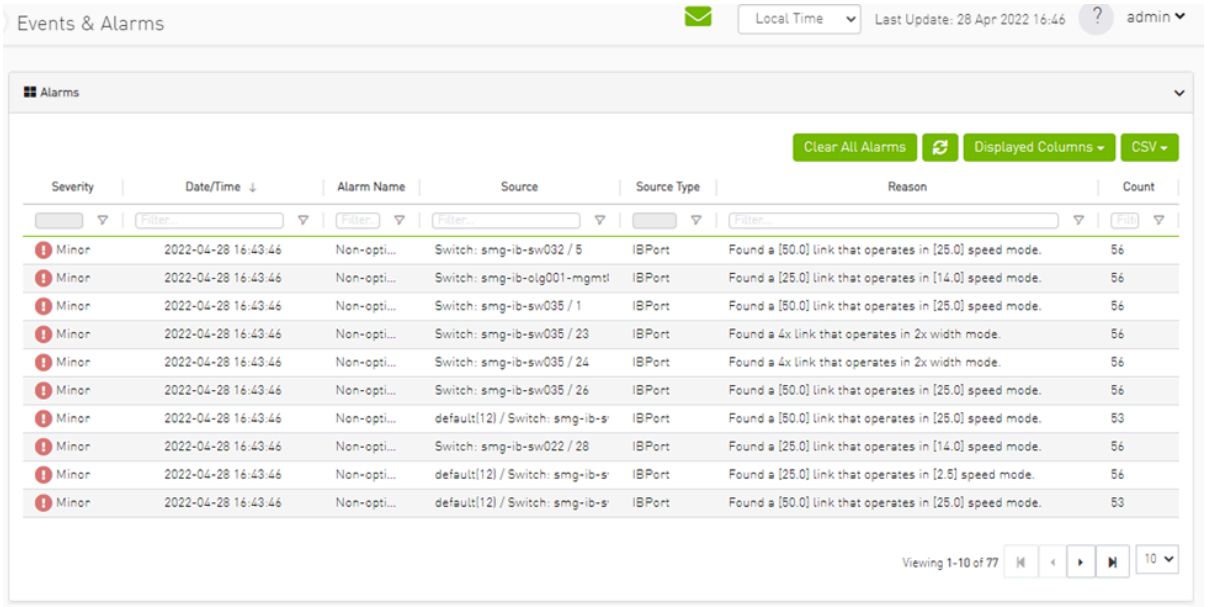


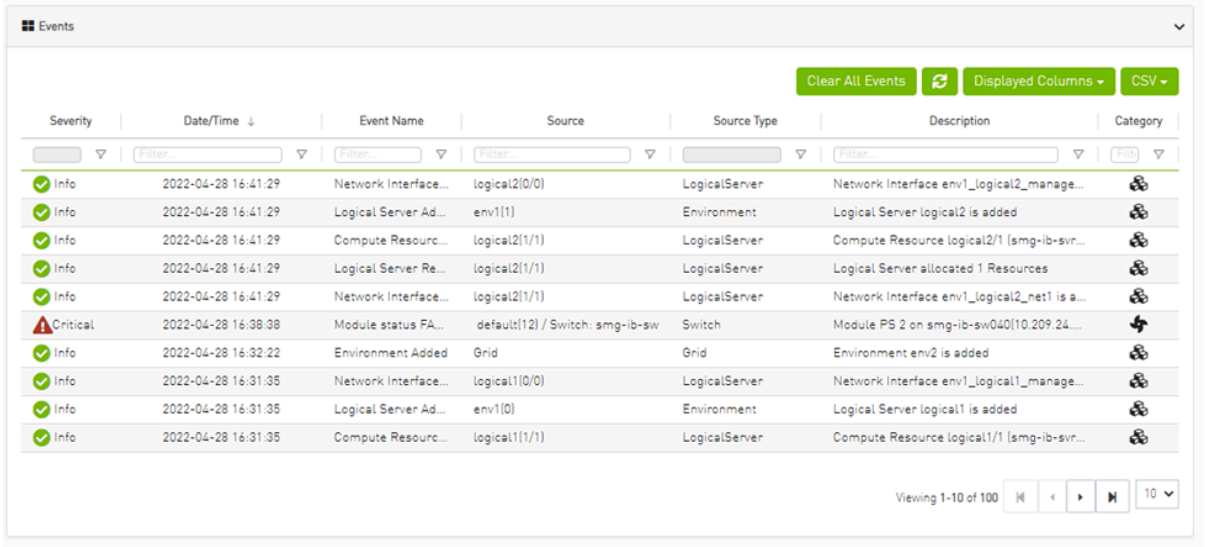
Events & Alarms

> **Note**
>
> All information provided in a tabular format in UFM web UI can be exported into a CSV file.

UFM allows you to identify any problem including ports and device connectivity using events and alarms. Problems can be detected both prior to running applications and during standard operation.

Events trigger alarms (except for "normal" events. i.e., Info events) when they exceed a predefined threshold. Events and alarms can be configured under Events Policy tab under Settings window. For more information, refer to Events Policy Tab.





Users can enable the events persistency mechanism from the gv.cfg. This allows the user to see the events in the case of restarting the UFM or in HA mode.

> **Note**
>
> Alternatively you can run the following commands:
>
> - ufm events persistency enable
> - ufm events max-restored

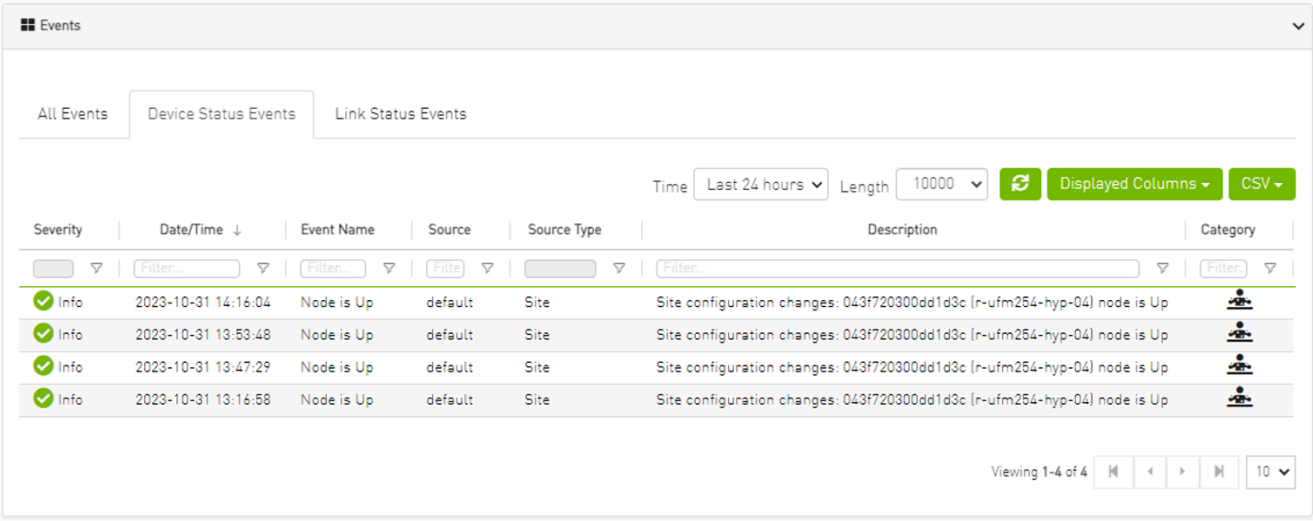
The persistency is deactivated by default and can be enabled by the following controlled parameters in the config file:

- `max_restored_events = 50 #` – will determine the number of events to restore
- `events_persistency_enabled = true #` – will set to true for the feature to work

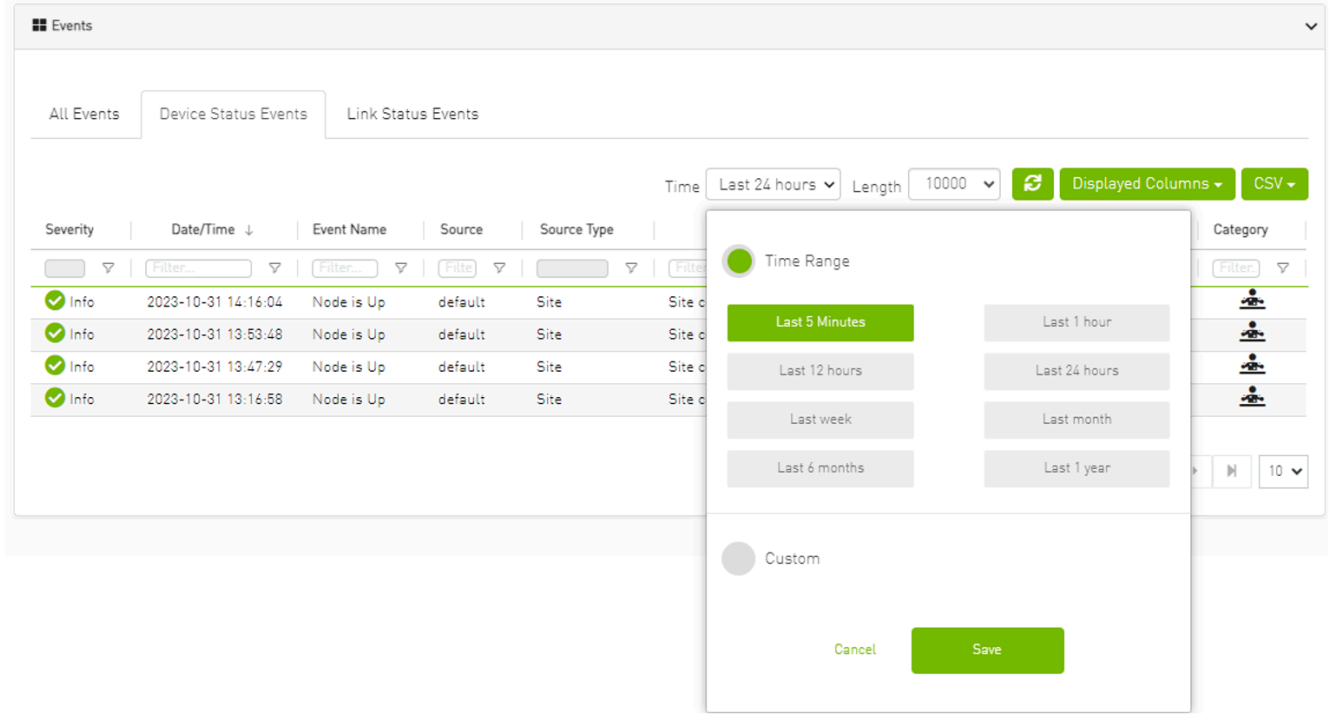
Device Status Events

The Device Status Events tab displays topology change events related to devices in a table. it will support the following event types:

- None is Up/Down
- Switch is Up/Down
- Director Switch is Up/Down



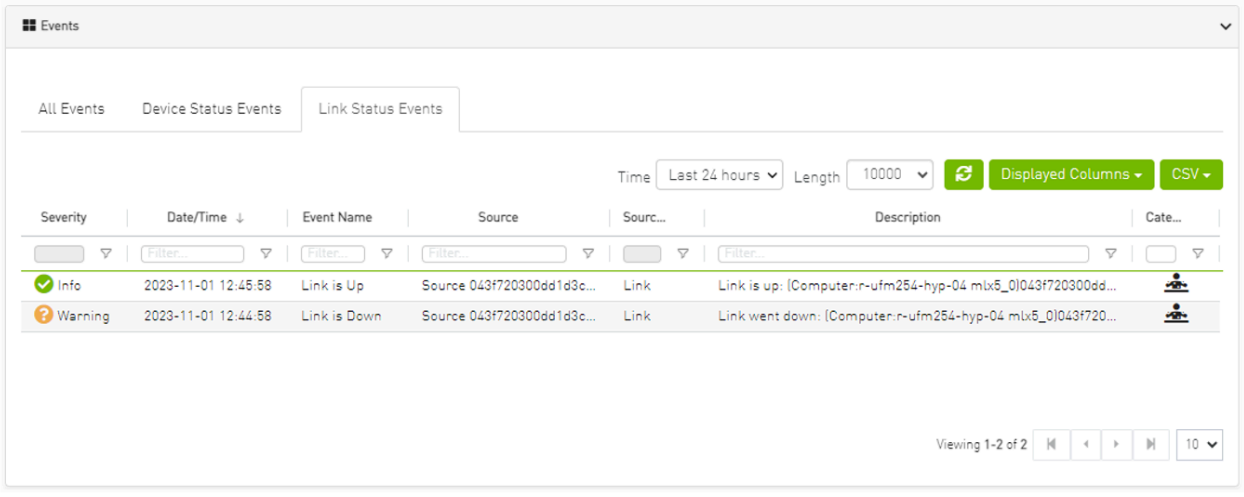
Filters are be provided to allow events filtering by the desired time interval with a length limit.



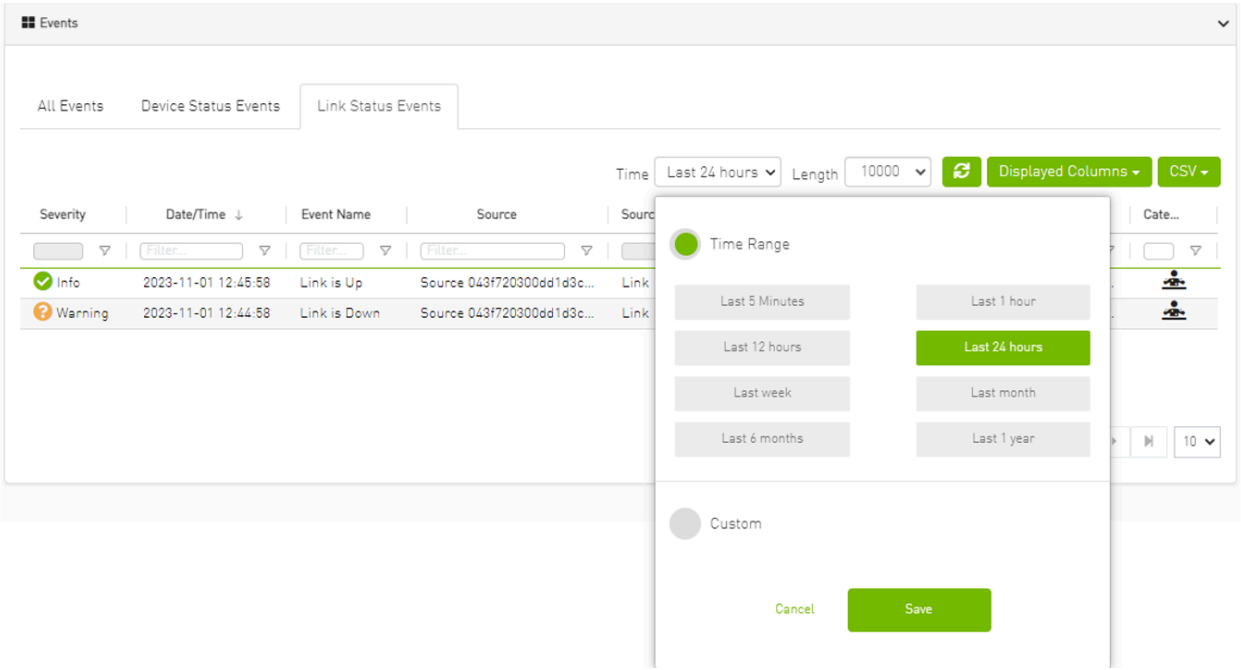
Link Status Events

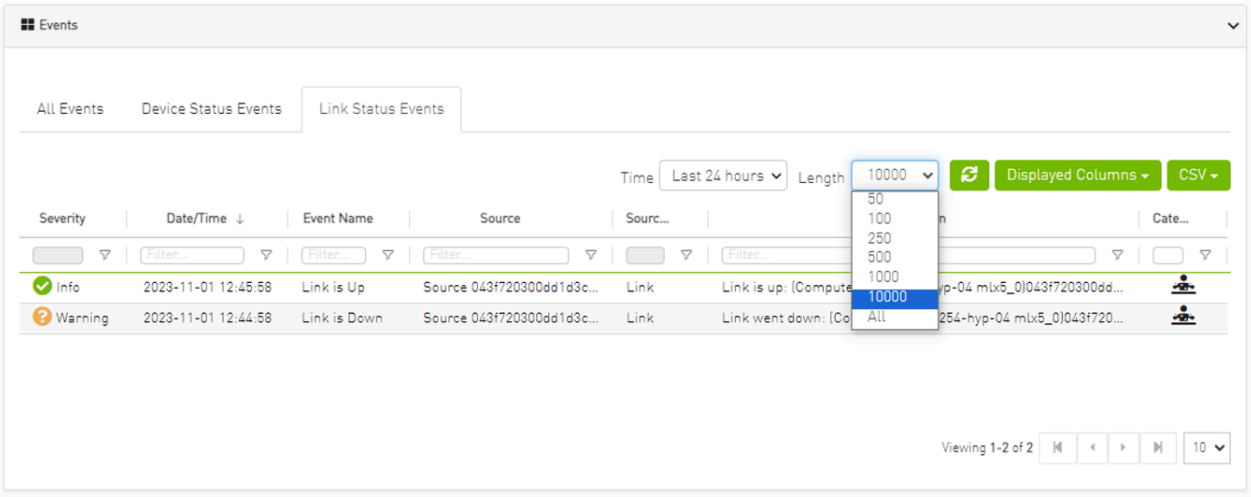
The Link Status Events tab displays topology change events related to links in a table. It supports the following event type:

- Link is Up/Down



Filters are provided to allow filtering by the desired time interval in a time range.





> **Note**
>
> Rge related switch context menu is displayed only if the event type is 'Switch is Up/Down'. Other event types show the default context menu, which is 'Copy Cell'.

Telemetry

Establishing Telemetry Sessions

UFM Telemetry allows tracking network bandwidth, congestion, errors, and latency. UFM offers the following telemetry features:

- Real-time monitoring views
- Multiple attributes monitoring
- Intelligent Counters: provide error and congestion counters
- InfiniBand port-based error counters
- InfiniBand congestion XmitWait counter-based congestion measurement
- InfiniBand port-based bandwidth data

The following actions may be taken with the telemetry session panels:

- Rearranging – using a simple drag-and-drop function
- Resizing – by hovering over the panel's border

It is also possible to get a larger view of a telemetry session by clicking the pop-out button on the top right-hand corner of each panel.

Telemetry Session Objects and Attributes

Monitored objects may be ports or devices in the fabric.

Monitored attributes can be raw counters or calculated counters:

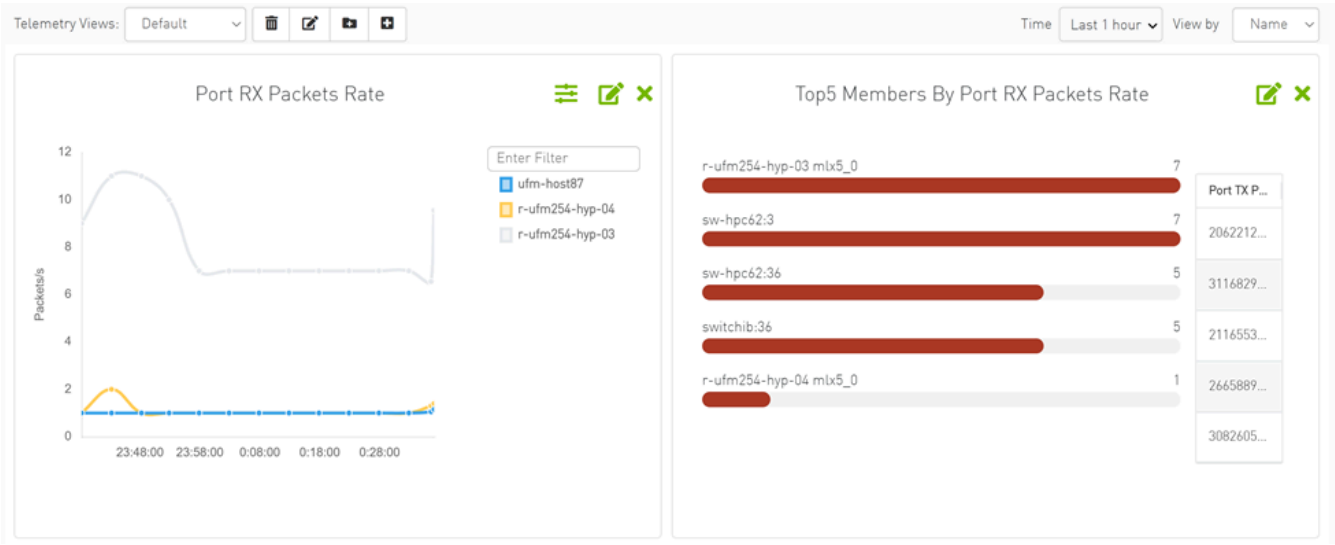
- A raw attribute is a simple attribute to be monitored (e.g., Port TX Wait)
- A calculated attribute is an attribute that has been calculated based on one or more counters (e.g., PortXmitPktsRate)

Telemetry Views

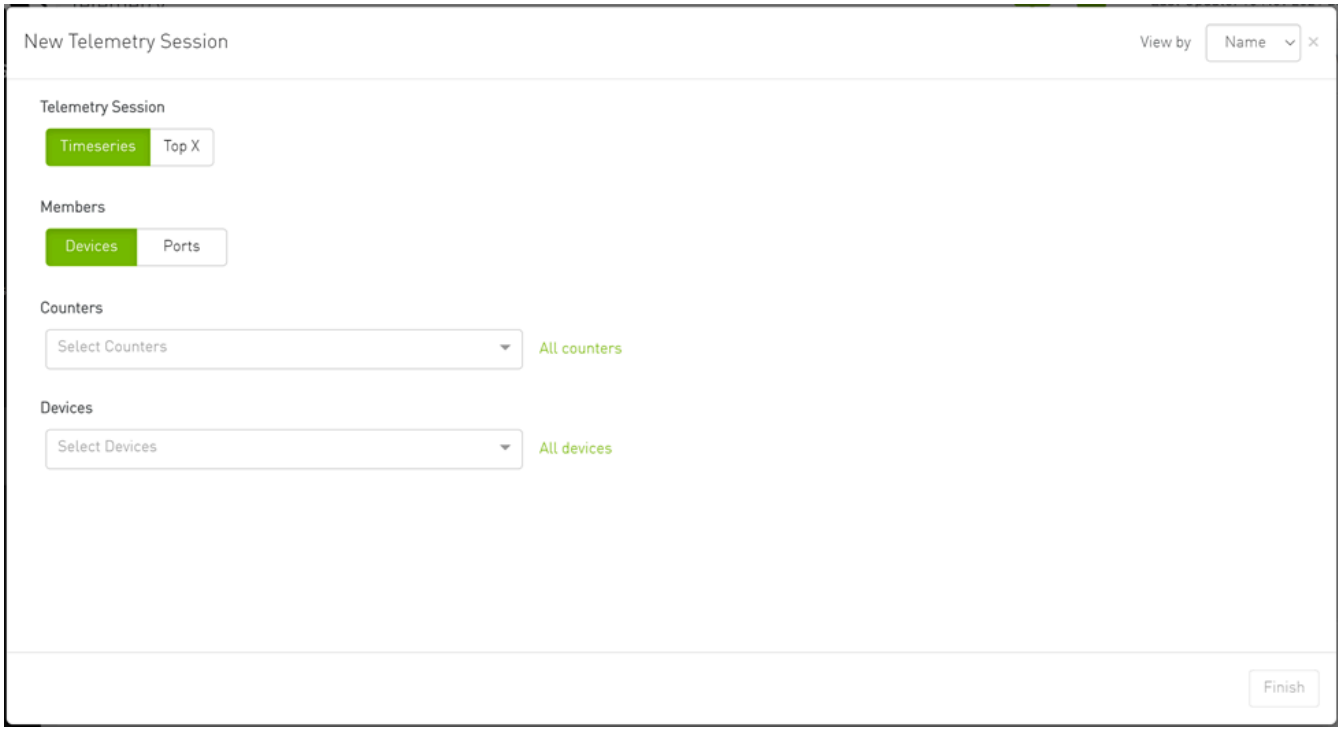
Telemetry contains multiple views; the user can create, edit, and delete views.

Telemetry supports two types of panels, time-series which show the relationship between time and counter value for a specific device, and topX, which show all ports with pick by counter greater than topX value.

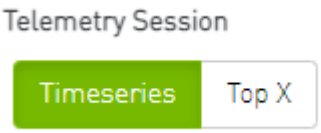
> **Note**
>
> TopX is not supported in the case of the ibpm telemetry provider. The telemetry provider is hidden in this case.



The panel can be created by filling in the following model:



The user can select one of the following telemetry session modes:



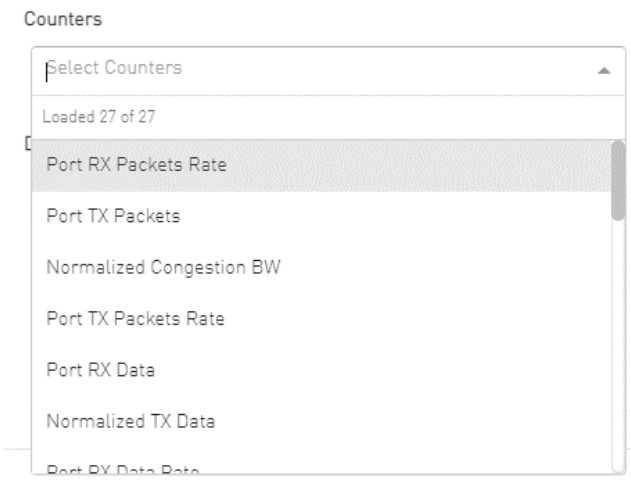
- Timeseries: Provide the user with historical/live time-series graphs of the selected counters for the selected devices/ports.
- Top X: Provides the user with Top X ports by the selected counters (where X is 5, 10, 15, 20).

You can select the members grouping type; Devices or Ports:

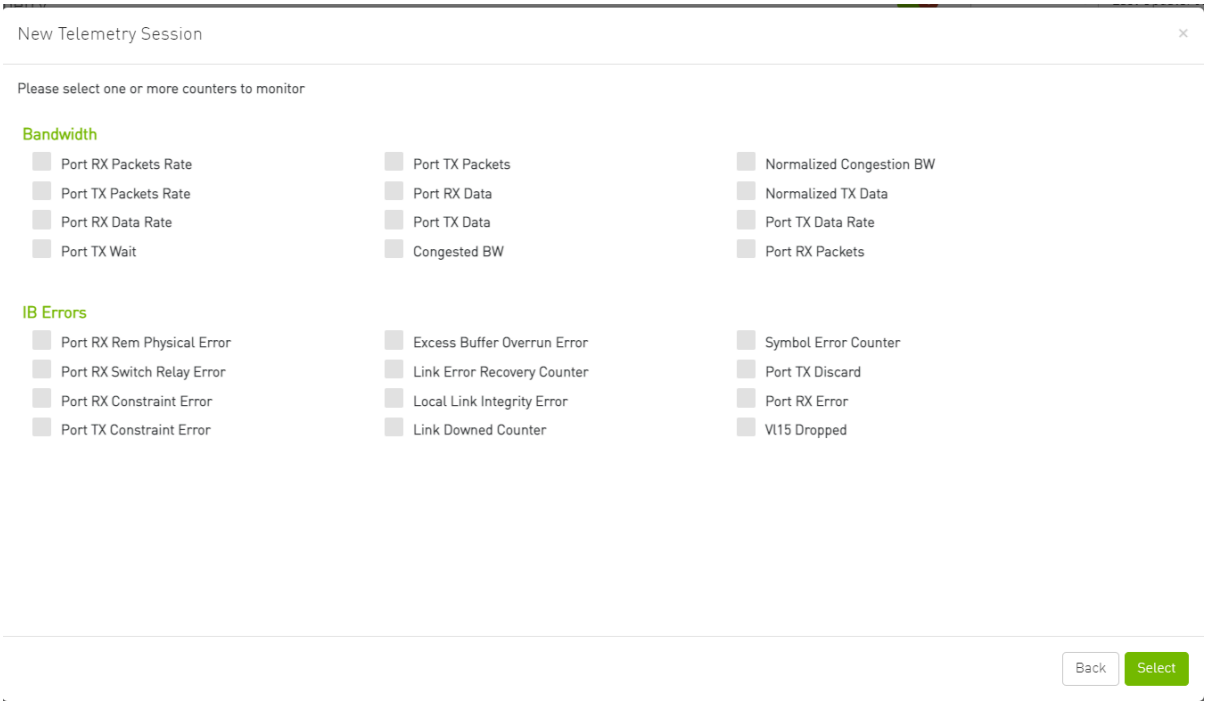
Note

In case the selected telemetry session is Top-X, only the ports are supported.

The user can select one or more counters from the counters dropdown menu:

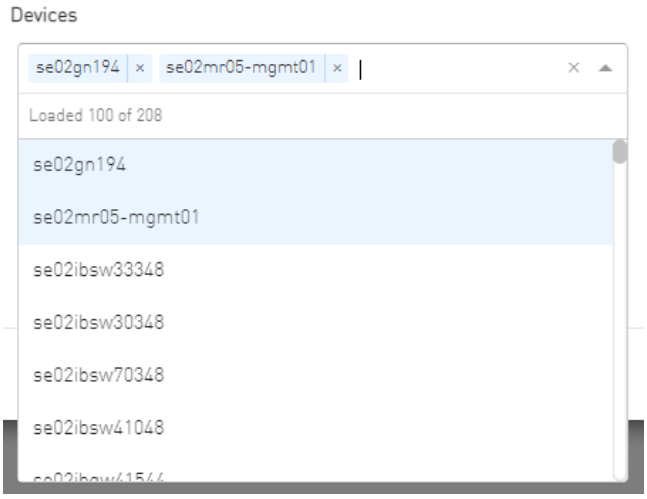


Alternatively, the user can get a full view of all the supported counters and select one or more by clicking on the "All Counters" button:



The user can select one or more devices/ports from the relevant dropdown menu:

- Devices:



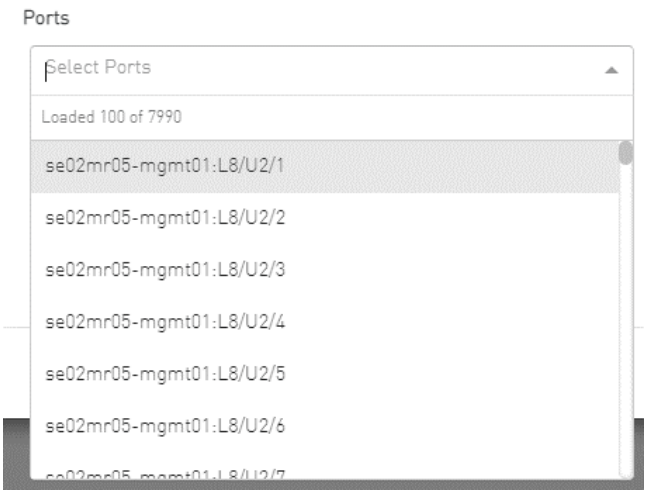
  Alternatively, the user can choose to get a full view of the devices by clicking on the "All Devices" button:

  Broken image

  com.atlassian.confluence.content.render.xhtml.XhtmlException: Missing required attribute: {http://atlassian.com/resource/identifier}value

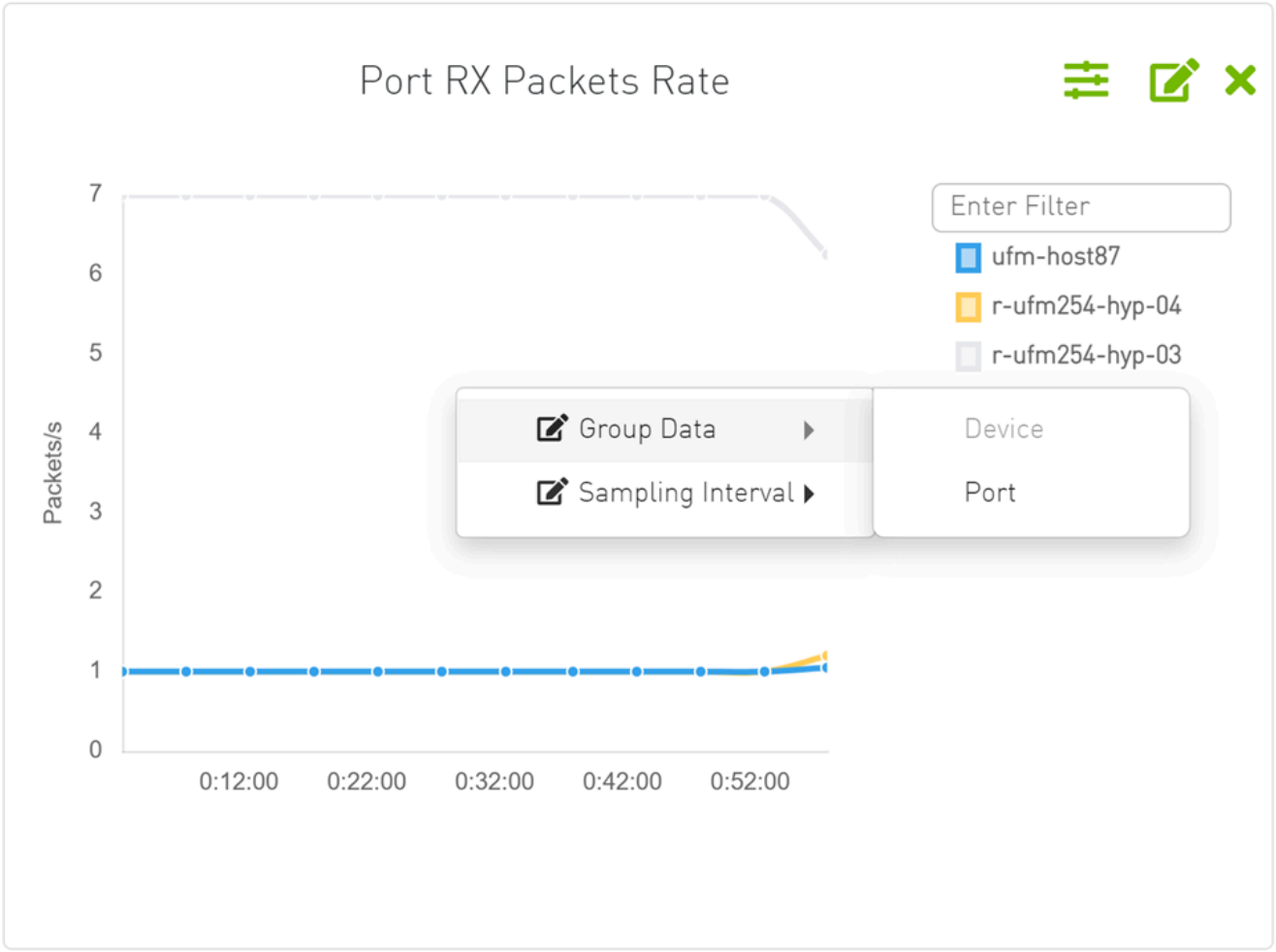
- Ports:

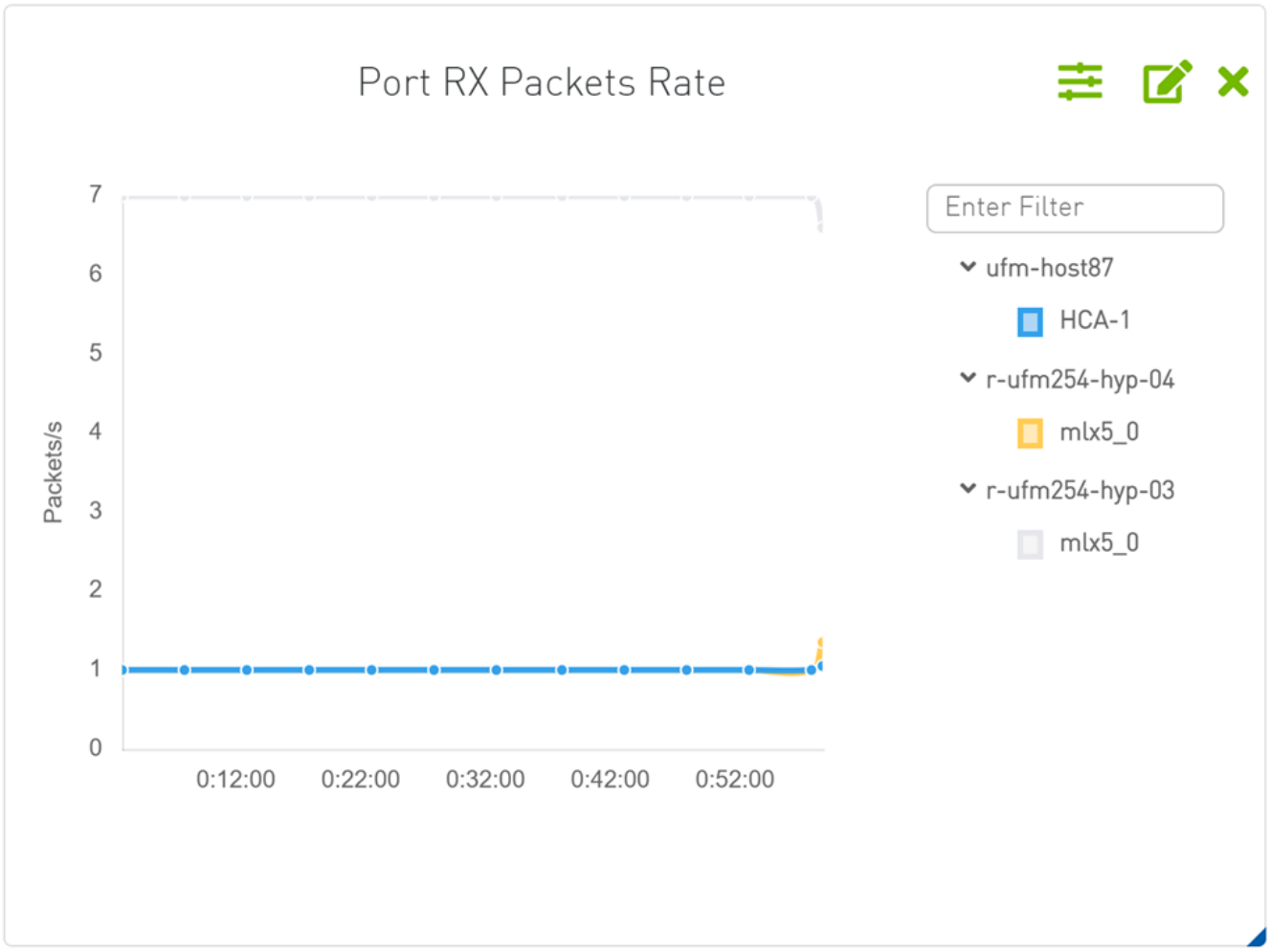
  After switching from "Devices" to "Ports," you user can view the ports' dropdown menu:



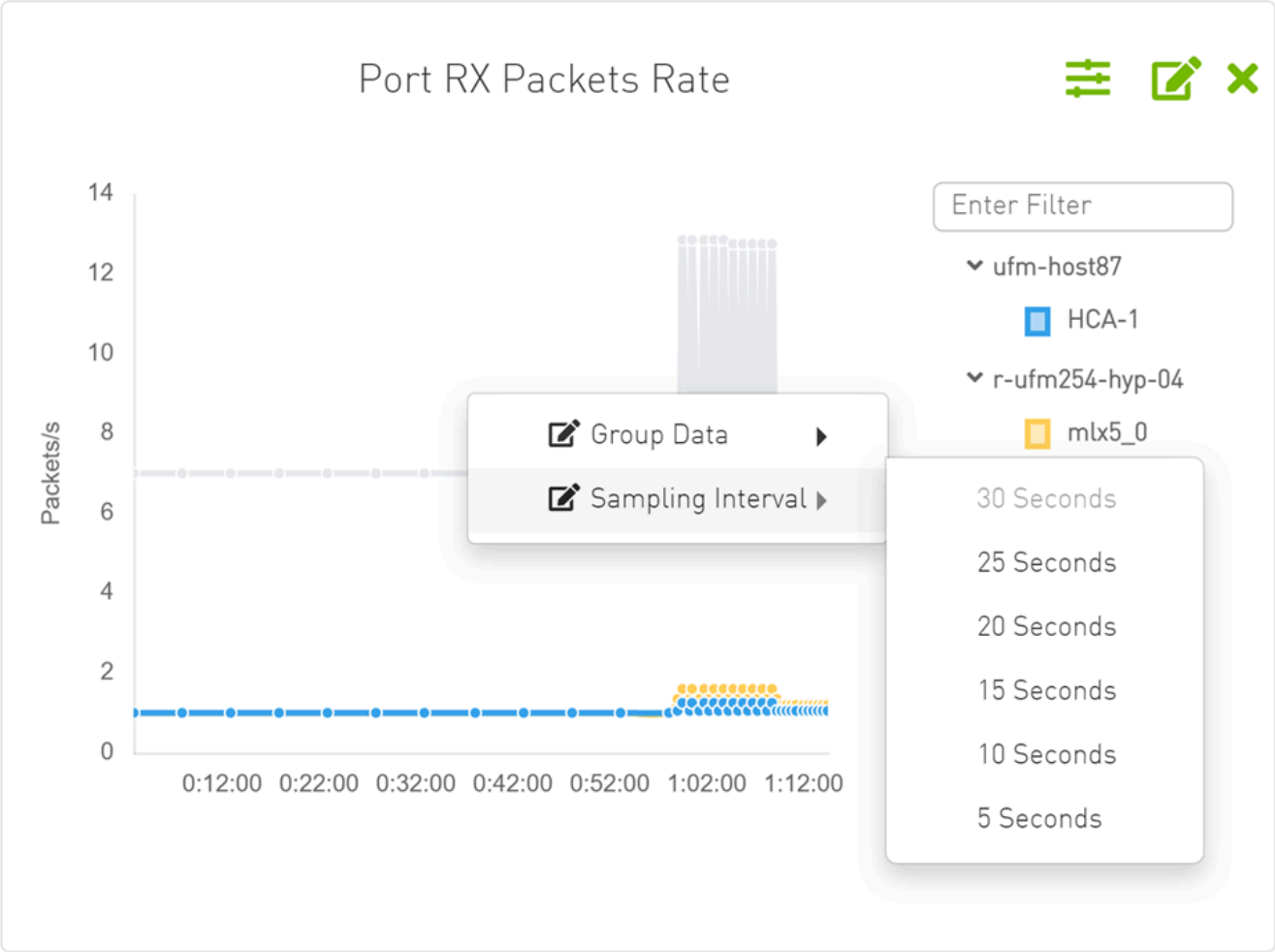
Alternatively, the user can choose to get a full view of the ports by clicking on the "All Ports" button.

Data aggregation can be changed in the timeseries panel by grouping the members by device or ports; this functionality is an option in the context menu. Therefore, if the timeseries panel is created with the "Devices" members, the panel shows each port in an individual line by right-clicking and then grouping by ports.



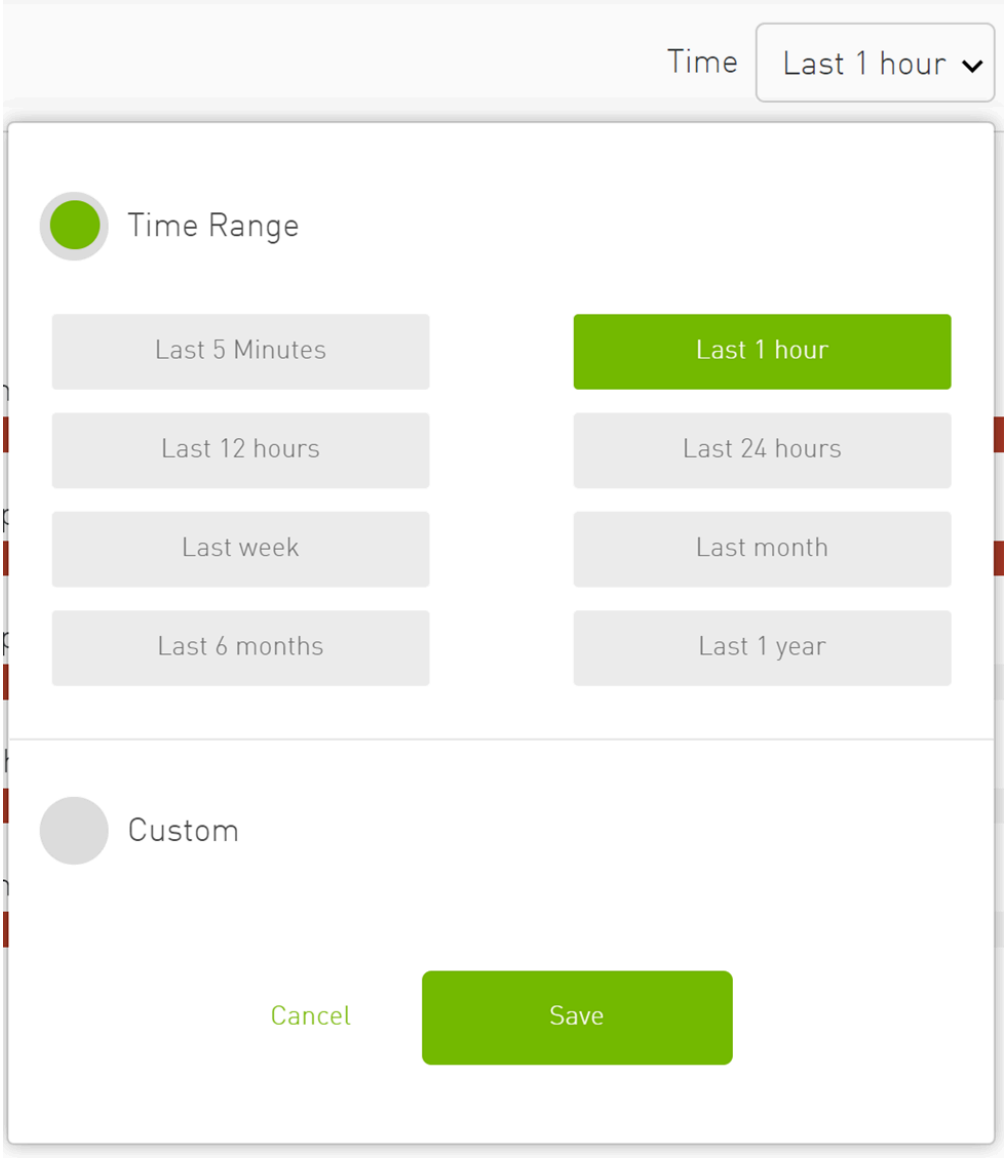


The Telemetry obtains live data from the server's each specific interval which equals the default session interval. The interval can be changed from the sampling rate option in the context menu.

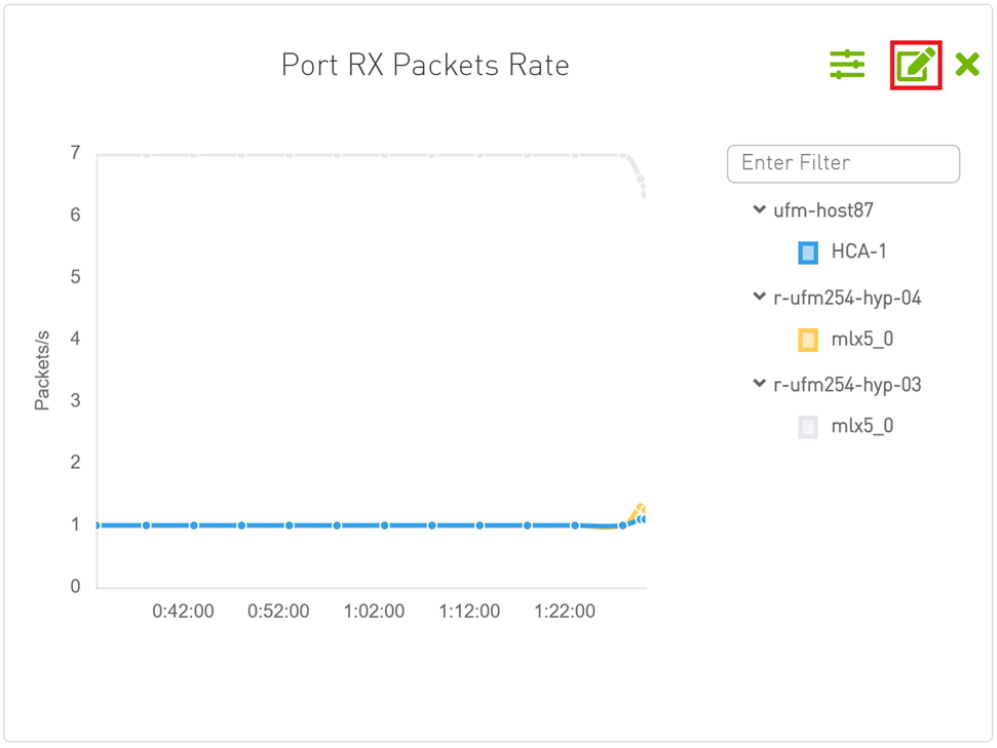


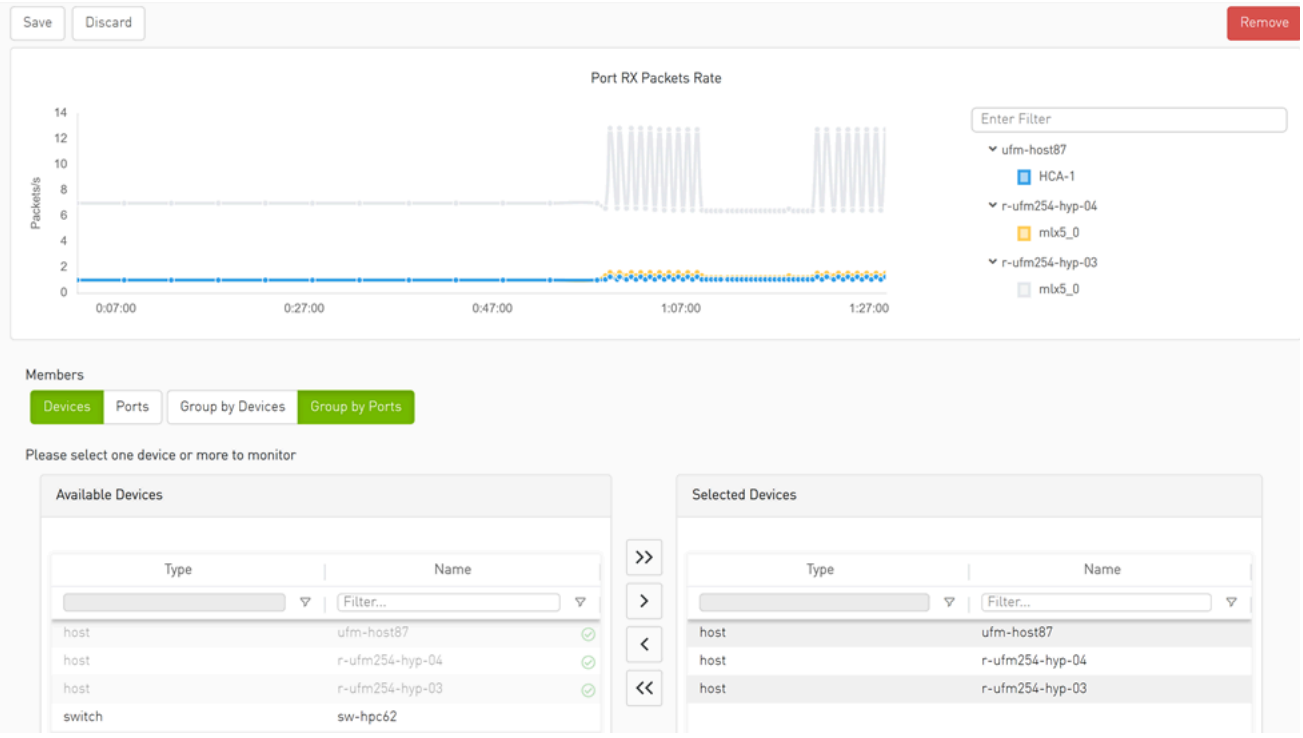
Time Range

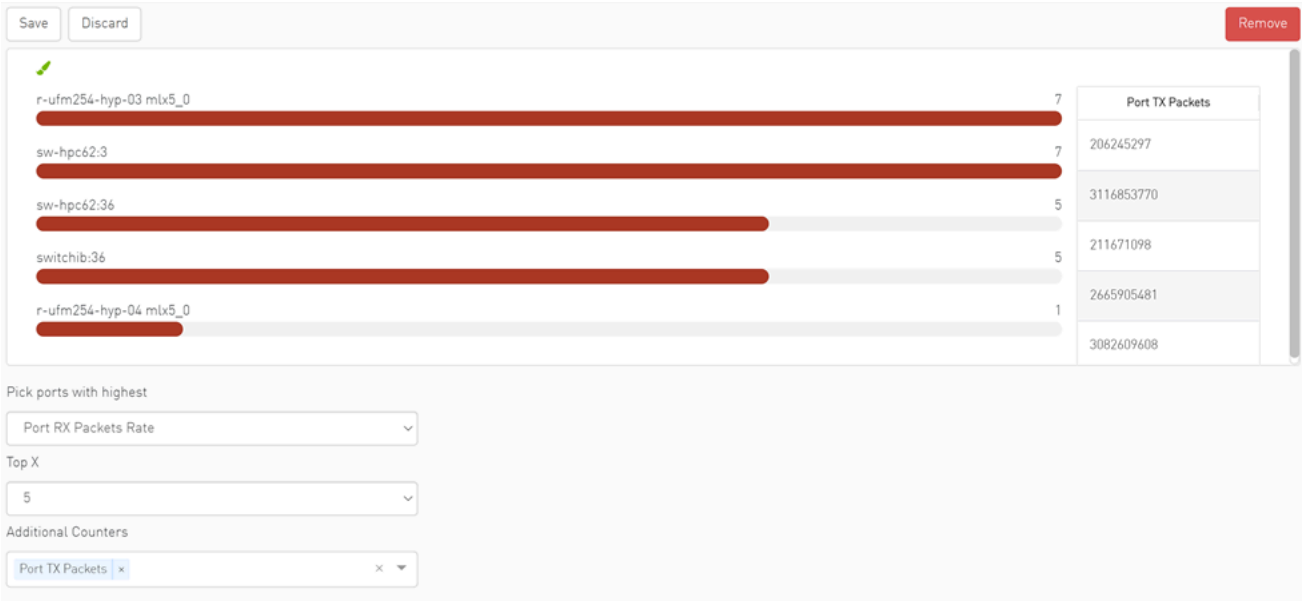
The starting time of timeseries panel can be changed from the time calendar at the top of the page, time can be "Time Range" or "Custom". In case the "Custom" option is chosen, only history data is shown.



The panel can be edited by changing members, members' type and grouping. The changes can be discarded or saved. The panel can also be deleted.

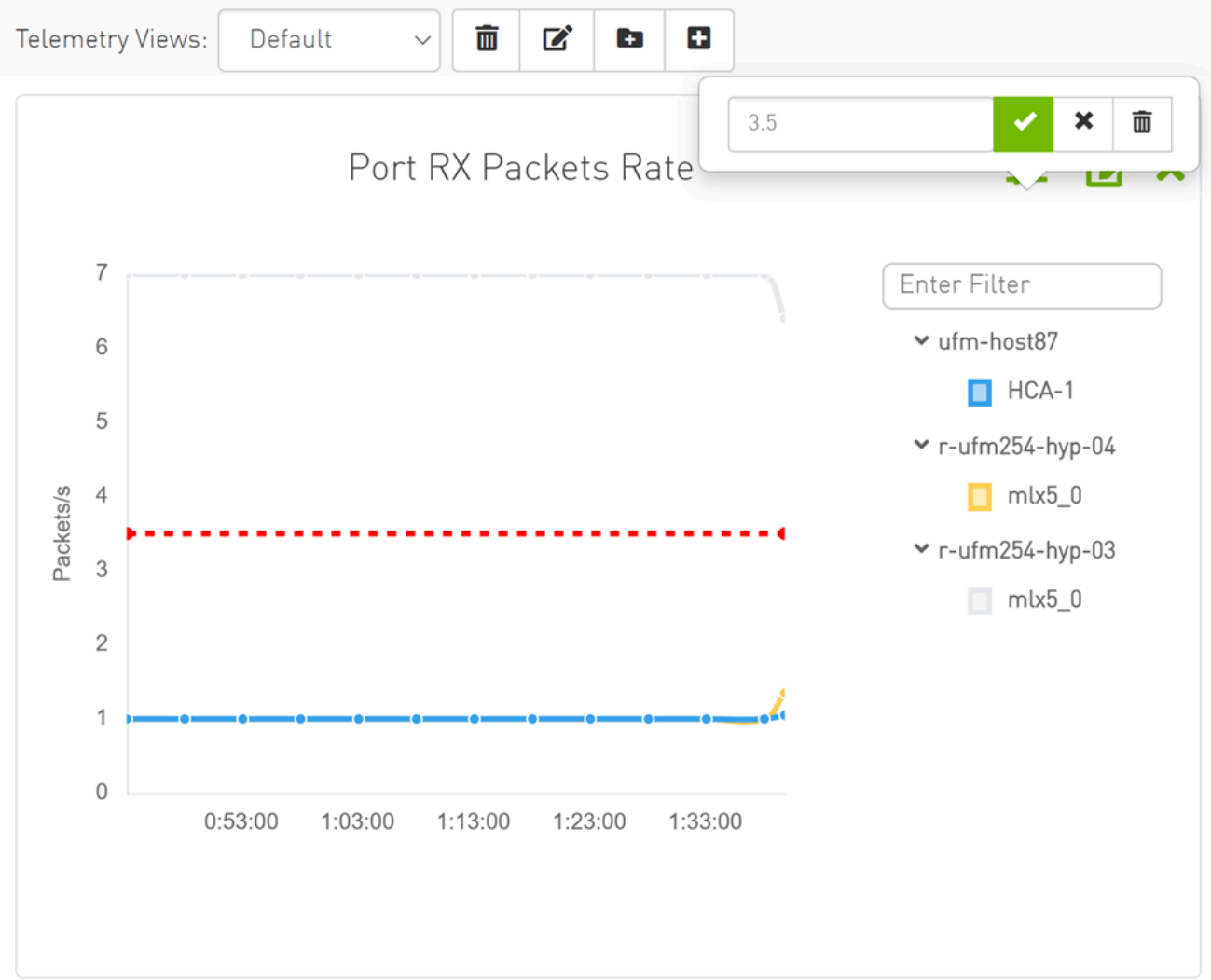

Port RX Packets Rate
Packets/s
7
6
5
4
3
2
1
0
0:42:00
0:52:00
1:02:00
1:12:00
1:22:00
Enter Filter
ufm-host87
HCA-1
r-ufm254-hyp-04
mlx5_0
r-ufm254-hyp-03
mlx5_0


Save
Discard
Remove
Port RX Packets Rate
Packets/s
14
12
10
8
6
4
2
0
0:07:00
0:27:00
0:47:00
1:07:00
1:27:00
Enter Filter
ufm-host87
HCA-1
r-ufm254-hyp-04
mlx5_0
r-ufm254-hyp-03
mlx5_0
Members
Devices
Ports
Group by Devices
Group by Ports
Please select one device or more to monitor
Available Devices
Selected Devices
Type
Name
Filter...
host
ufm-host87
host
r-ufm254-hyp-04
host
r-ufm254-hyp-03
switch
sw-hpc62



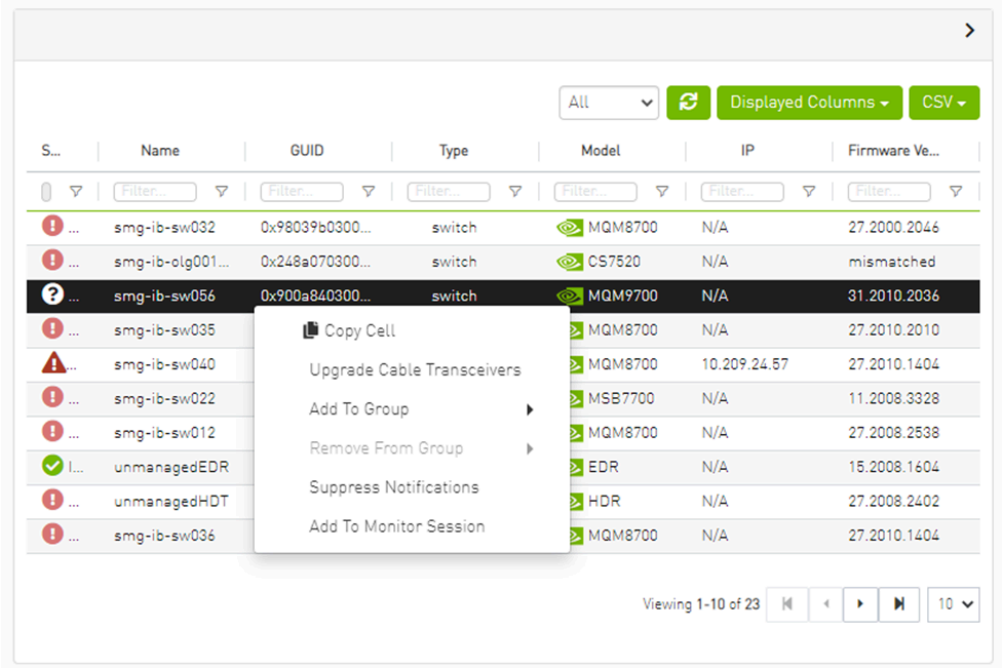
Threshold

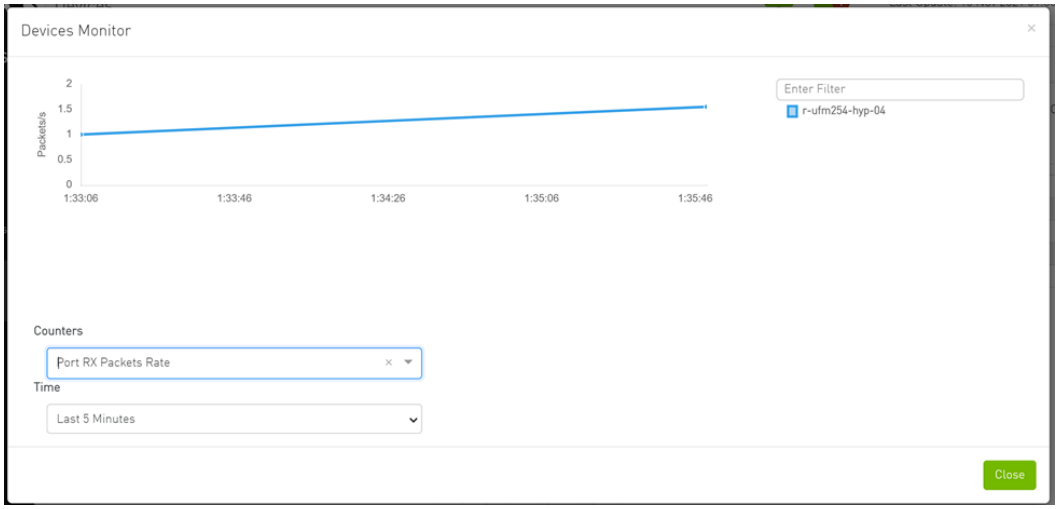
The threshold is supported in Telemetry as a line drawn at the threshold value.



Add Monitoring Session from Devices Table

In the Devices table, the user can see telemetry data for one or multiple devices as timeseries chart by clicking on the monitoring option in the context menu.





System Health

The System Health window enables running and viewing reports and logs for monitoring and analyzing UFM server and fabric health through the following tabs: UFM Health, UFM Logs, UFM Snapshot, Fabric Health, Daily Reports and Topology Compare.

- UFM Health Tab
- UFM Logs Tab
- UFM System Dump Tab

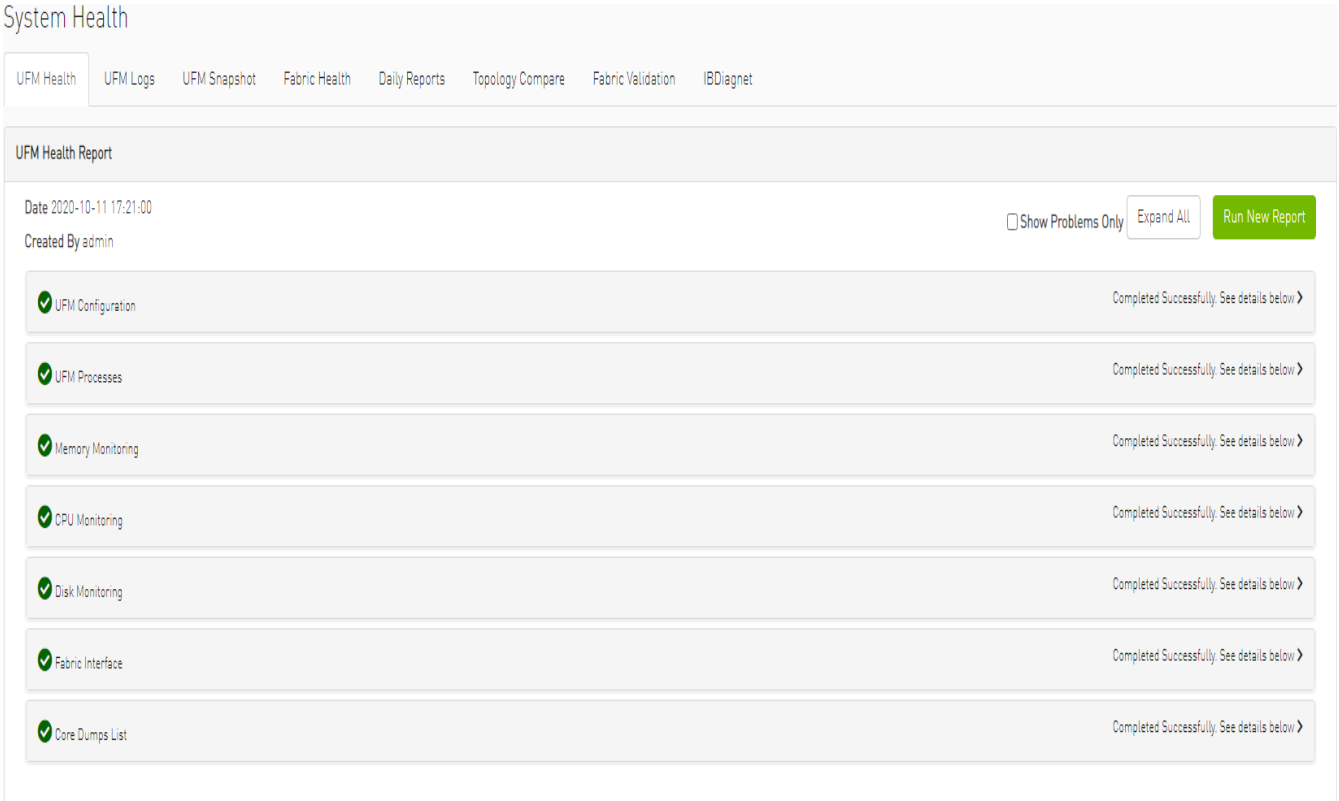
- Fabric Health Tab
- Daily Reports Tab
- Topology Compare Tab
- Fabric Validation Tab
- IBDiagnet Tab

UFM Health Tab

Through **UFM Health** tab, you can create reports that run a series of checks on the UFM server.

Each check that is run for a report triggers a corresponding event. Events are also triggered when a report starts and ends. For more information, see Events & Alarms.

To run a new report, click “Run New Report”. Results will be displayed inline automatically.



You can expand the results of each check or expand the results of all checks at once by clicking the "Expand All" button.

To view only the errors of the report results, click the "Show Problems Only" checkbox.

The following tables describe the checks included in the report.

UFM Health Report Checks

UFM Configuration	
Check	Description
Release Number	UFM software version and build.
License Type	Type of license, permanent or evaluation.
License Customer Number	The customer number provided by NVIDIA.
License UID	The UFM serial number provided by NVIDIA.
License Expiration Date	License expiration date for limited licenses.
License Functionality	Level of functionality enabled for the end-user, standard or advanced.
License Devices Limit	The maximum number of devices that UFM is licensed to manage. Note that it displays the current active and valid UFM licenses (not the sum of all valid licenses devices)
Running Mode	UFM running mode, Standalone or High Availability (HA). When UFM is in HA mode, additional information is displayed for the master and standby servers.

UFM Processing	
Check	Description
OpenSM	Status of the OpenSM service.
ibpm	Status of the ibpm (Performance Manager) service.
ModelMain	Status of the main UFM service.
httpd	Status of the httpd service.
MySql	Status of the MySql service.

Memory Monitoring	
Check	Description
Total memory usage	Percentage of total memory usage.
UFM memory usage	Percentage of UFM memory usage

CPU Monitoring	
Check	Description
Total CPU Capacity	Percentage of CPU capacity available
CPUs Number	Number of CPUs
Total CPU utilization	Percentage of total CPU utilization.
UFM CPU utilization	Percentage of UFM CPU utilization.

Disk Monitoring	
Check	Description
Disk <diskname>	Percentage of disk usage.

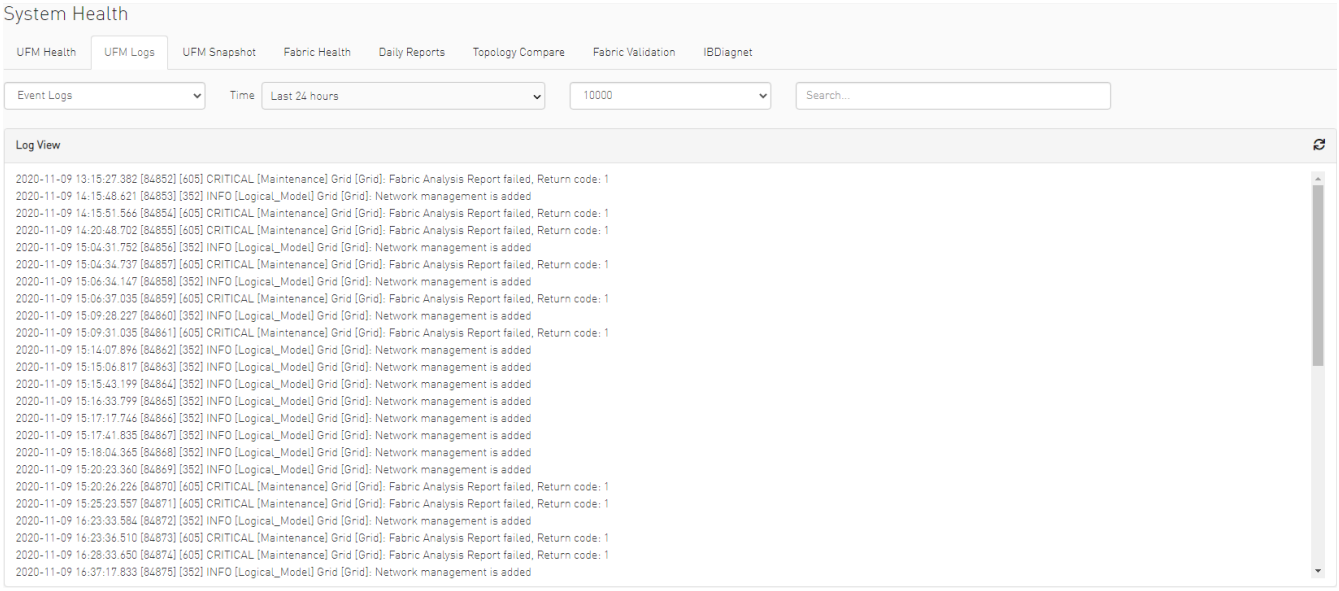
Fabric Interface	
Check	Description
Fabric Interface	Name and state of fabric interface.

UFM Logs Tab

UFM logging records events and actions that can serve to identify fabric and UFM server issues and assist in troubleshooting.

The logs are categorized into three files according to the activities they record: **Event** logs, **SM** logs, and **UFM** logs.

To view the log files, select the desired log file from the drop-down menu. Log data will be displayed:



In the Logs window, you can do the following:

- Refresh the data using the Refresh button on the right-hand side of the screen
- Search for a specific value using the Search bar
- Limit the display to a specific time period using the Time drop-down menu
- Limit the display to a specific number of lines using the drop-down menu (use "All" option to display all lines)
- Control the display of log occurrences by either showing all lines or hiding the duplicated ones.

Event Logs

Event Logs show the history of fabric events detected and initiated by the UFM server. The timestamp and severity of an event is indicated as well as the cause of the event and additional relevant information. *The Event log is kept on the UFM server in the /opt/ufm/log/events.log* file. Events can be configured whether to appear in the log files under the Events Policy tab in the Settings window. For more information, see Events Policy.

See "Appendix - Supported Port Counters and Events" for a comprehensive list of Events.

Subnet Manager (SM) Logs

SM Logs show messages of the Subnet Manager and communication plug-in.

The log verbosity is defined by selecting the Log Levels in the Subnet Manager tab under Settings window. For more information, see Subnet Manager Tab.

UFM Logs

UFM Logs is a general log of UFM Server. The log saves a history of user actions, events, polling results and other server activities and errors. Log verbosity is defined on start-up in the configuration file /opt/ufm/conf/gv.cfg:

```
[Logging]
# optional logging levels
#CRITICAL, ERROR, WARNING, INFO, DEBUG
level = WARNING
```

The default verbosity level is WARNING.

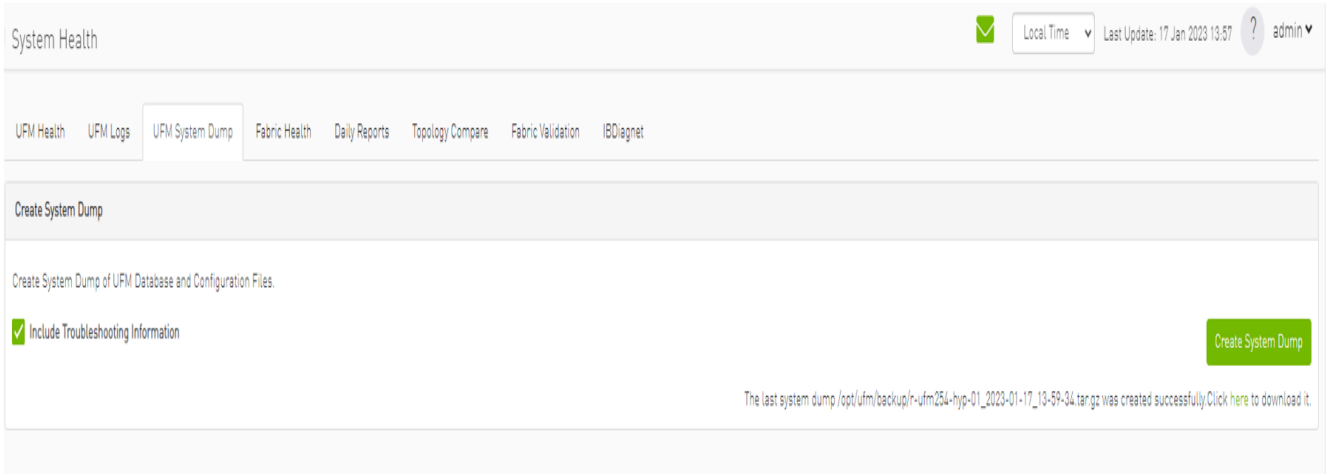
UFM System Dump Tab

You can export and save UFM database information, configuration and log files in a predefined location allowing you to create full system dump before upgrading, or for NVIDIA Enterprise Support.

By default, the system dump includes UFM database, UFM configuration, machine configuration and log files. You can also save troubleshooting information to send the required information for debugging with NVIDIA Enterprise Support. The additional troubleshooting information includes system snapshot files, system configurations and UFM reports.

To create a system dump, click the “Create System dump” button.

To extend the troubleshooting information for debugging purposes, check the "Include Troubleshooting" Information checkbox.

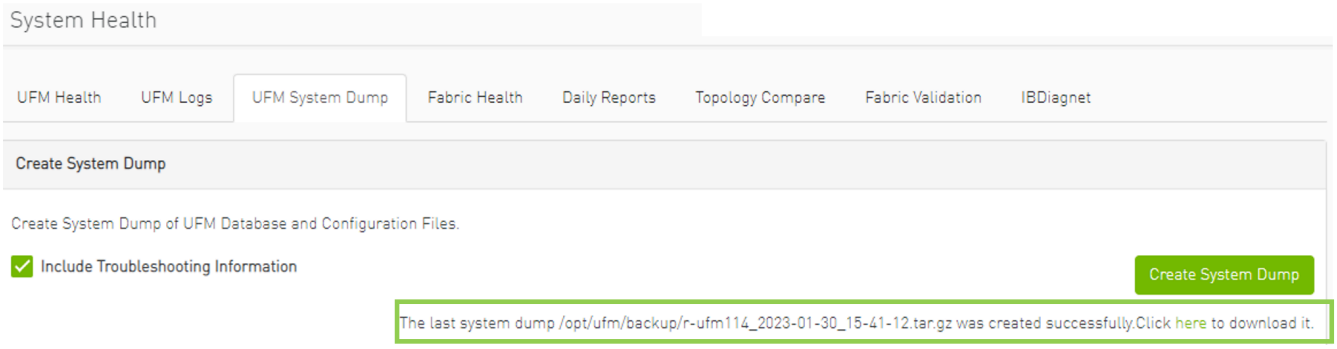


UFM will create the system dump and save the data to the predefined location. By default, the system dump files are stored under */opt/ufm/backup* directory. You can change the location of the system dump files in the *gv.cfg* configuration file in the backup folder location section.

For example:

```
#backup folder location
backup_folder=/opt/ufm/backup
```

In addition, if you did not switch from the tab, once the system dump creation process is complete, a download link will be available for downloading the system dump file directly to the user's machine, as shown in the below example:



The `ufm_sysdump` script can be employed to extract UFM system information. The script is located in diverse locations depending on the UFM installation method.

The `ufm_sysdump` can be run without any arguments. The default location of the script output depends on the installation method. To change the default location of the script output, add the `-o` argument and specify the desired script location (e.g. `ufm_sysdump -o <`output location`>`).

Additionally, the UFM script gathers the Cyber-AI and HA modules system dump output and stores it in the same tar file.

Location of the `ufm_sysdump` script is as follow:

- On baremetal/HA master or standby Modes: `/usr/bin/ufm_sysdump.sh`
- Standalone Mode: it is located in `/opt/ufm/files/scripts/ufm_sysdump.sh`

The default script output location:

- Baremetal Mode: backup folder `/opt/ufm/backup`
- Standalone Mode: backup folder inside the docker. Additionally, the working directory has been established for easier copying of the results
- HA master and standby Modes: `/tmp folder`

Fabric Health Tab

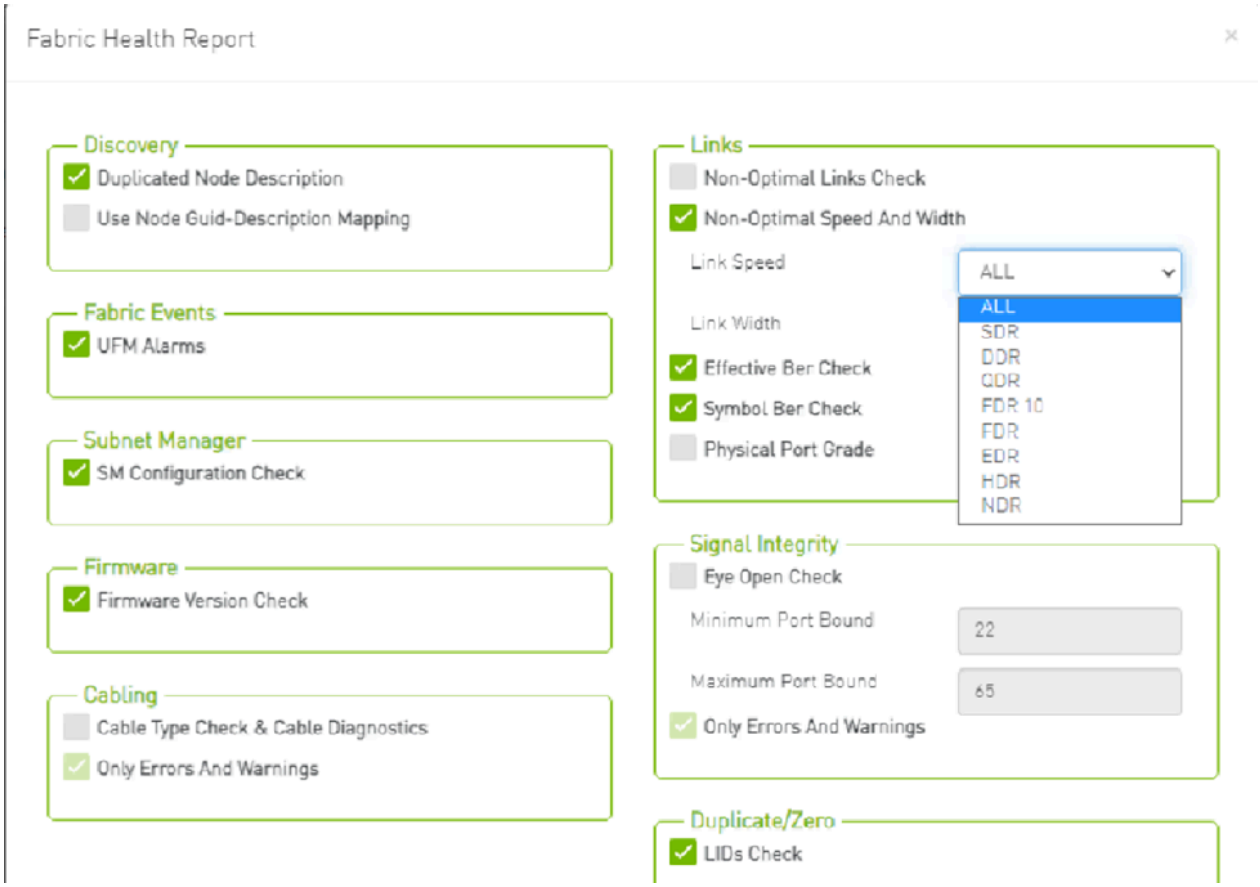
Through **Fabric Health** tab, you can create reports that run a series of checks on the fabric.

Each check that is run for a report triggers a corresponding event. Events are also triggered when a report starts and ends. For more information, see Events & Alarms.

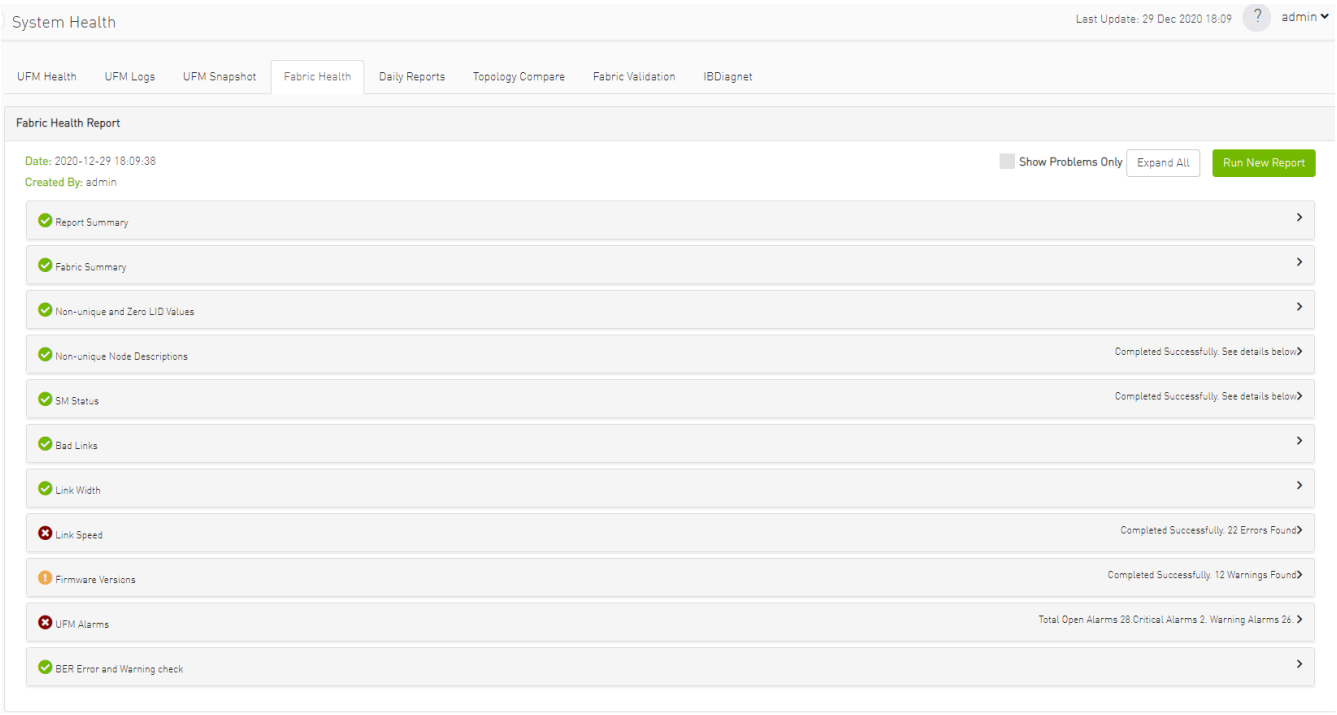
To run a new report, do the following:

1. Click “Run New Report."

2. Select the desired fabric health checks to run in the Fabric Health Report window and click “Run Report."



Results will be displayed automatically:

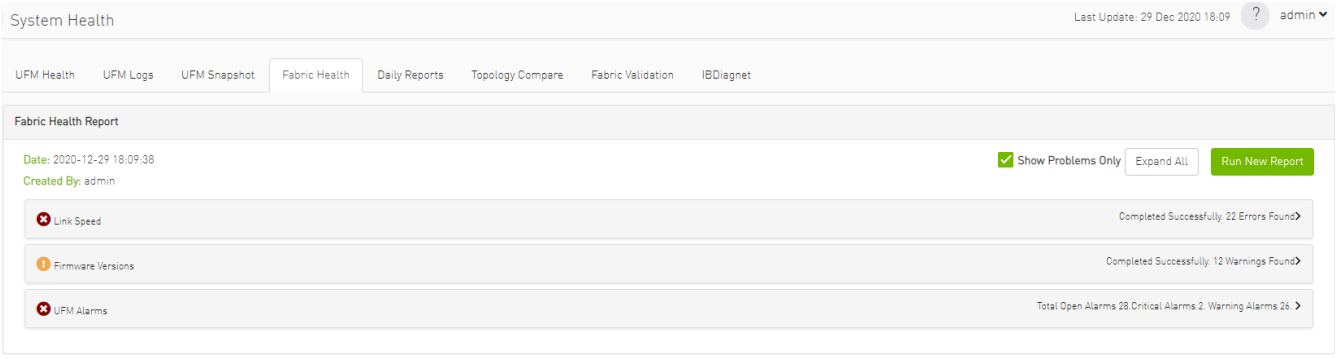


The report displays, the following:

- A report summary table of the errors and warnings generated by the report.
- A fabric summary of the devices and ports in the fabric.
- Details of the results of each check run by the report.

You can expand the view of each check or expand the view of all checks at once by clicking “Expand All."

To view only the errors of the report results, click the "Show Problems Only" checkbox.



The following table describes the checks included in the report.

Fabric Health Report Checks

Check	Description	To run, select:
Duplicate/Zero LID Check	Lists all ports with same LID or zero LID value.	LIDs Check Default: Selected
Duplicated Node Description	Lists all nodes with same node description. Does not include switches with the same description.	Duplicated Node Description Default: Selected
Use Node GUID-Description Mapping	Enables the usage of a mapping file (between node GUID and node description) when running duplicate node description analysis of the fabric. This file is located on the UFM server side at: */opt/ufm/conf/sm_guid_desc_mapping.cfg,* and uses the following format (node_guid → description): *0x248a070300702710 "Desc1"* *0x248a0703007026f0 "Desc2"* *0x0002c90300494100 "Desc3"*	**Use Node GUID-Description Mapping** Default: Unchecked **Note**: In order for this checkbox to be available, the Duplicated Node Description checkbox should also be selected. Otherwise, this checkbox will be greyed-out.
SM Check	Checks that: • There is one and only one active (master) Subnet Manager in the fabric. • The master is selected according to highest priority and lowest port GUID. The report lists all SMs in the fabric with their attributes.	SM Configuration Check Default: Selected
Bad Links Check	Performs a full-fabric discovery and reports “non-responsive” ports with their path.	Non-Optimal Links Check Default: Selected

Check	Description	To run, select:
Link Width	Checks if link width is optimally used. • When a width is selected, the report lists the active links that do not meet the optimum for the selection. • When no width is selected (All), the test checks whether the enabled width on both sides of the link equals the configured maximum (confirms that auto-negotiation was successful).	None-Optimal Speed and Width Default: Selected Link Width: The default is ALL.
Link Speed	Checks if link speed is optimally used. • When a speed is selected, the report lists the active links that do not meet the optimum for the selection. • When no speed is selected (All), the test checks whether the enabled speed on both sides of the link equals the configured maximum (confirms that auto-negotiation was successful).	None-Optimal Speed and Width Default: Selected Link Speed: The default is ALL.
Effective Ber Check	Provides a BER test for each port, calculates BER for each port and check no BER value has exceeded the BER thresholds. In the results, this section will display all ports that has exceeded the BER thresholds. Note that there are two levels of threshold: Warning threshold (default=1e-13) and Error threshold (default=1e-8).	Effective Ber Check Default: Selected
Effective Port Grade	Provides a grade per port lane in the fabric, which indicates the current port lane quality.	Physical Port Grade Default: Not Selected
Firmware Check	Checks for firmware inconsistencies. For each device model in the fabric, the test finds the latest installed version of the firmware and reports devices with older versions.	Firmware Version Check Default: Selected

Check	Description	To run, select:
Eye Open Check	(For QDR only) Lists Eye-Opener information for each link. When minimum and maximum port bounds are specified, the report lists the links with eye size outside of the specified bounds.	Eye Open Check Default: Selected Minimum and Maximum port bound: By default no bounds are defined.
Cable Information	Reports cable information as stored in EPROM on each port: cable vendor, type, length and serial number.	Cable Type Check & Cable Diagnostics Default: NOT selected because this test might take a long time to complete (40 msec per port)
UFM Alarms	Lists all open alarms in UFM.	UFM Alarms Default: Selected

Daily Reports Tab

The Daily Report feature collects, analyzes, and reports the most significant issues of the fabric in the last 24 hours (from 00:00 to 24:00). The reports present statistical information such as Summary of Traffic, Congestions and UFM events that occurred during the last 24 hours. These statistics are sent to a pre-defined recipients list on a daily basis. It is also possible to specify a non-24-hour range, by updating the UFM configuration file—see section Other Daily Report Configurations for details.

The following are the formats of the Daily Report:

- Interactive—opened via the browser. The charts are displayed in SVG format. This format can be accessed from the UFM Web UI and is also sent by email as an attachment (see Daily Report View in the Web UI section below).

- Static—opened via mail client (Outlook, Gmail, Hotmail, etc). The charts are displayed in PNG format.

Activating and Deactivating the Daily Report

Daily Report can be activated/deactivated via the `/opt/ufm/conf/gv.cfg` file.

> **Note**
>
> Daily Reports mechanism is activated by default.

To deactivate the Daily Report, do the following:

1. Open the `/opt/ufm/conf/gv.cfg` file.

2. Find the DailyReport section.

3. Set the daily_report_enabled option to false.

```
daily_report_enabled = false
```

To re-activate the Daily Report:, do the following:

1. Open the `/opt/ufm/conf/gv.cfg` file.

2. Find the DailyReport section.

3. Set the daily_report_enabled option to true.

```
daily_report_enabled = true
```

Saving Daily Reports

UFM saves the interactive Daily Reports under the `/opt/ufm/files/reports/Daily` directory. Each report will be saved under a directory with its respective date. For example,

report for Sept. 28th, 2014 will be located under: /opt/ufm/files/reports/Daily/2014-09-28/By default, the maximum number of reports that will be saved is 365 (one per day).

To configure the maximum number of reports to save, do the following:

1. Open the `/opt/ufm/conf/gv.cfg` file.

2. Find the DailyReport section.

3. Set the max_reports option to the desired value. A count of 0 (zero) means no copies are retained. (default and max is 365).

4. Restart UFM.

Other Daily Report Configurations

All the Daily Report configuration parameters can be found in the "DailyReport" section in gv.cfg configuration file.

The following are additional Daily Report configurations options:

- top_x option specifies the number of results in the "Top X" charts. Max number can be 20. (Default value is 10). top_x value will be applied to all charts existing in the Daily Report.

- mail_send_interval option specifies the epoch in minutes after midnight that the report can be emailed. By default, if UFM was down during midnight, and was restarted after 1:00, the report of the previous day will be generated and saved, but will not be emailed. This can be changed by editing the mail_send_interval. (default value is 60 minutes, meaning that the report will be send only between 00:00 to 1:00).

- log_level option specifies the Daily Report log verbosity. Default value is INFO (optional values: INFO, WARNING and ERROR).

- attach_fabric_health_report option indicates whether or not to add the fabric health report as attachment to the mail. Default value is true (optional values: true or false).

- fabric_health_report_timeout specifies the max time in seconds, to wait for fabric health report generation. Default value is 900 seconds (15 minutes).

In case of large fabrics, fabric health report might take longer than the default 15 minutes. User can enlarge the timeout for fabric health report to complete.

- **max_attached_file_size** specifies the maximum file size in Bytes for each email attachment that can be sent. Default value is 2 Megabytes.

  If the size of a certain file has exceeded this value, the file will not be sent as an attachment in the Daily Report mail.

```
[DailyReport]
# top_x specifies the number of results per each top x chart.
# max number can be 20.(default is 10)
top_x=10
# max_reports specifies the number of reports to save.
# A count of 0 (zero) means no copies are retained.(default and
max is 365)
max_reports = 365
#time interval in minutes after midnight
#when passed mail will not be sent
mail_send_interval=60
log_level = INFO
daily_report_enabled = true
attach_fabric_health_report = true
fabric_health_report_timeout = 900
# max attached file size in bytes, default is 2M (2097152 Bytes)
max_attached_file_size = 2097152
```

- **max_attached_file_size** specifies the maximum file size in Bytes for each email attachment that can be sent. Default value is 2 Megabytes.

- The **start_hour** and **end_hour** options enable selecting a sub-range of the day, during which, the relevant report data will be collected. Since by default this option is configured to collect data from the last 24 hours, the default start_hour is set to 0 (or 00), and the default end_hour is set to 24.

If these options are configured to different values, the generated report will include data from the specified interval only. The start_hour values range is 00 to 23, and the end_hour values range is 00 to 24. The specified end_hour must be greater than the specified

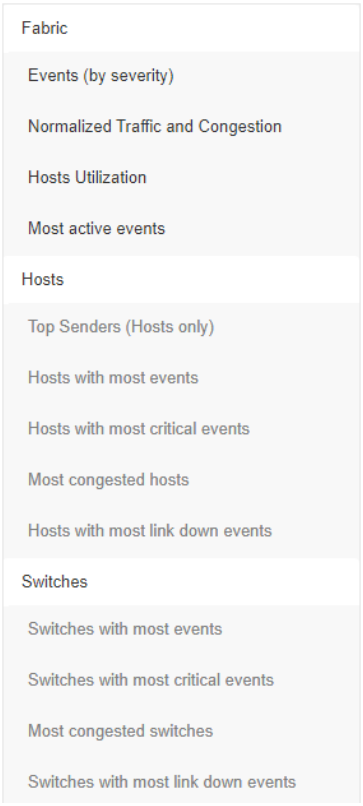
start_hour. If, for example, the start_hour is configured to 08, and the end_hour is configured to 10, the generated report will include data collected between 08:00-10:00 (excluding 10:00).

Report Content

Sidebar

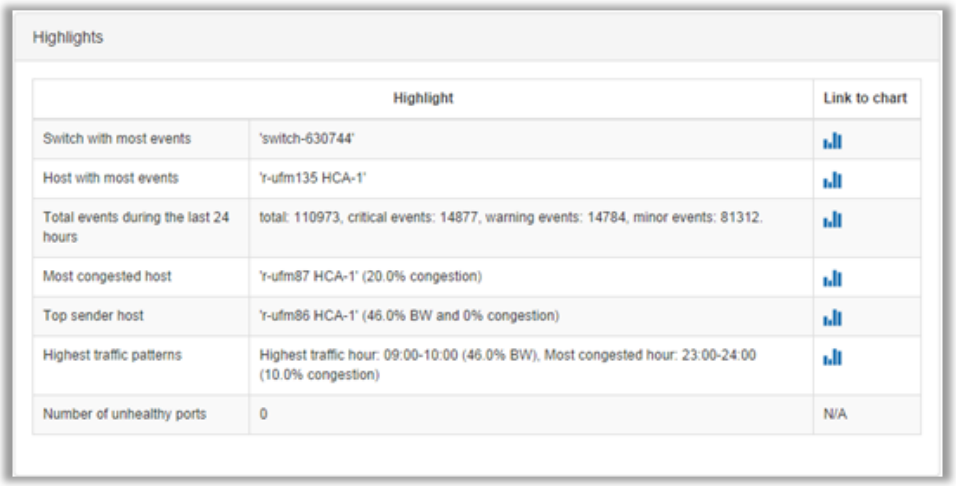
The Sidebar includes general information regarding the fabric, such as: the site name, number of switches and hosts in the fabric, and the dates on which the report was generated.

Navigation between the charts can be done via the menu charts on the sidebar.



Daily Report Highlights

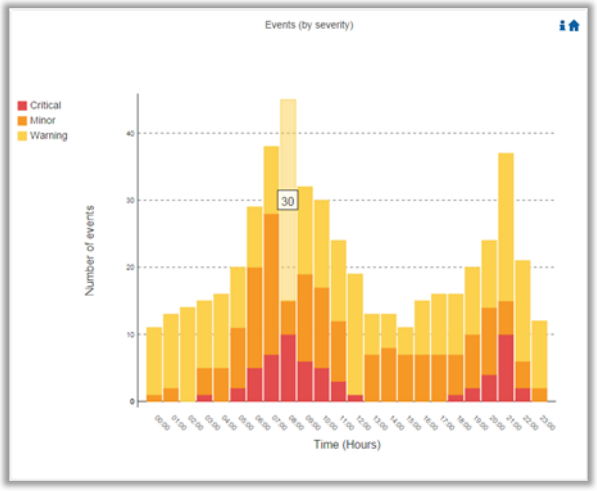
The top of the report shows highlight activities of the network, such as: the host with the most events, the most congested host and switch, and top sender host. To see the related chart of each highlight, click the corresponding icon in the "Link to chart column.



Available Charts

Events by Severity

Events by Severity displays in a graphical view the distribution of all the UFM events that occurred during each hour. Events are separated into the following severity levels: Critical, Minor, and Warning.



Note

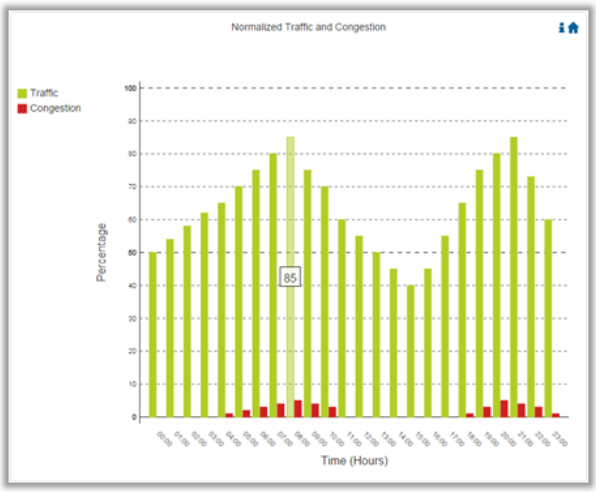
Hovering over the bars in the interactive report displays the amount of events per hour.

Normalized Traffic and Congestion

Normalized Traffic and Congestion displays in a graphical view the normalized traffic and congestions of the fabric. This graph displays the accumulated data for the Senders in the fabric (not including switches).

Congestion normalization is based on the number of delayed packets (packets that wait in the queue) and bandwidth loss.

The graph displays the percentage of the traffic utilization in green and the percentage of the congestion in red.



Note

Hovering over the bars in the interactive report displays the percentage of the traffic/congestion per hour.

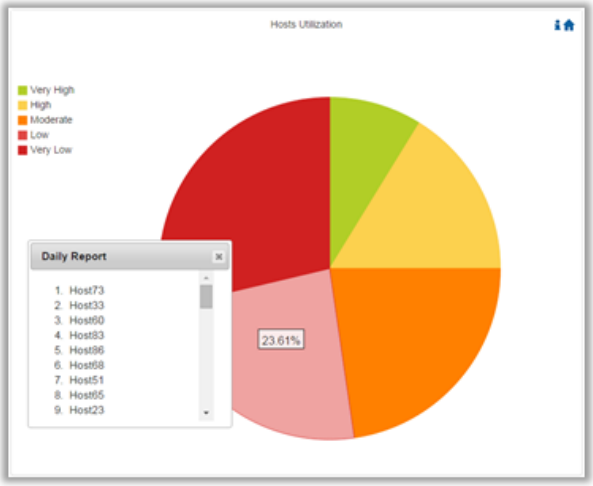
Hosts Utilization Distribution

Hosts Utilization Distribution displays in a graphical view the groups of hosts, where each host belongs to a specific group according to its utilization status.

To see the hosts in each group, click on the pie chart (at the interactive report).

The utilization groups are:

- Very low—up to 20% utilized
- Low—20–40% utilized
- Moderate—40–60% utilized
- High—60–80% utilized
- Very high—80–100% utilized

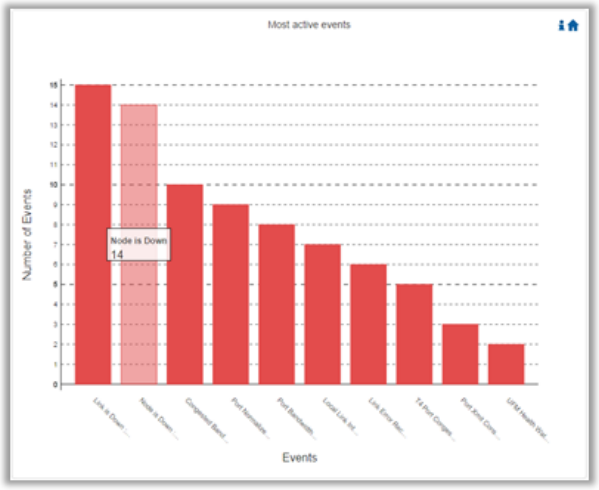


Note

Hovering over the slices in the interactive report displays the percentage of hosts in this group.

Most Active Events

Most Active Events displays in a graphical view the most active events, ordered by the number of occurrences during the last 24 hours.

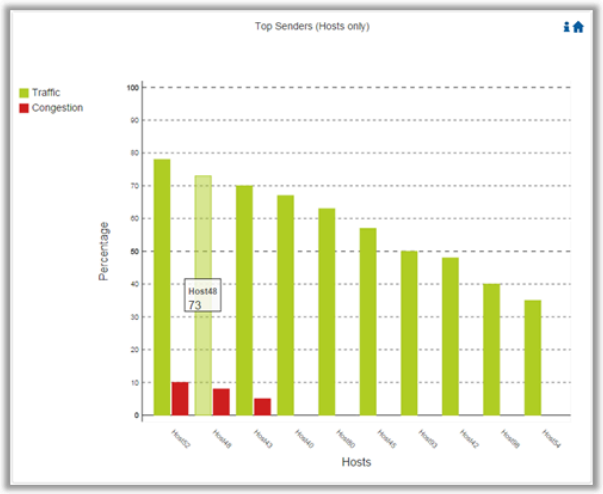


> **Note**
>
> Hovering over the bars in the interactive report displays the number of occurrences for each active event, and hovering on each event's name displays a tooltip with the event's description.

Top Senders

Top Senders displays in a graphical view the normalized traffic and congestions of the top sender hosts. Congestion normalization is based on the number of the delayed packets (packets that wait in queue) and bandwidth loss.

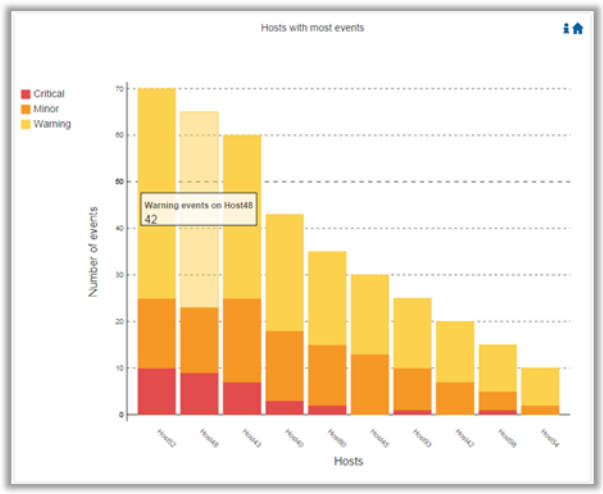
The graph displays the percentage of the traffic utilization in green and the percentage of the congestion in red.



> **Note**
>
> Hovering over the bars in the interactive report displays the percentage of the traffic/congestion for a selected host.

Hosts with Most Events

Hosts with Most Events displays in a graphical view the hosts with the most events. Events are separated into the following severity levels: Critical, Minor, and Warning.

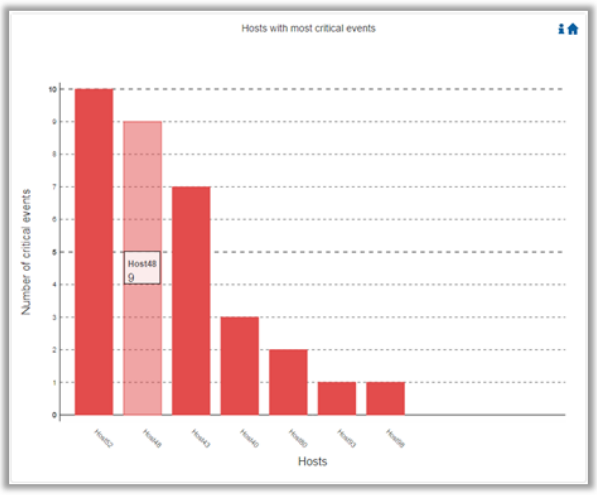


Note

Hovering over the bars in the interactive report displays the amount of events per severity for a selected host.

Hosts with Most Critical Events

Hosts with Most Critical Events displays in a graphical view the hosts with the most critical events.



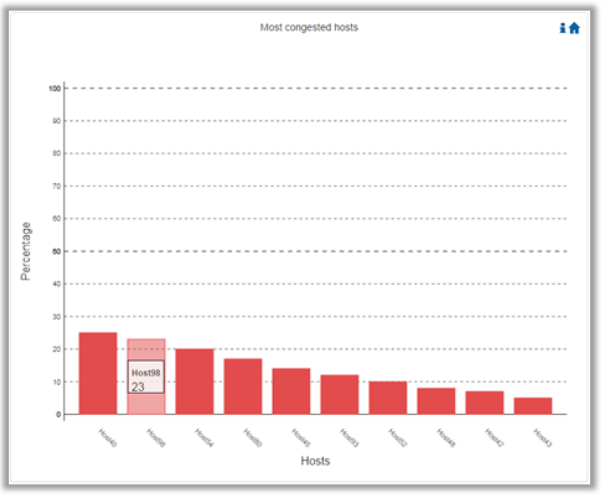
Note

Hovering over the bars in the interactive report displays the amount of critical events for a selected host.

Most Congested Hosts

Most Congested Hosts displays in a graphical view the normalized congestions of the most congested hosts. Congestion normalization is based on the number of the delayed

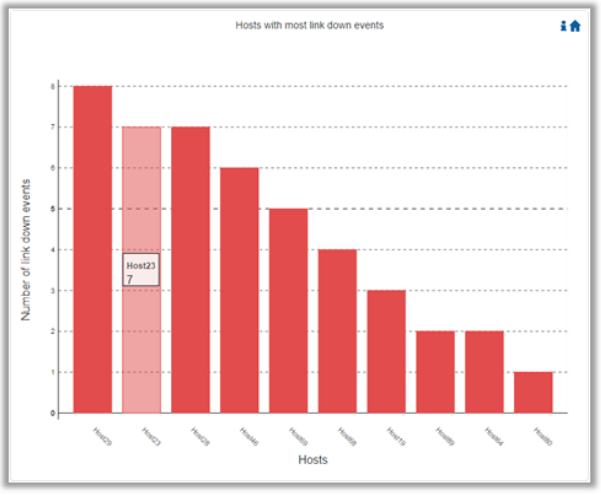
packets (packets that wait in queue) and bandwidth loss.



> **Note**
>
> Hovering over the bars in the interactive report displays the percentage of the congestion for a selected host.

Hosts with Most Link Down Events

Hosts with Most Link Down Events displays in a graphical view the list of the hosts with the most link down events during the last 24 hours.

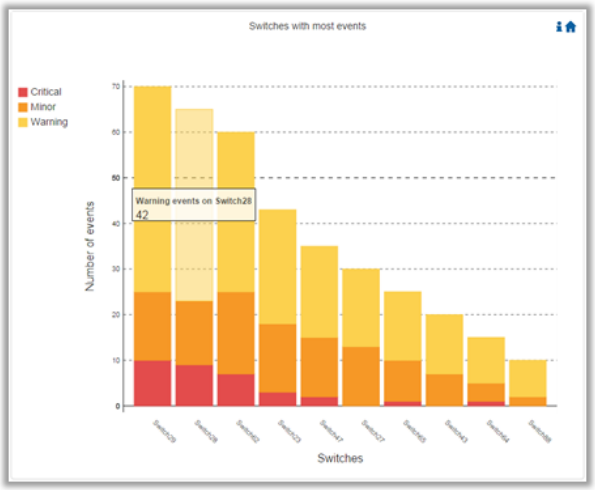


Note

Hovering over the bars in the interactive report displays the amount of link-down events for a selected host.

Switches with Most Events

Switches with Most Events displays in a graphical view the switches with the most events. Events are separated into the following severity levels: Critical, Minor, and Warning.

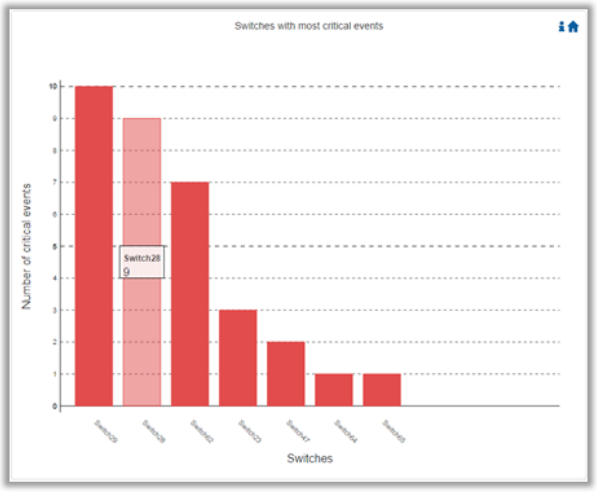


Note

Hovering over the bars in the interactive report displays the amount of events per severity for a selected switch.

Switches with Most Critical Events

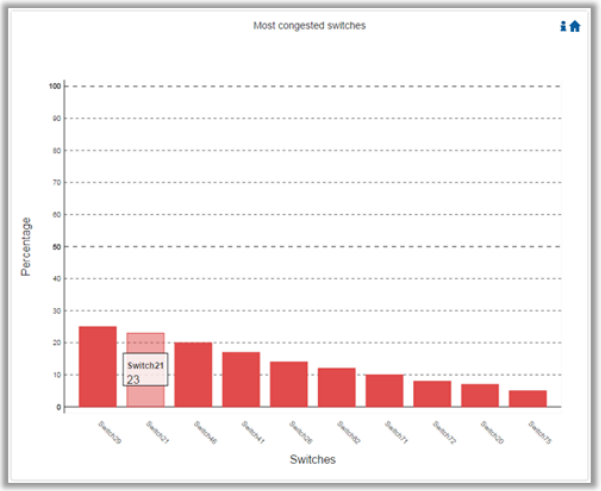
Switches with Most Critical Events displays in a graphical view the switches with the most critical events.



> **Note**
>
> Hovering over the bars in the interactive report displays the amount of critical events for a selected switch.

Most Congested Switches

Most Congested Switches displays in a graphical view the normalized congestions of the most congested switches. Congestion normalization is based on the number of delayed packets (packets that wait in queue) and bandwidth loss.

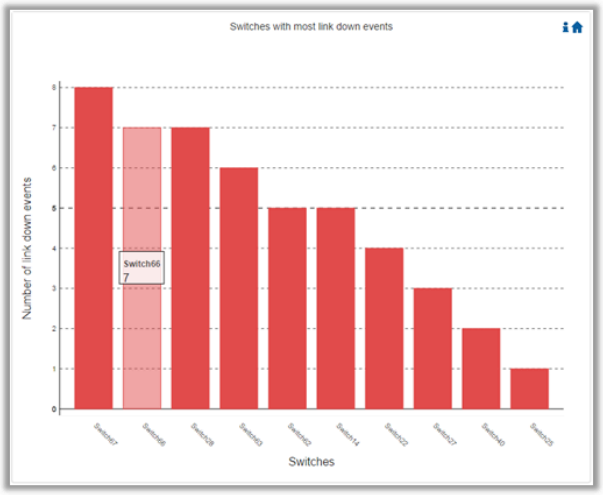


Note

Hovering over the bars in the interactive report displays the percentage of the congestion for a selected switch.

Switches with Most Link Down Events

Switches with Most Link Down Events displays in a graphical view the list of the switches with the most link down events during the last 24 hours.



Note

Hovering over the bars in the interactive report displays the amount of link-down events for a selected switch.

Note

Clicking on the “help” icon in the upper right corner of each chart, in the interactive report , will display a short description of the chart.

Clicking on the “home” icon in the upper right corner of each chart, in the interactive report , will move the display to the beginning of the report.

Note

On charts: “Events by Severity”, “Hosts with Most Events”, and “Switches with Most Events”, if the maximum value in the Y-axis is less than 5, an “m” unit will appear and stand for “milli".

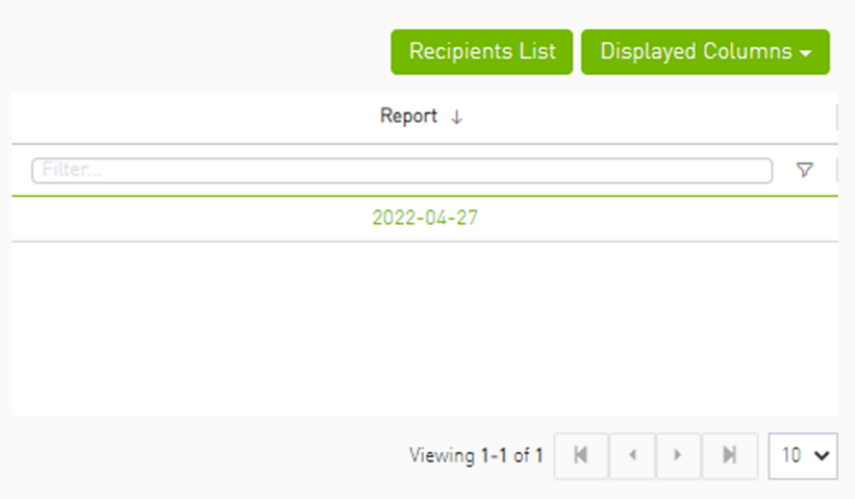
Note

For all charts, if the value is higher than 1000 in the Y-axis, a “k” unit will appear and stand for “killo”.

Daily Report View in the Web UI

In this tab, you can select the UFM daily reports that you wish to view and you can specify the recipients to which these daily reports will be sent.

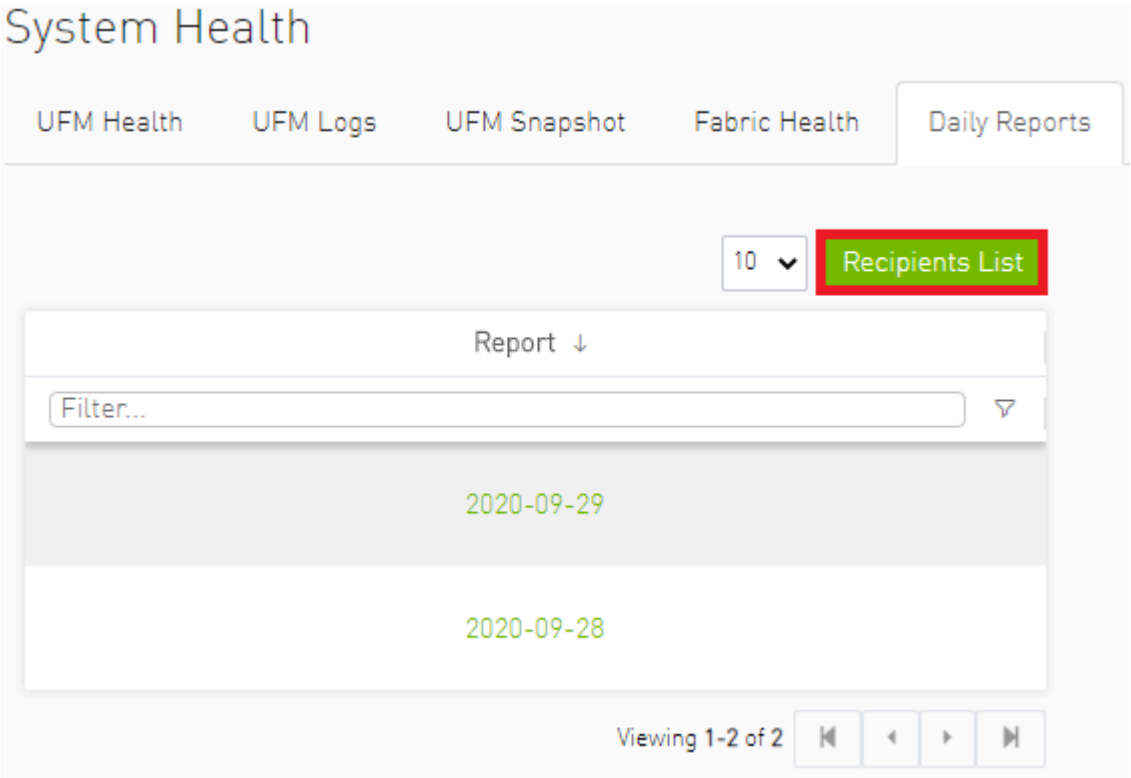
To view a specific daily report, click the relevant report date from the list of available daily reports.



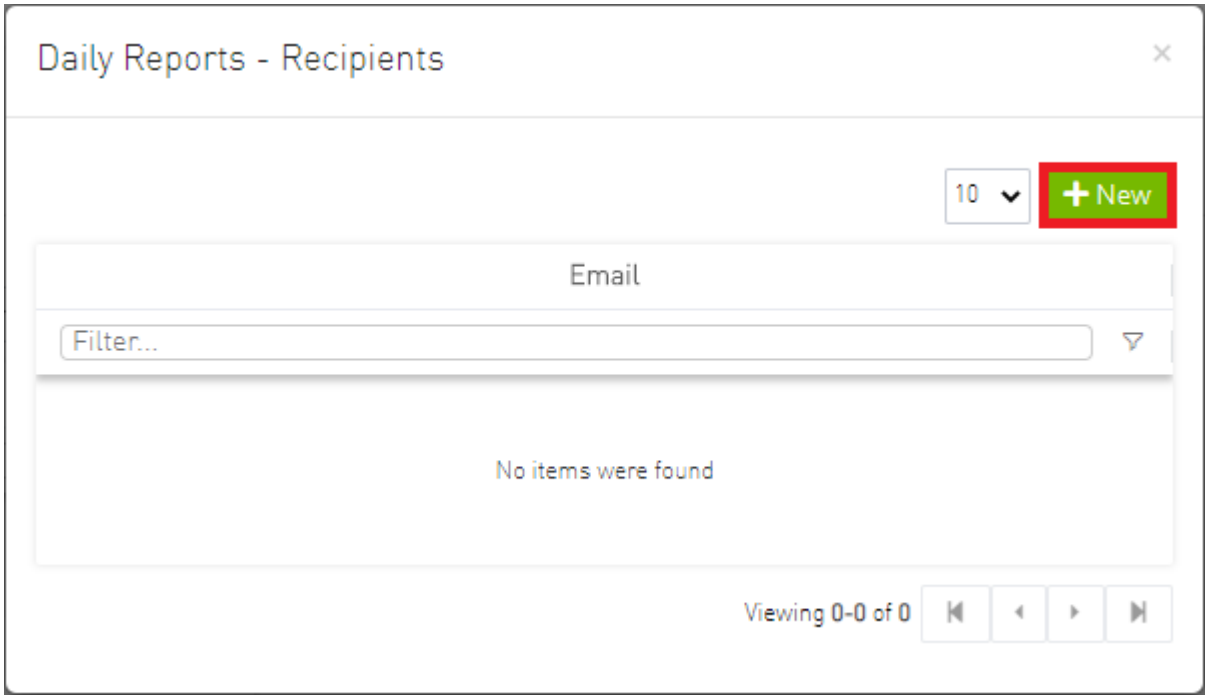
The specified report content will be displayed when clicking the report (see Activating and Deactivating the Daily Report).

➢ ***To configure the Recipients list for the daily reports, do the following:***

1. Click **Recipients List** under System Health → Daily Reports tab.

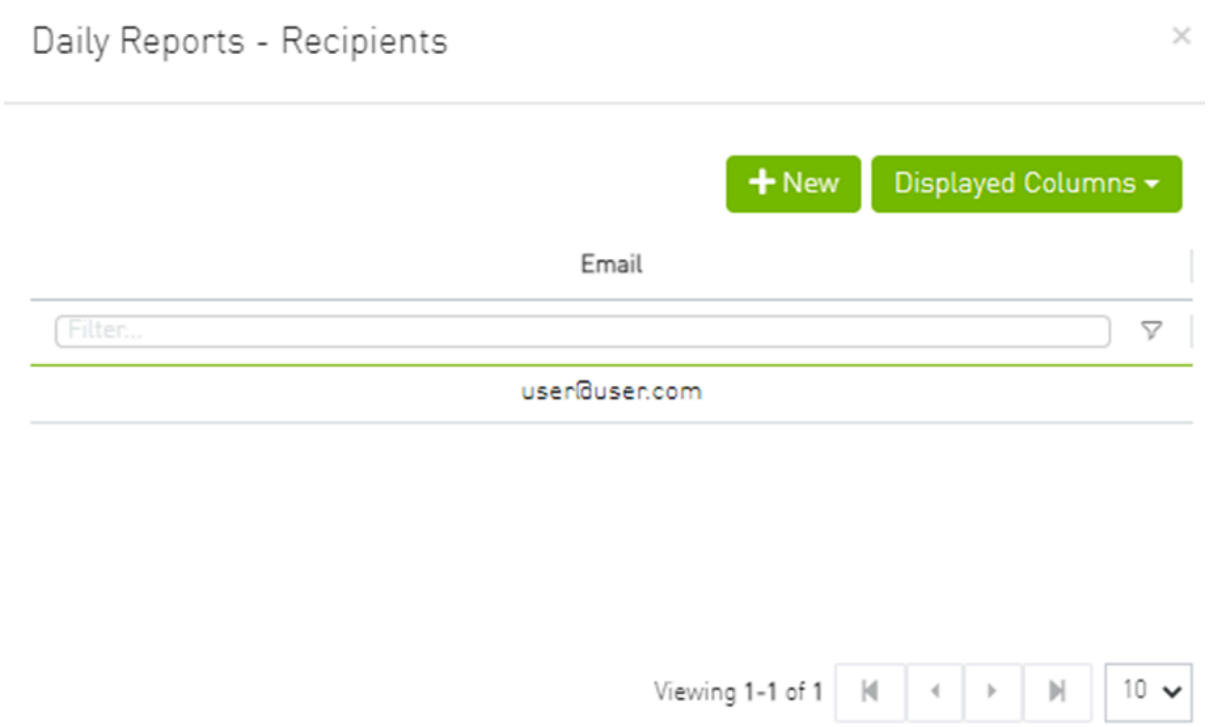


2. Click **New**.



3. In the Recipients List window, enter valid recipient email addresses, comma-separated, and click **Submit**.

The new recipient/recipients will be added to the Daily Reports Recipients list.



These recipients will automatically start receiving the UFM daily reports.

Topology Compare Tab

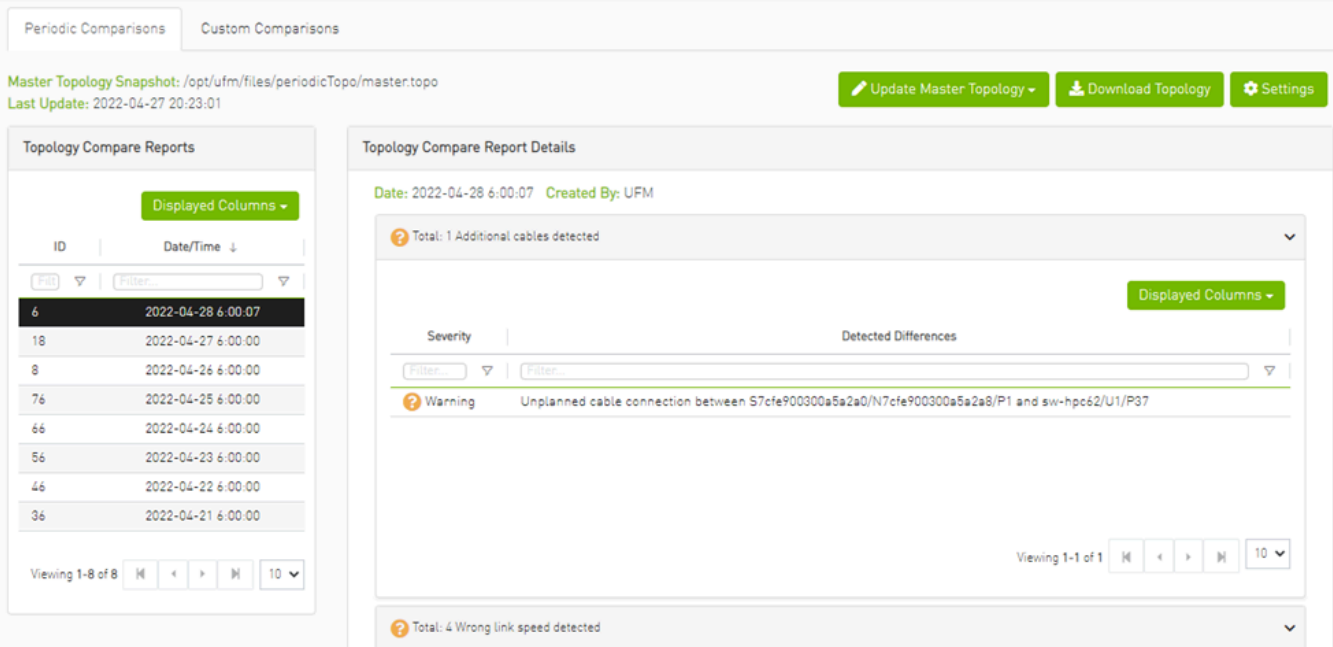
Overview

The Topology Compare tab allows two methods of topology comparison:

- Periodic Comparison
- Custom Comparison

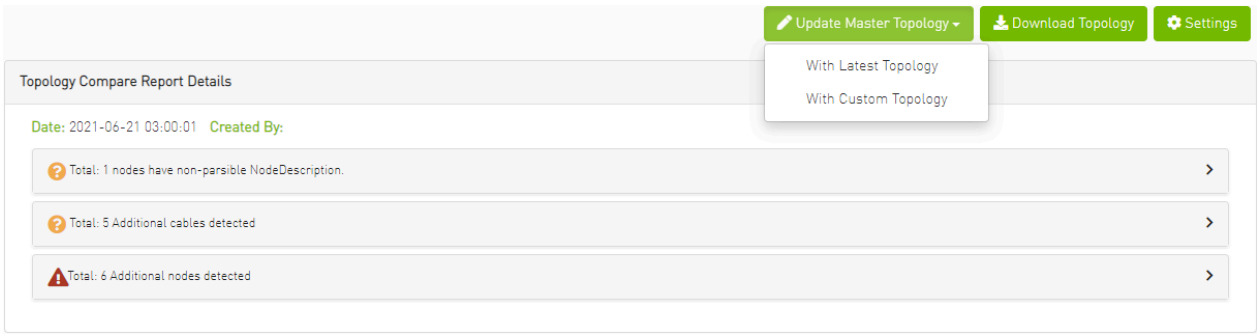
Periodic Comparison

Periodic comparison allows users to compare the current fabric topology with a preset master topology. The master topology may be set either by selecting the current topology or uploading a predefined custom topology.



When a report is selected from the "Topology Compare Reports" table, its result are displayed on the right side under "Topology Compare Report Details".

- To update the master topology with the latest (current) topology or a custom topology saved in external file, click the "Updated Master Topology" dropdown button.



- To download the current topology as a `.topo` file, click the "Download Topology" button.

- The Settings button navigates to the Topology Compare tab of the Settings view which allows users to configure periodic comparison settings.

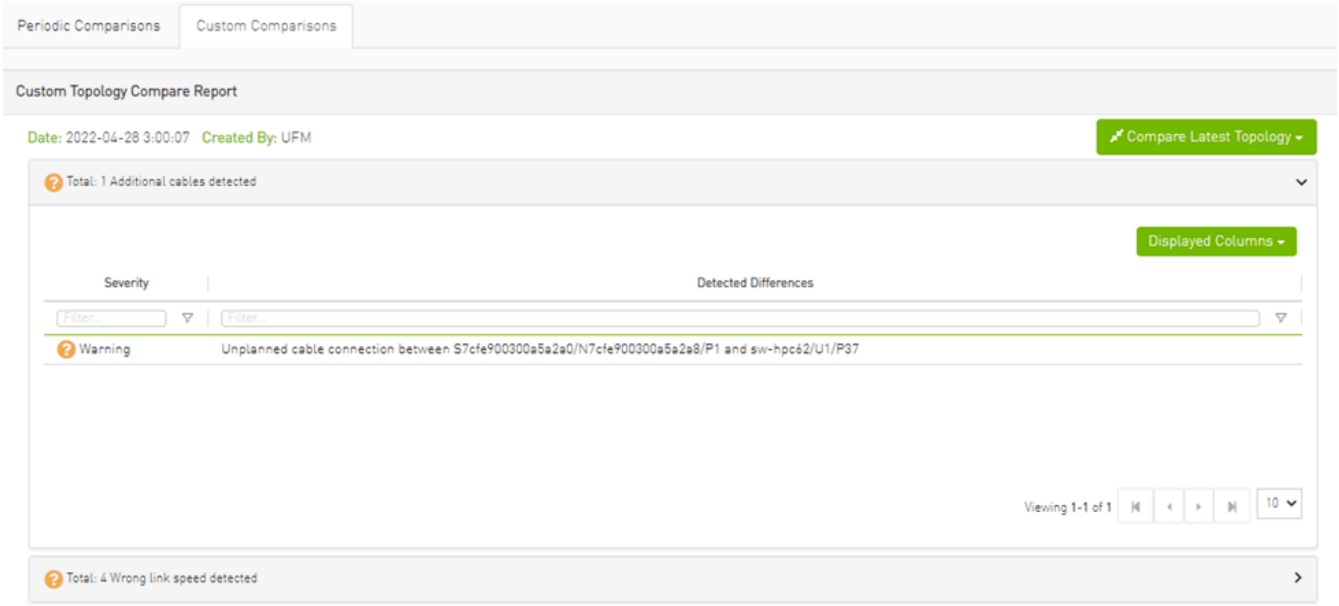
Custom Comparison

Custom comparison compares user-defined topology with the current fabric topology. UFM compares the current fabric topology to a topology snapshot (of the same setup) and reports any differences between them.

To be able to use the UFM topology comparison mechanism, first you need to create a TOPO file that defines the current topology of the fabric.

> **Info**
>
> Ideally, the topology snapshot (`.topo` file) should be taken after the setup bring-up phase has been completed so that no more topology changes are expected to take place.

Once the TOPO file is created, you can use the topology comparison mechanism to compare the current fabric topology to the one in the TOPO file and view their differences (if found).



To compare the current topology with the master topology or a custom topology (external file), make a selection from the "Compare Latest Topology" dropdown button and upload the `.topo` file to compare against.

Topology Comparison Flow

To create the topology file for later comparison with the current topology, do the following:

1. Verify that the following path for ibdiagnet ibnl directory exists: `/opt/ufm/tmp/ibdiagnet.out/tmp/ibdiag_ibnl`. If the path does not exist, make sure to create it manually.

2. Run the following command on the UFM server machine to create the topology file (`mytopo.topo`). Note that the file extension must be `.topo` for UFM to recognize it.

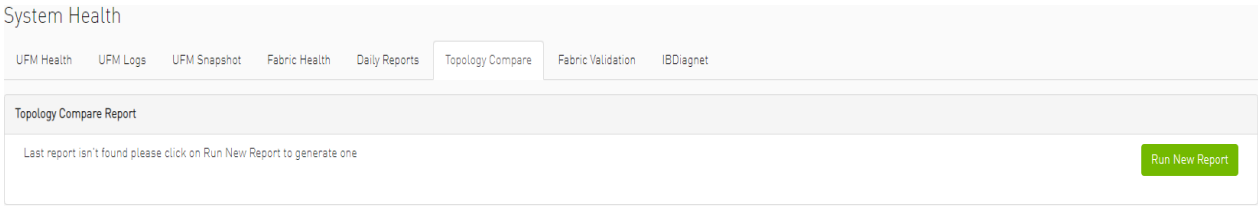
   ```
   /opt/ufm/opensm/bin/ibdiagnet -w /tmp/mytopo.topo
   --out_ibnl_dir /opt/ufm/tmp/ibdiagnet.out/tmp/ibdiag_ibnl
   ```

   Once command execution is completed, the new topology file (`/tmp/mytopo.topo`) will be created and can be used for later comparison with the current fabric topology. Also, several .ibnl files that were (optionally) created will be found in the defined output directory (`/opt/ufm/tmp/ibdiagnet.out/tmp/ibdiag_ibnl`). These `.ibnl` files will be used when comparing any topology file to the current fabric topology.

   At any time during your UFM session, you can view the last generated report through the UFM web UI or in HTML format in a browser window.

To perform topology comparison, do the following:

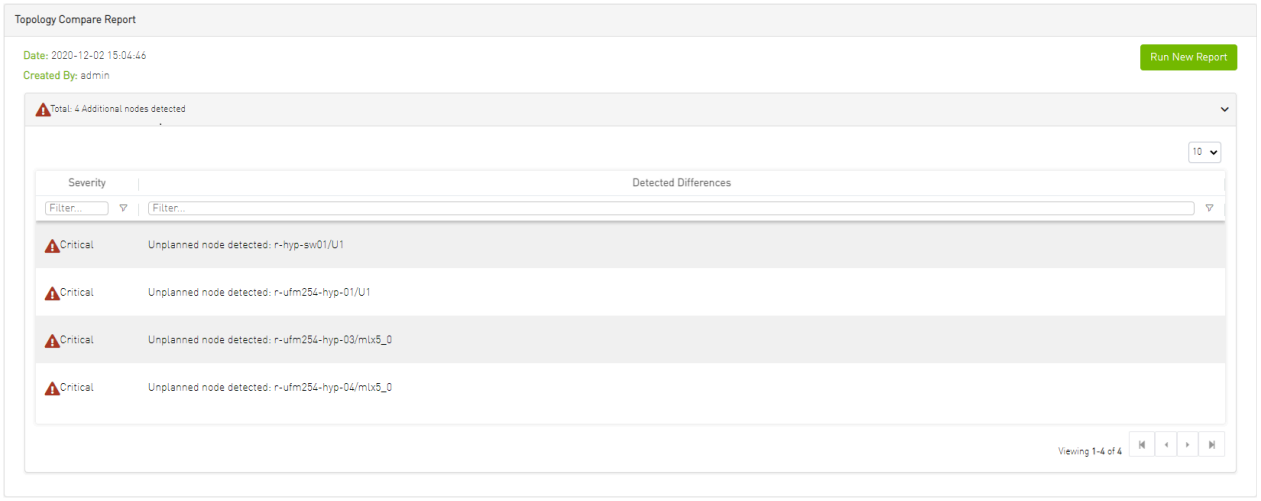
1. Click **Run Now Report** under System Health à Topology Compare.



2. Browse for the required topology setup file in the *Load Topology File* dialog box.

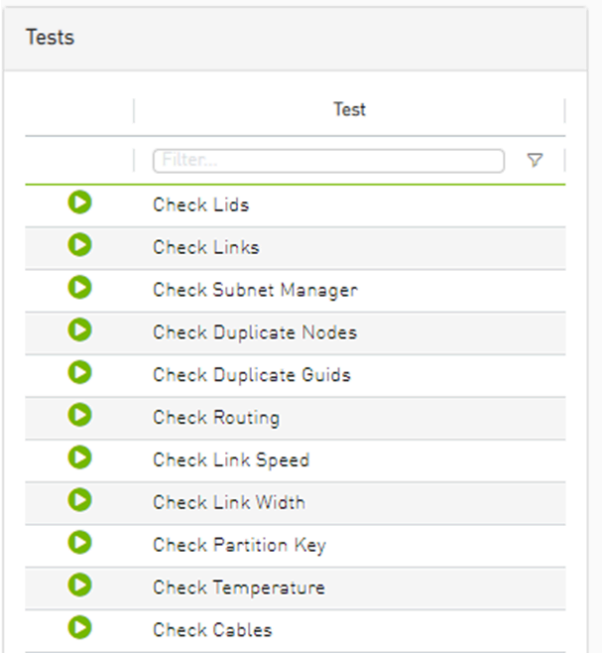
3. Click Load.

   UFM will compare topologies and display the results.



Fabric Validation Tab

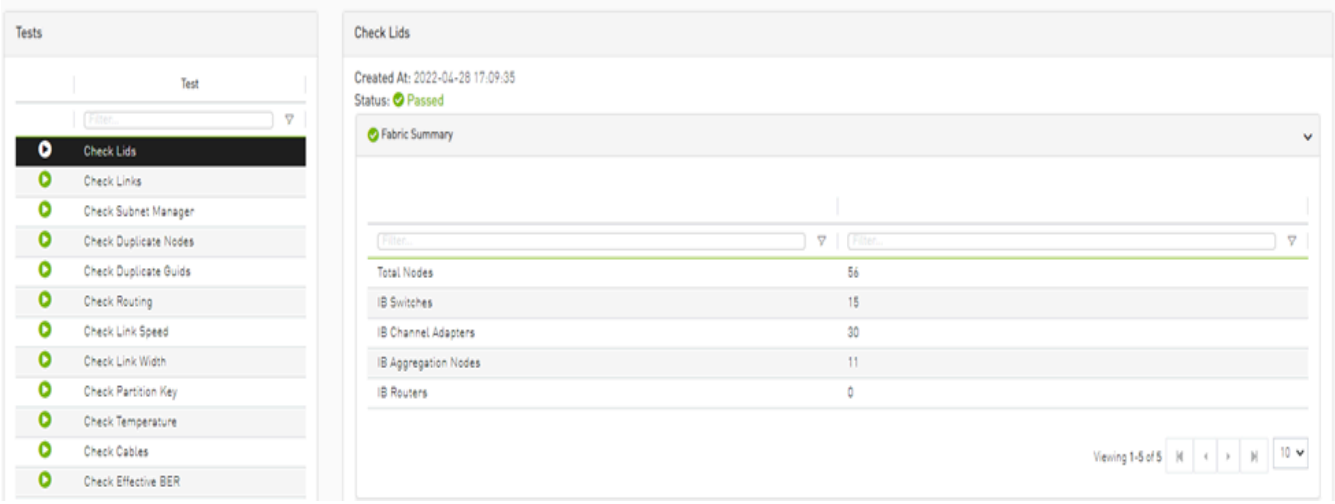
The Fabric Validation tab displays the fabric validation tests and gives the ability to run the test and receive/view the summary as a job output. Summary of the job contains all errors and warnings that were found during the test execution.



Test	Description
Check Lids	Checks for bad lids. Possible lid errors are: • zero lid • lid duplication
Check Links	Checks for connectivity issues where all ports connected are not in the same state (active)
Check Subnet Manager	Checks for errors related to subnet manager. Possible SM errors are: • Failed to get SMInfo Mad • SM Not Found • SM Not Correct (master SM with wrong priority) • Many master SMs exists
Check Duplicate Nodes	Checks for duplications in nodes description
Check Duplicate Guids	Checks for GUIDs duplications
Check Routing	Checks for failures in getting routing MADs

Test	Description
Check Link Speed	Checks for errors related to link speed. Possible link speed errors are: • Different speed between ports • Wrong configuration – 'enable' not part of the 'supported' • Unexpected speed
Check Link Width	Checks for errors related to link width. Possible link width errors are: • Different width between ports • Wrong configuration – 'enable' not part of the 'supported' • Unexpected width
Check Partition Key	Checks for errors related to PKey. Possible PKey errors are: • Failed to get Pkey Tables • Mismatching pkeys between ports
Check Temperature	Checks for failure in getting temperature sensing.
Check Cables	Checks for errors related to cables. Possible cable errors are: • This device does not support cable info capability • Failed to get cable information (provides a reason)
Check Effective BER	Checks that the Effective BER does not exceed the threshold
Dragonfly Topology Validation	Validate if the topology is Dragonfly
SHARP Fabric Validation	Checks for SHARP Configurations in the fabric
Tree Topology Validation	Checks if the fabric is a tree topology
Socket Direct Mode Reporting	Presents the inventory of fabric HCAs that are using socket direct

To run a specific test, click the play button. The job will be displayed once completed.



Note

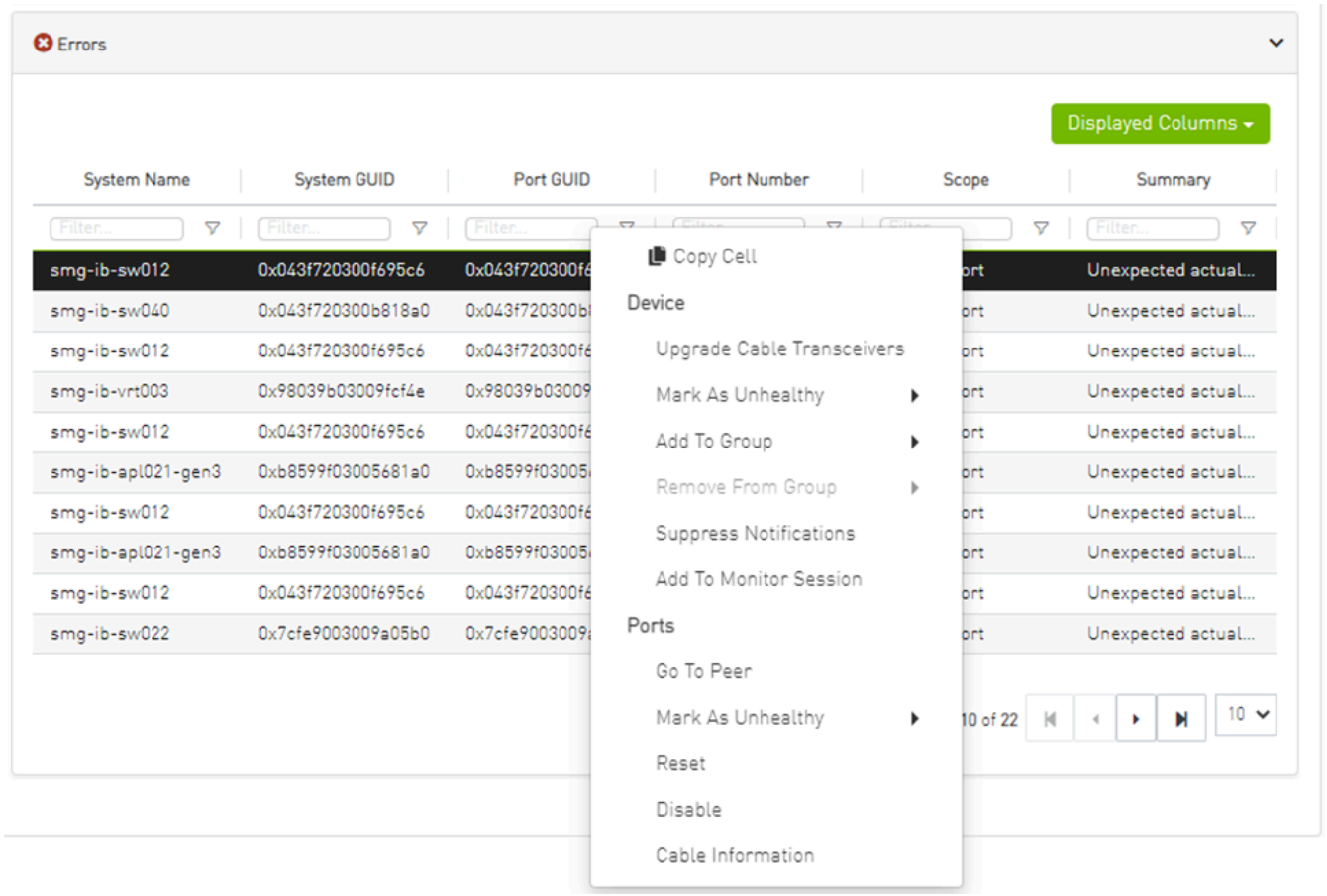
The job will also be displayed in the Jobs window.

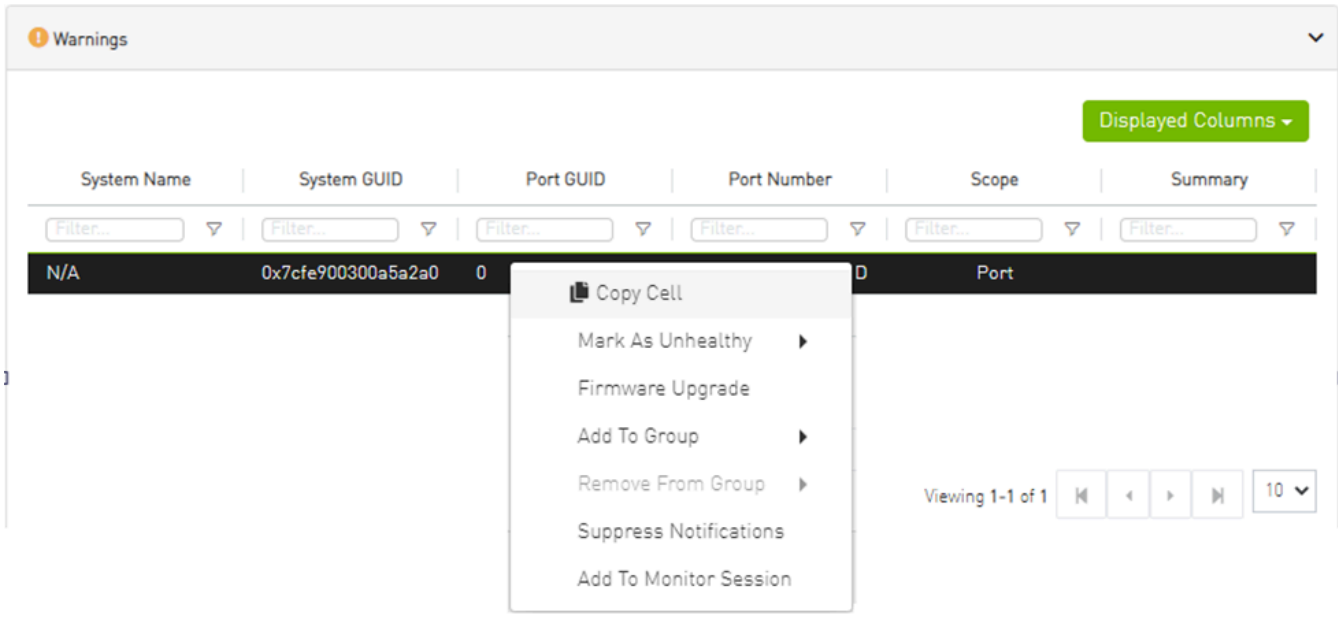
Some validation tests contain data related to devices or ports like device GUID and port GUID.

Depending on that information a context menu for each related device/port can be shown.

Note

I f the data i s related to a port the context menu will contain both port and device options .



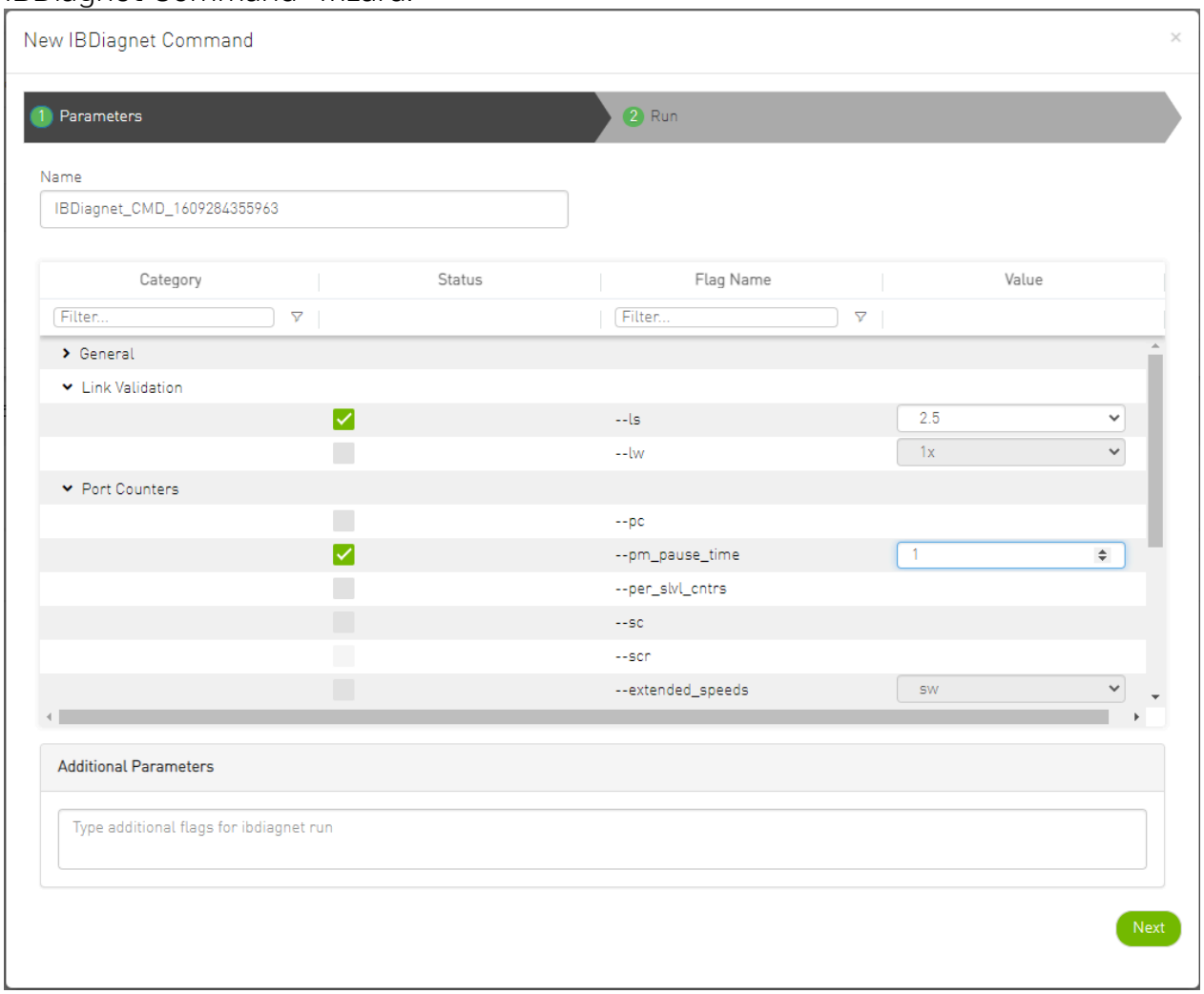


IBDiagnet Tab

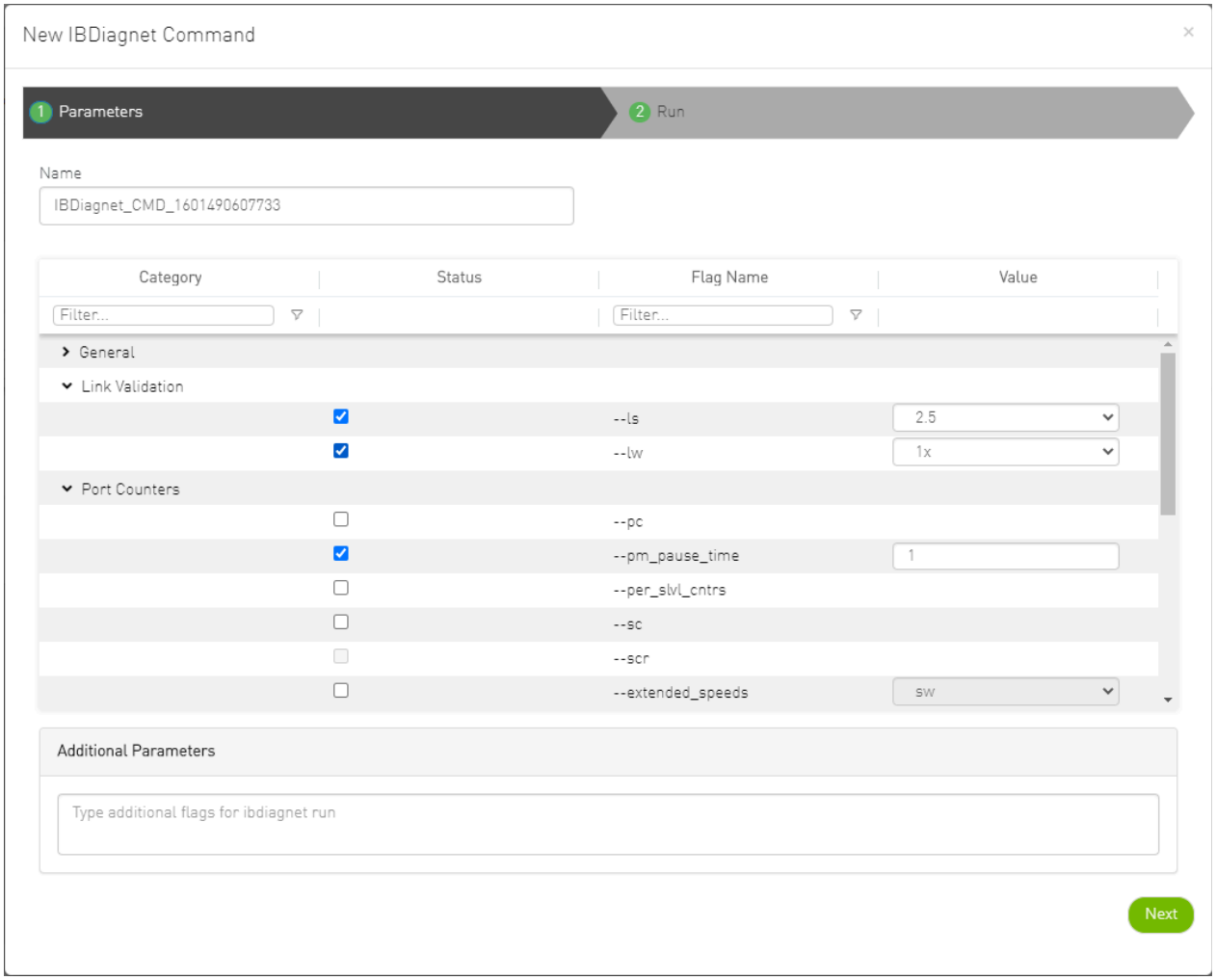
The periodic IBDiagnet tab allows users to create scheduled ibdiagnet tasks on their devices using any of the defined parameters.

Users can also configure a remote location (local/remote) to save the ibdiagnet output to.To create a new ibdiagnet command:

1. Click the New button on the top right of the IBDiagnet tab to open the “New IBDiagnet Command” wizard.



2. Select the desired ibdiagnet flags for your command by selecting the listed flags (categories are expandable), or by manually adding the desired flags into the Additional Parameters box below, and then click Next.

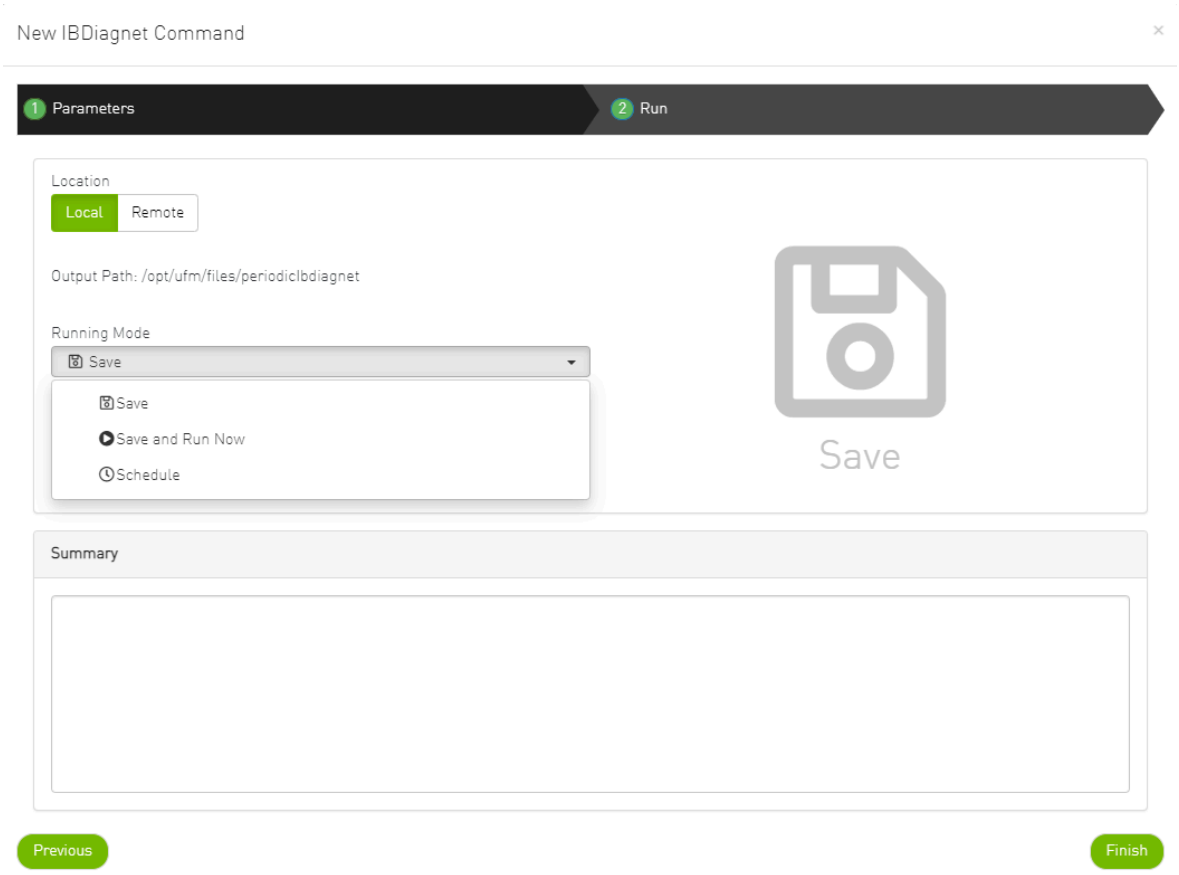


> **Note**
>
> It is possible to use the filters at the top of the Category and Flag Name columns in order to search for flags.

3. In the Run screen:

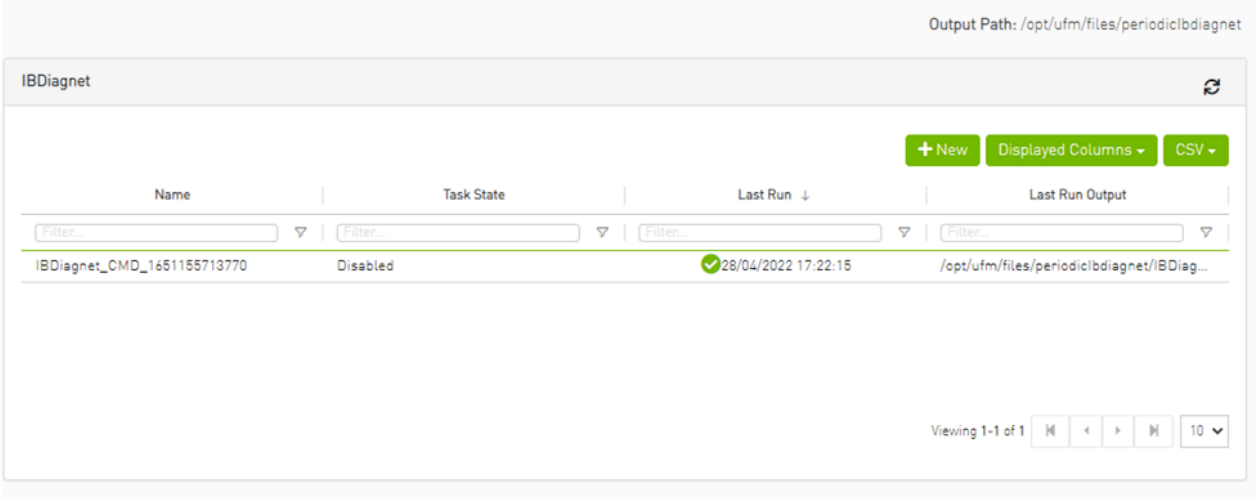
    1. Select the location of the ibdiagnet results. UFM can export ibdiagnet command run results to a local location on the UFM server, or to a configurable remote location.

    2. Select whether you would like to save this run for later (Save), run it immediately (Save and Run Now), or schedule it for a later time (Schedule) and then click Finish.



> **Note**
>
> Note that you can see the summary of your chosen flags for this run in the Summary panel.

You will then be able to see run results on the tab which will display where the output is saved on the server.



It is also optional to edit/activate/deactivate/delete a running task using right-click.

Under gv.cfg, it is possible to configure other parameters.

```
[PeriodicIbdiagnet]
# Directory location where outputs are written
periodic_ibdiagnet_dir_location=/opt/ufm/files/periodicIbdiagnet
# Minimum time between two tasks (in minutes)
minimum_task_interval=60
# Maximum number of tasks running simultaneously
max_optional_tasks=5
# Maximum number of outputs to save per task (oldest gets
deleted)
max_saved_outputs=5
# Percentage threshold for disk usage from which UFM deletes old
task results
disk_usage_threshold=80
```

Note

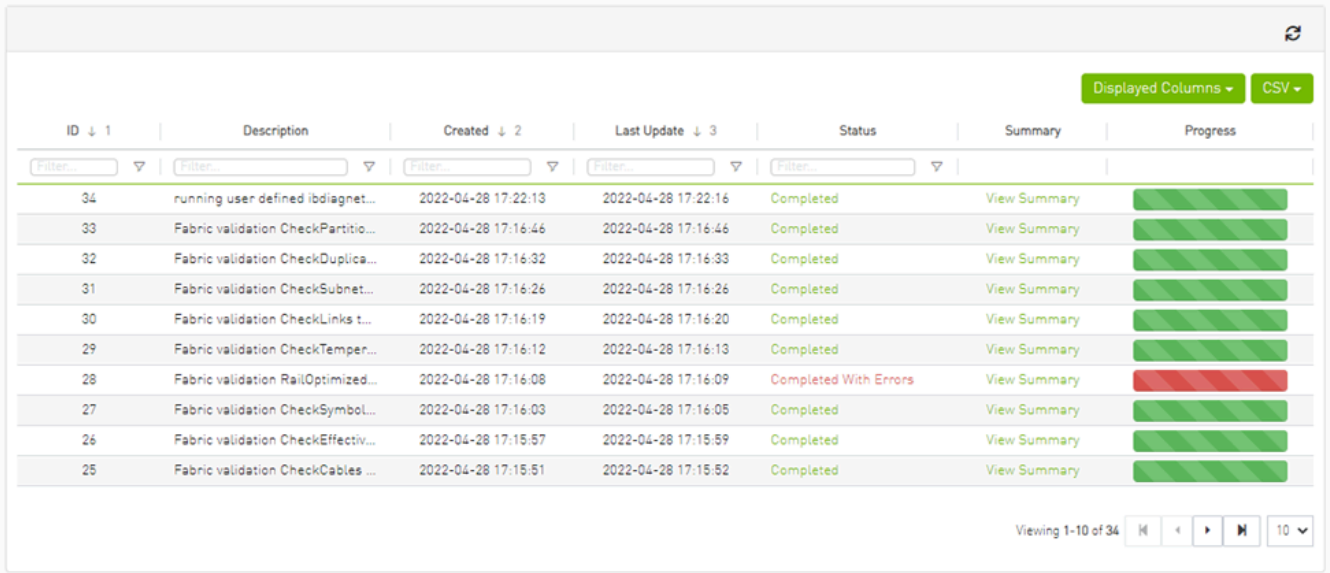
UFM restart is required for these changes to take effect.

Jobs

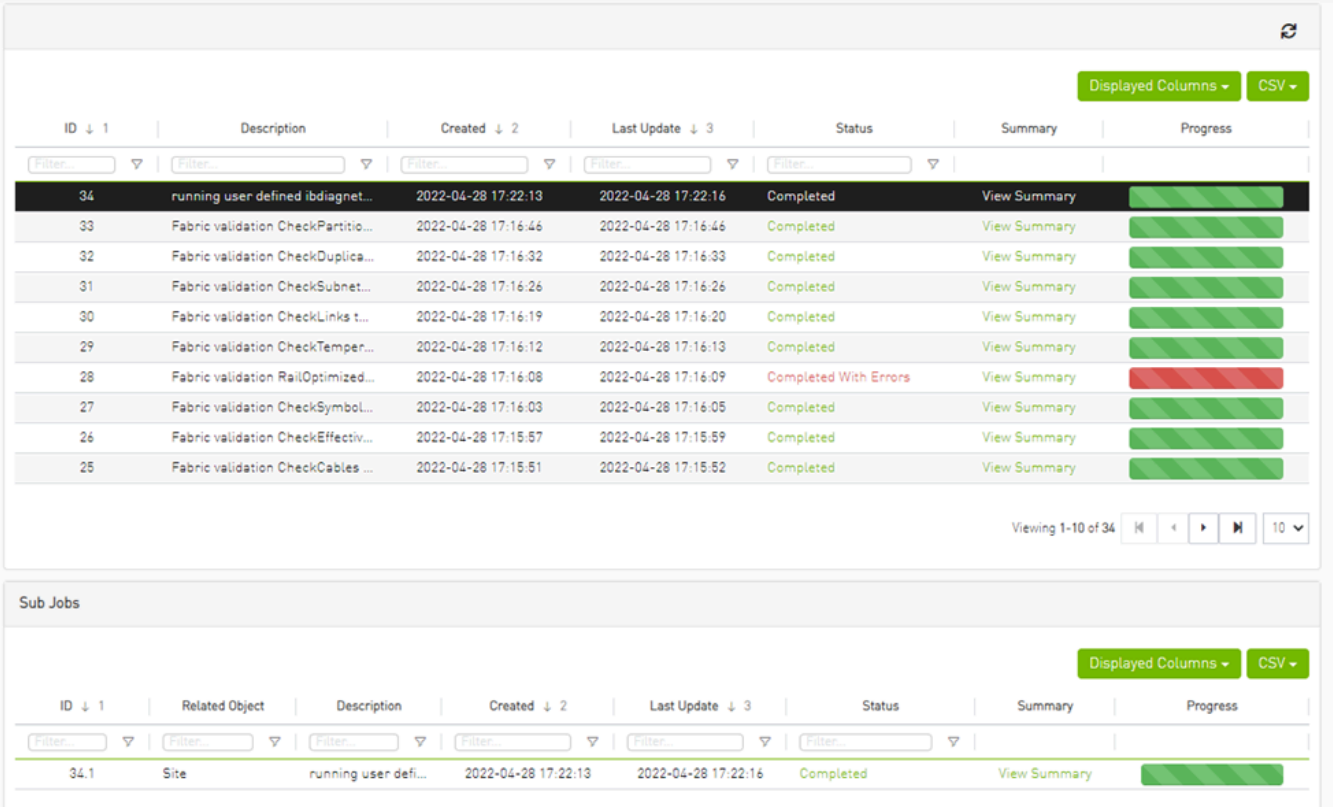
> **Note**
>
> All information provided in a tabular format in UFM web UI can be exported into a CSV file.

The Jobs window displays all of UFM running Jobs. A Job is a running task defined by the user and applied on one or more of the devices (provisioning, software upgrade, firmware upgrade, reboot, etc.).

UFM users can monitor the progress of a running job, as well as the time it was created, its last update description and its status. The status value can be “Running” (during operation) “Completed with Errors”, in case an error has occurred, and “Completed."



When selecting a job from the main Jobs table, its related sub jobs will be displayed in the Sub Jobs table below.



Settings

> **Note**
>
> All information provided in a tabular format in UFM web UI can be exported into a CSV file.

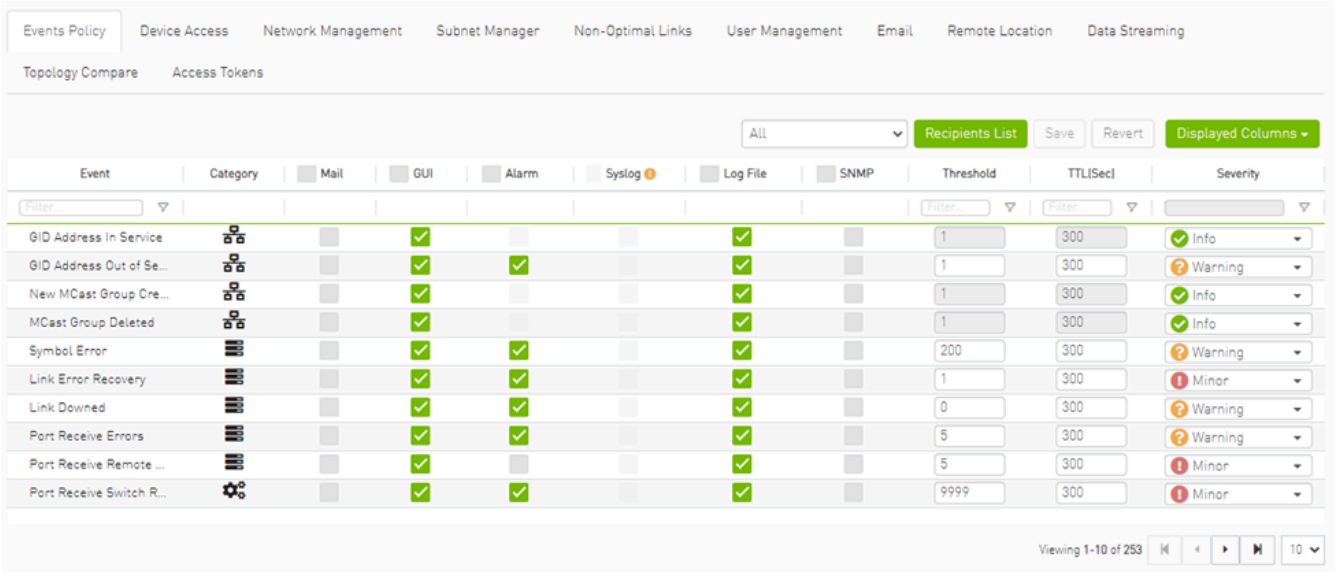
This window enables configuring the following UFM server and fabric-related settings:

- Events Policy
- Device Access
- Network Management
- Subnet Manager Tab
- Non-Optimal Links

- User Management Tab
- Email
- Remote Location
- Data Streaming
- Topology Compare
- Token-based Authentication
- Plugin Management
- Rest Roles Access Control
- User Preferences

Events Policy

The Events Policy tab allows you to define how and when events are triggered for effective troubleshooting and fabric maintenance.



Events are reported by setting the following parameters:

Option	Description/Instructions
Event	Event description.

Option	Description/Instructions
Category	Event category, such as Communication Error and Hardware represented by icons.
Mail	When selected, the corresponding events will be sent a list of recipients according to Configuring Email-on-Events.
Web UI	When selected, the corresponding events are displayed in the Events & Alarms window in the Web UI.
Alarm	Select the Alarm option to trigger an alarm for a specific event. When selected, the alarms will appear in the Events & Alarms window in the Web UI.
Syslog	When checked along with the Log file option, the selected events will be written to Syslog.
Log File	Select the Log File option if you would like the selected event to be reported in a log file.
SNMP	The UFM Server will send events to third-party clients by means of SNMP traps. Select the event SNMP check box option to enable the system to send an SNMP trap for the specific event. The SNMP trap will be sent to the port defined in Configuration file located under: /opt/ufm/conf/gv.cfg. For further information, refer to SNMP Settings.
Threshold	An event will be triggered when the traffic/error rate exceeds the defined threshold. For example: when PortXmit Discards is set to 5 and the counter value grows by 5 units or more between two sequential reads, an event is generated.
TTL (Sec)	TTL (Alarm Time to Live) sets the time during which the alarm on the event is visible on UFM Web UI. TTL is defined in seconds. CAUTION: Setting the TTL to 0 makes the alarm permanent, meaning that the alarm does not disappear from the Web UI until cleared manually.
Action	The action that will be executed in case the event which has triggered the action can be none or isolated (make the port unhealthy or isolated). This attribute can be set only for ports event policy.
Severity	Select the severity level of the event and its alarm from the drop-down list: Info, Warning, Minor, and Critical.

Note

- Category column in the Events Policy table indicates to which category the event belongs. These categories are defined in the event configuration file and cannot be modified. Categories are: Hardware, Fabric Configuration, Communication Error, Fabric Notification, Maintenance, Logical Model, Fabric Topology, Gateway, Module Status, and UFM Server.
- Event logs can still be checked even if the events.log file checkbox was not checked during Syslog configuration.
- For a certain event to be sent to Syslog, both the Syslog and the Log File checkboxes must be checked. Otherwise, the selected events will not be sent to Syslog.

See Appendix - Supported Port Counters and Events for detailed information on port counters and events.

SNMP Settings

When UFM is running, the Web UI Policy Table shows the SNMP traps. You can then modify and save an SNMP Trap flag for each event. SNMP settings are enabled only after the installation of the UFM license.

UFM sends SNMP Trap using version SNMPV2 to the default port 162.

To set the SNMP properties:

1. Open the `/opt/ufm/conf/gv.cfg` configuration file.
2. Under the [Notifications] line (see the following example):
    1. Set the (snmp_listeners) IP addresses and ports
    2. Port is optional – the default port number is 162
    3. Use a comma to separate multiple listeners

Format:

```
snmp_listeners = <IP Address 1>[:<port 1>][,<IP Address 2>[:
<port 2>]…]
```

Example:

```
[Notifications]
snmp_listeners = host1, host2:166
```

Configuring Email-on-Events

UFM enables you to configure each event to be sent by email to a list of pre-defined recipients. Every 5 minutes (configurable) UFM will collect all “Mail” selected events and send them to the list of pre-defined recipients. By default, the maximum number of events which can be sent in a single email is 100 (configurable, should be in the range of 1–1000)

The order of events in the email body can be set as desired. The available options are: order by severity or order by time (by default: order by severity)

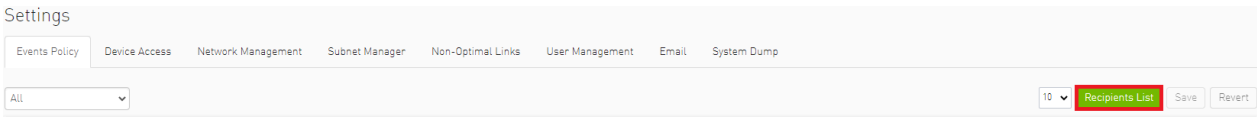
To change email-on-events setting, do the following:

1. Edit the `/opt/ufm/conf/gv.cfg` file.
2. Go to section “[Events]” and set the relevant parameters:
    - sending_interval (default=5)—Time interval for keeping events (minimum 10 seconds, maximum 24 hours)
    - sending_interval_unit (default = minute)—Optional units: minute, second, hour
    - cyclic_buffer (default=false)—If the cyclic buffer is set to true, older events will be dropped, otherwise newer events will be dropped (if reaches max count)
    - max_events (default=100)—Maximum number of events to be sent in one mail (buffer size), should be in the range of 1–1000
    - group_by_severity (default=true)—Group events in mail by severity or by time

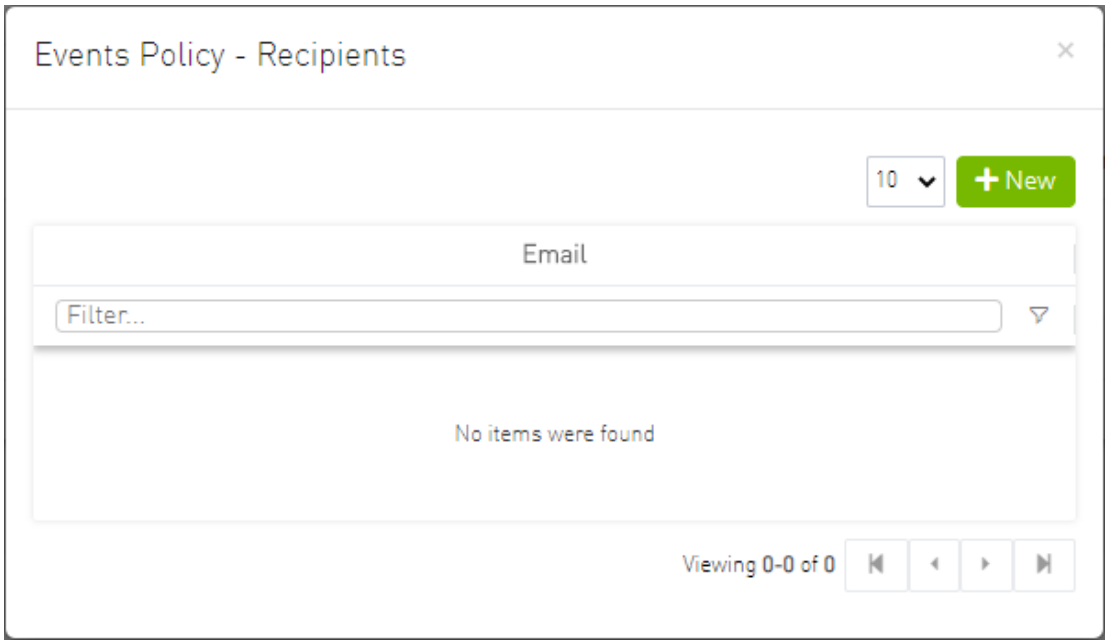
To receive the email-on-events, do the following:

> **Note**
>
> Configure SMTP settings under Settings window → Email tab – see Email Tab.

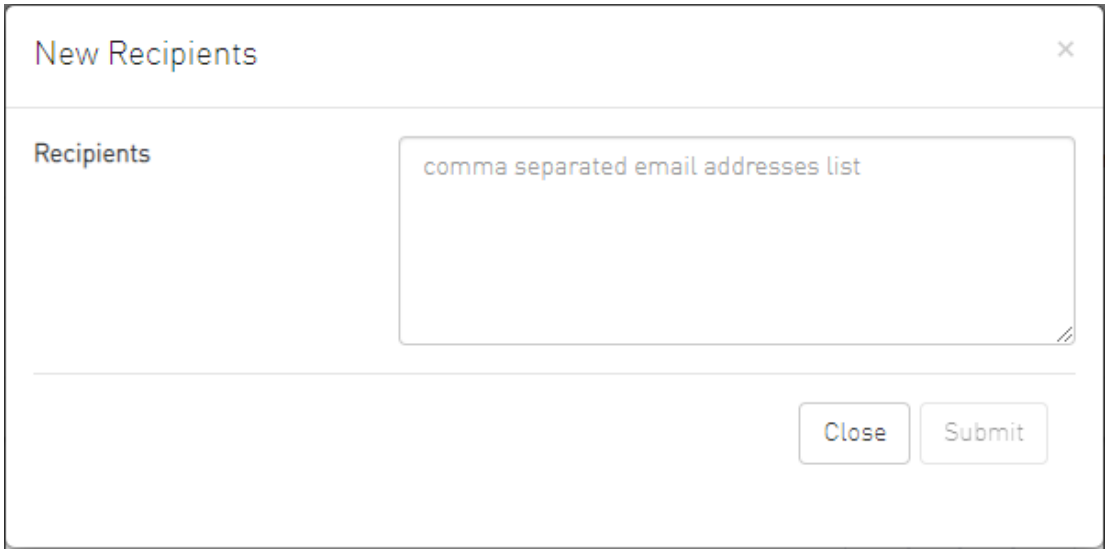
1. Configure the **Recipients List** under Settings → Events Policy.



2. Click **New**.



3. In the Recipients List window, enter valid recipient email addresses, comma-separated, and click **Submit**.



The new recipients are then added to the Events Policy Recipients list.

These recipients automatically start receiving emails on the events for which the Mail checkbox is checked in the table under Events Policy.

Device Access

You can configure default access parameters for remote administration via the following protocols:

- **Switch/Server SSH** – allows you to define the SSH parameters to open an SSH session on your device
- **IPMI** – allows you to set the IPMI parameters to open an IPMI session on your device for remote power control
- **HTTP** – allows you to define the HTTP parameters to open an HTTP session on your device

  Default credentials are applicable to all switches and servers in the fabric.

> **Note**
>
> The default SSH (CLI) switch credentials match the Grid Director series switch. To change the credentials for IS5030/IS5035 edit the [SSH_Switch] section in the gv.cfg file.

Define access parameters for the remote user as described in the following table.

Site Access Credential Parameters

Parameter	Description
User	The name of the user allowed remote access.
Password	Enter the user password.
Confirmation	Re-enter the password.
Port	Each communication protocol has a default port for connection. You can modify the port number, if required.
Timeout	Each communication protocol has a default timeout, i.e. the maximum time, in seconds, to wait for a response from the peer. You can modify the timeout, if required.

Network Management

UFM achieves maximum performance with latency-critical tasks by implementing traffic isolation, which minimizes cross-application interference by prioritizing traffic to ensure critical applications get the optimal service levels.

UFM Routing Protocols

UFM web UI supports the following routing engines:

- MINHOP – based on the minimum hops to each node where the path length is optimized (i.e., shortest path available).
- UPDN – also based on the minimum hops to each node but it is constrained to ranking rules. Select this algorithm if the subnet is not a pure Fat Tree topology and deadlock may occur due to a credit loops in the subnet.

- DNUP – similar to UPDN, but allows routing in fabrics that have some channel adapter (CA) nodes attached closer to the roots than some switch nodes.

- File-Based (FILE) – The FILE routing engine loads the LFTs from the specified file, with no reaction to real topology changes.

- Fat Tree – an algorithm that optimizes routing for congestion-free "shift" communication pattern.

  Select Fat Tree algorithm if a subnet is a symmetrical or almost symmetrical fat-tree. The Fat Tree also optimizes K-ary-N-Trees by handling non-constant K in cases where leafs (CAs) are not fully staffed, and the algorithm also handles any Constant Bisectional Bandwidth (CBB) ratio. As with the UPDN routing algorithm, Fat Tree routing is constrained to ranking rules.

    - Quasi Fat Tree – PQFT routing engine is a closed formula algorithm for two flavors of fat trees

    - Quasi Fat Tree (QFT)

    - Parallel Ports Generalized Fat Tree (PGFT)

      PGFT topology may use parallel links between switches at adjacent levels, while QFT uses parallel links between adjacent switches in different sub-trees. The main motivation for that is the need for a topology that is not just optimized for a single large job but also for smaller concurrent jobs.

- Dimension Order Routing (DOR) – based on the Min Hop algorithm, but avoids port equalization, except for redundant links between the same two switches. The DOR algorithm provides deadlock-free routes for hypercubes, when the fabric is cabled as a hypercube and for meshes when cabled as a mesh.

- Torus-2QoS – designed for large-scale 2D/3D torus fabrics. In addition, you can configure Torus-2QoS routing to be *traffic aware*, and thus optimized for neighbor-based traffic.

- Routing Engine Chain (Chain) – an algorithm that allows configuring different routing engines on different parts of the IB fabric.

- Adaptive Routing (AR) – enables the switch to select the output port based on the port's load. This option is not available via UFM Web UI.

    - AR_UPDN

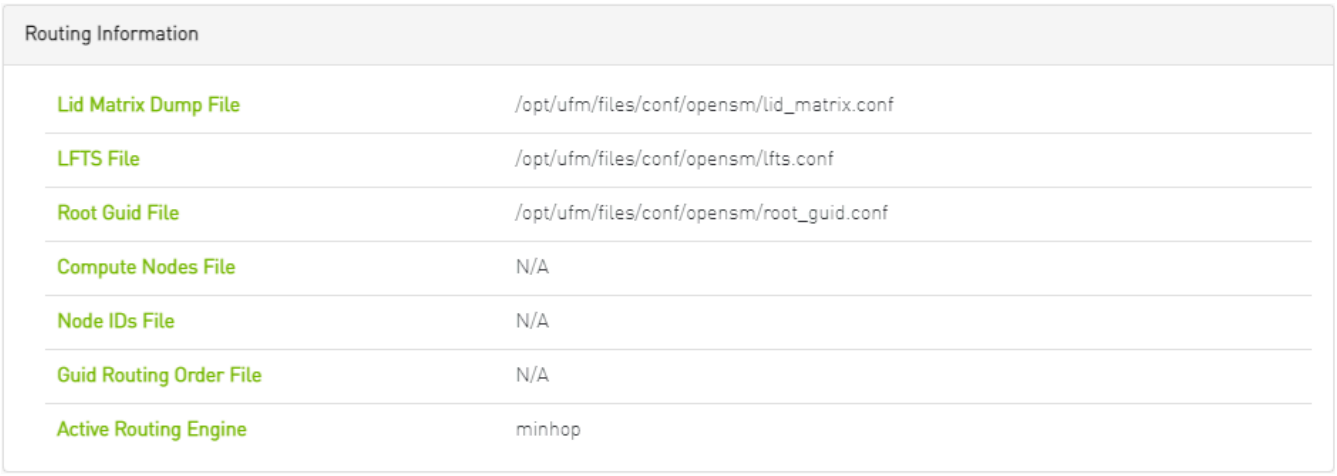
    - AR_FTREE

    - AR_TORUS
    - AR_DOR
- Dragonfly+ (DFP, DPF2)

Configuring Routing Protocol

Network Management tab enables setting the preferred routing protocol supported by the UFM software, as well as routing priority.

To set the desired routing protocol, move one routing protocol or more from the **Available** list to the **Selected** list, and click "Save" in the upper right corner.

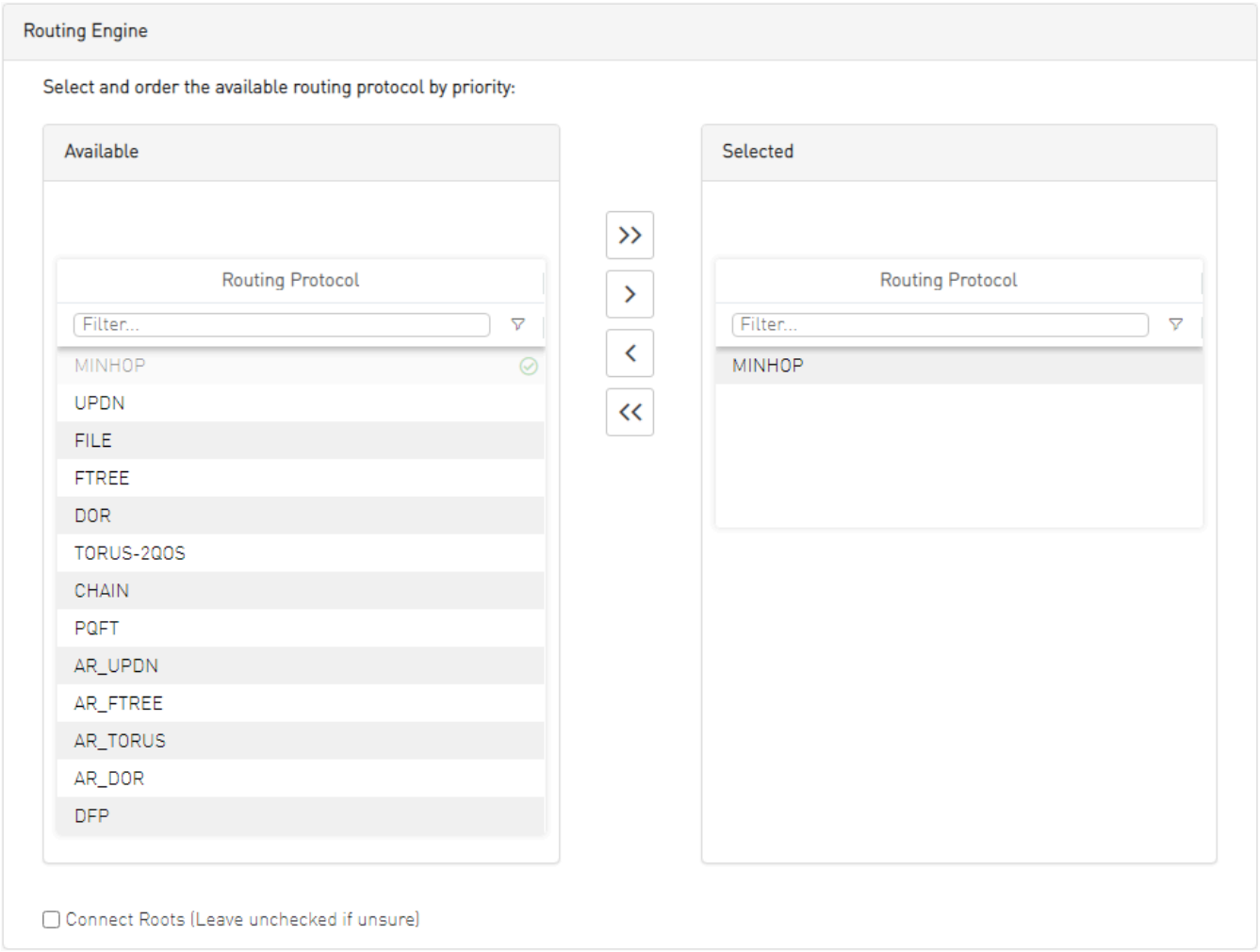


The protocol at the top of the list has the highest priority and will be chosen as the **Active Routing Engine**. If the settings for this protocol are not successful, UFM takes the next available protocol.

Routing Information is listed on the top of the screen:

Field/Box	Description
LID Matrix Dump File	File holding the LID matrix dump configuration
LFTS File	File holding the LFT routing configuration
Root GUID File	File holding the root node GUIDS (for fat-tree or Up/Down)
Compute Nodes File	File holding GUIDs of compute nodes for fat-tree routing algorithm

Field/Box	Description
GUID Routing Order File	File holding the routing order GUIDs (for MinHop and Up/Down)
Node IDs File	File holding the node IDs
Active Routing Engine	The current active routing algorithm used by the managing OpenSM



Subnet Manager Tab

UFM is a management platform using a user-space application for InfiniBand fabric management. This application is developed within the context of an open-source environment. This application serves as an InfiniBand Subnet Manager and a Subnet Administration tool.

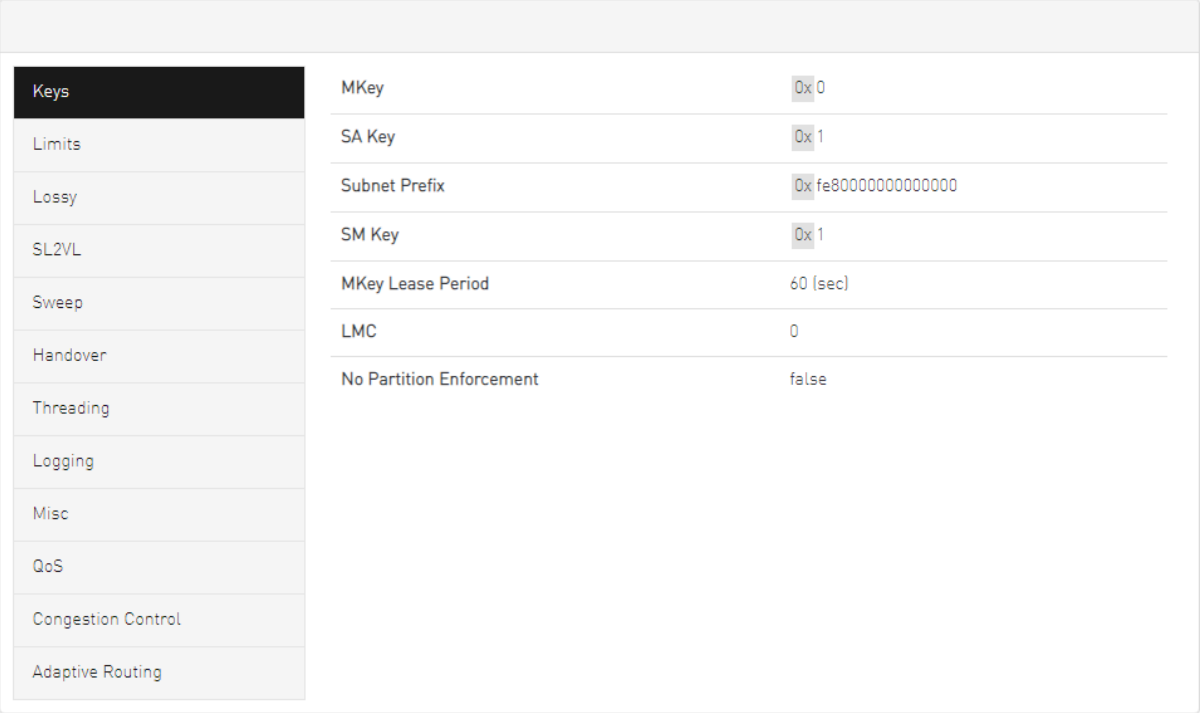
The UFM Subnet Manager (SM) is a centralized entity running on the server that discovers and configures all the InfiniBand fabric devices to enable traffic flow throughout the fabric.

To view and configure SM parameters in the ***Subnet Manager*** tab, select the relevant tab according to the required configuration.

For more information, please refer to Appendix – Enhanced Quality of Service.

SM Keys Configuration

The SM Keys tab enables you to view the Subnet Manager Keys. You cannot change the configuration in this tab.

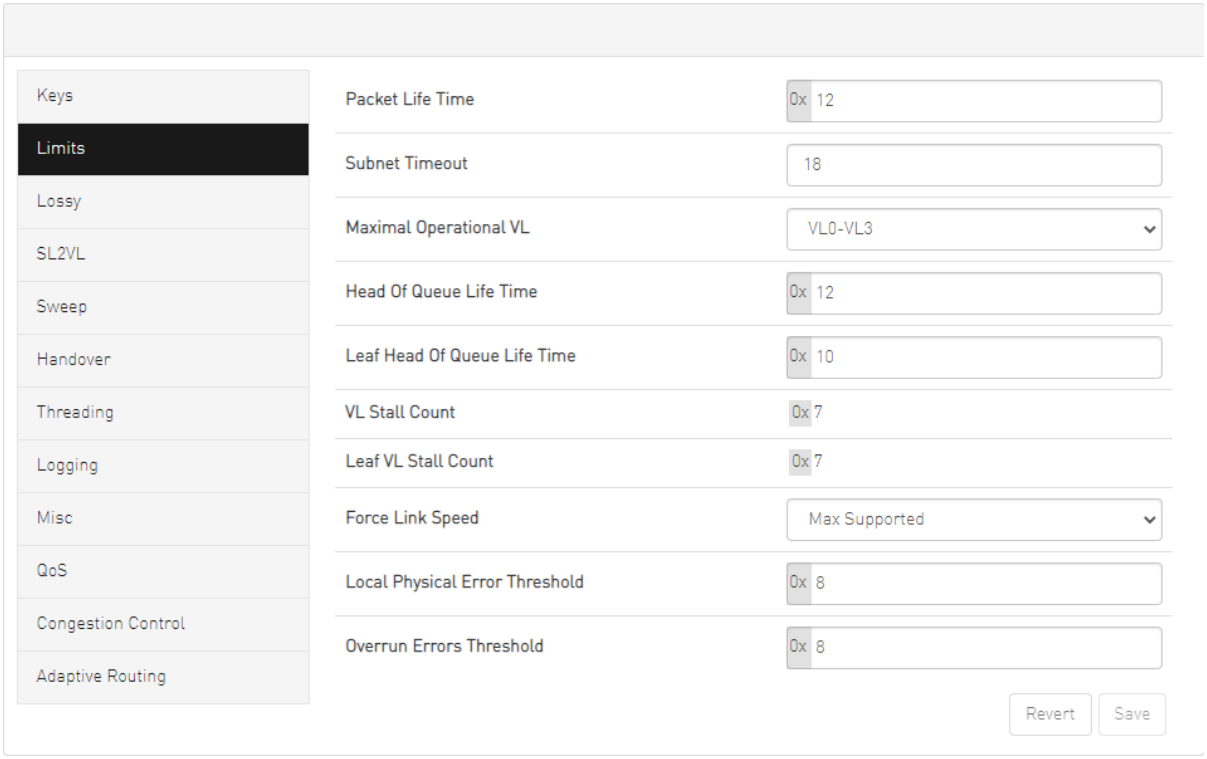


Field	Description	Default
MKey	A field that allows you to view or edit the M_Key value sent to all ports to qualify all the set (PortInfo). Authentication is performed by the management entity at the destination port and is achieved by comparing the key contained in the SMP with the key (the M_Key Management key) residing at the destination port.	0x0000000000000000
SA Key	Shows the SM_Key value to qualify the receive SA queries as 'trusted'.	0x0000000000000001

Field	Description	Default
Subnet Prefix	An identifier of the subnet. The subnet prefix is used as the most significant 64 bit of the GID of each InfiniBand node in the subnet.	0xfe80000000000000
SM Key	Read-only field that displays the Key of the Subnet Manager (SM).	0x0000000000000001
MKey Lease Period	A field that allows you to view or edit the lease period used for the M_Key on this subnet in [sec].	0
LMC	Defines the LID Mask Control value for the SM. Possible values are 0 to 7. LID Mask Control (LMC) allows you to assign more than one LID per port. NOTE: Changes to the LMC parameter require a UFM restart.	0
No Partition Enforcement	Disables partition enforcement by switches.	Disabled

SM Limits Configuration

The SM Limits tab enables you to view and set the Subnet Manager Limits.



To configure SM Limits, set the fields as described in the table below, and click “Save."

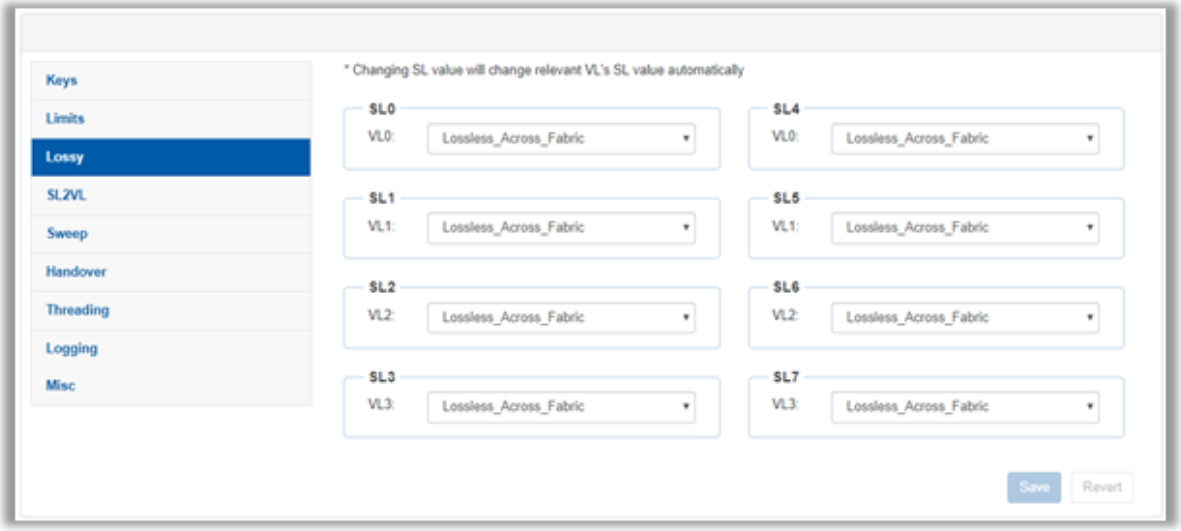
Field	Description	Default
Packet Life Time	A field that allows you to view and/or edit the code of maximum lifetime a packet in a switch. The actual time is 4.096 usec * 2^<packet_life_time>. The value 0x14 disables this mechanism	0x12
Subnet Timeout	A field that allows you to view and/or edit the subnet_timeout code that will be set for all the ports. The actual timeout is 4.096usec * 2^<subnet_timeout>	18
Maximal Operational VL	A field that allows you to view and/or edit the limit of the maximal operational VLs: • 0: NO_CHANGE • 1: VL0 1 • 2: VL0_VL1 • 3: VL0_VL3 • 4: VL0_VL7 • 5: VL0_VL14	3
Head of Queue Life Time	A field that allows you to view and/or edit the code of maximal time a packet can wait at the head of transmission queue. The actual time is 4.096usec * 2^<head of queue lifetime> The value 0x14 disables this mechanism.	0x12
Leaf Head of Queue Life Time	A field that allows you to view and/or edit the maximum time a packet can wait at the head of queue on a switch port connected to a CA or gateway port.	0x10
VL Stall Count	A field that allows you to view the number of sequential packets dropped that cause the port to enter the VLStalled state. The result of setting this value to zero is undefined.	0x07
Leaf VL Stall Count	This field allows you to view the number of sequential packets dropped that cause the port to enter the VLStalled state. This value is for switch ports driving a CA or gateway port. The result of setting the parameter to zero is undefined.	0x07

Field	Description	Default
Force Link Speed	A parameter that allows you to modify the PortInfo:LinkSpeedEnabled field on switch ports. If 0, do not modify. • Values are: • 1: 2.5 Gbps • 3: 2.5 or 5.0 Gbps • 5: 2.5 or 10.0 Gbps • 7: 2.5 or 5.0 or 10.0 Gbps • 2,4,6,8-14 Reserved • 15: set to PortInfo:LinkSpeedSupported	15 By default, UFM sets the enabled link speed equal to the supported link speed.
Local Physical Error Threshold	A field that allows you to view and/or edit the threshold of local phy errors for sending Trap 129.	0x08
Overrun Errors Threshold	A field that allows you to view and/or edit the threshold of credit overrun errors for sending Trap 130.	0x08

SM Lossy Manager Configuration

> **Note**
>
> This tab is available to users with an advanced license only.

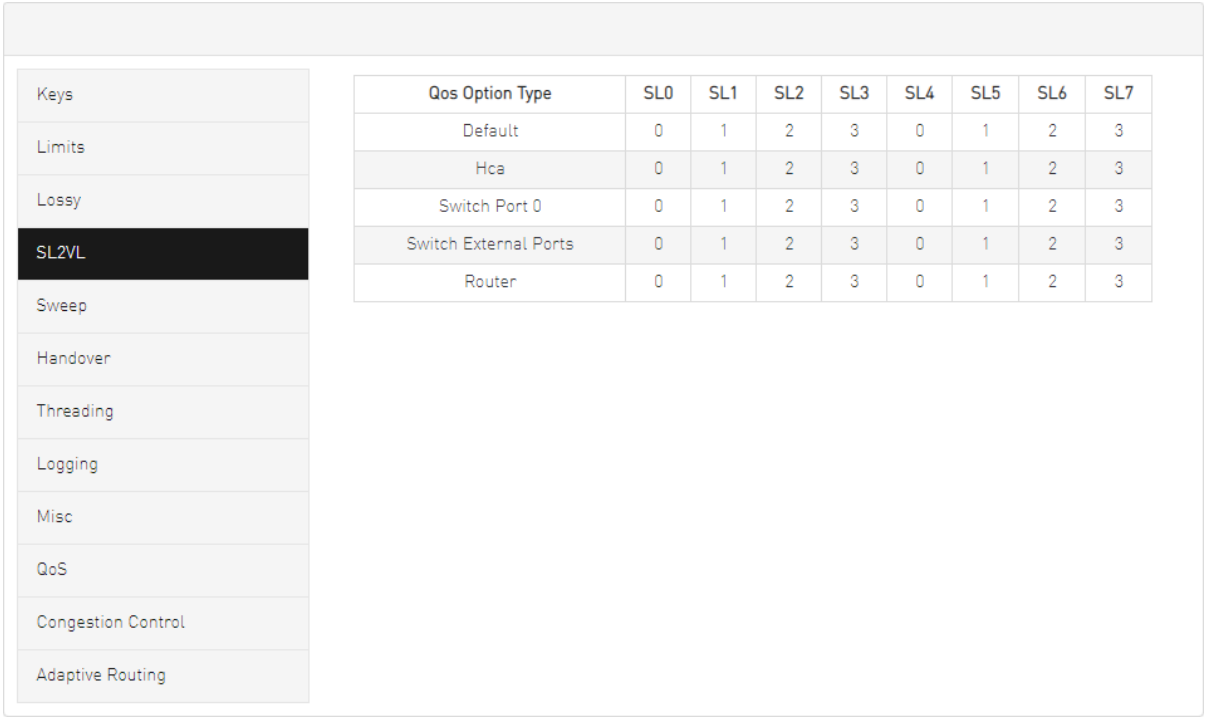
The SM Lossy tab enables you to view and set the Lossy Configuration Manager options after Lossy Configuration has been enabled.



SM SL2VL Mapping Configuration

The SM SL2VL tab enables you to view the SL (service level) to VL (virtual lane) mappings and the configured Lossy Management. You cannot change the configuration in this tab.

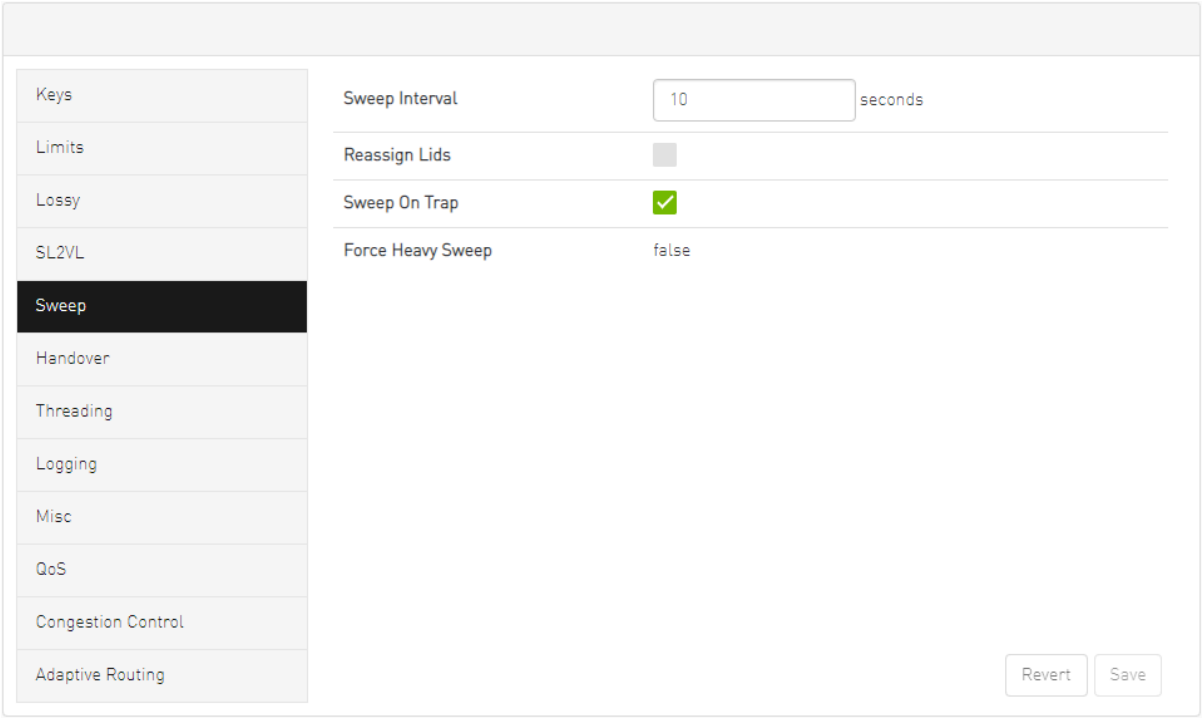
However, you can change it in the previous SM Lossy Manager Configuration (Advanced License only) tab.



Qos Option Type	SL0	SL1	SL2	SL3	SL4	SL5	SL6	SL7
Default	0	1	2	3	0	1	2	3
Hca	0	1	2	3	0	1	2	3
Switch Port 0	0	1	2	3	0	1	2	3
Switch External Ports	0	1	2	3	0	1	2	3
Router	0	1	2	3	0	1	2	3

SM Sweep Configuration

The Sweep tab enables you to view and/or set the Subnet Manager Sweep Configuration parameters.

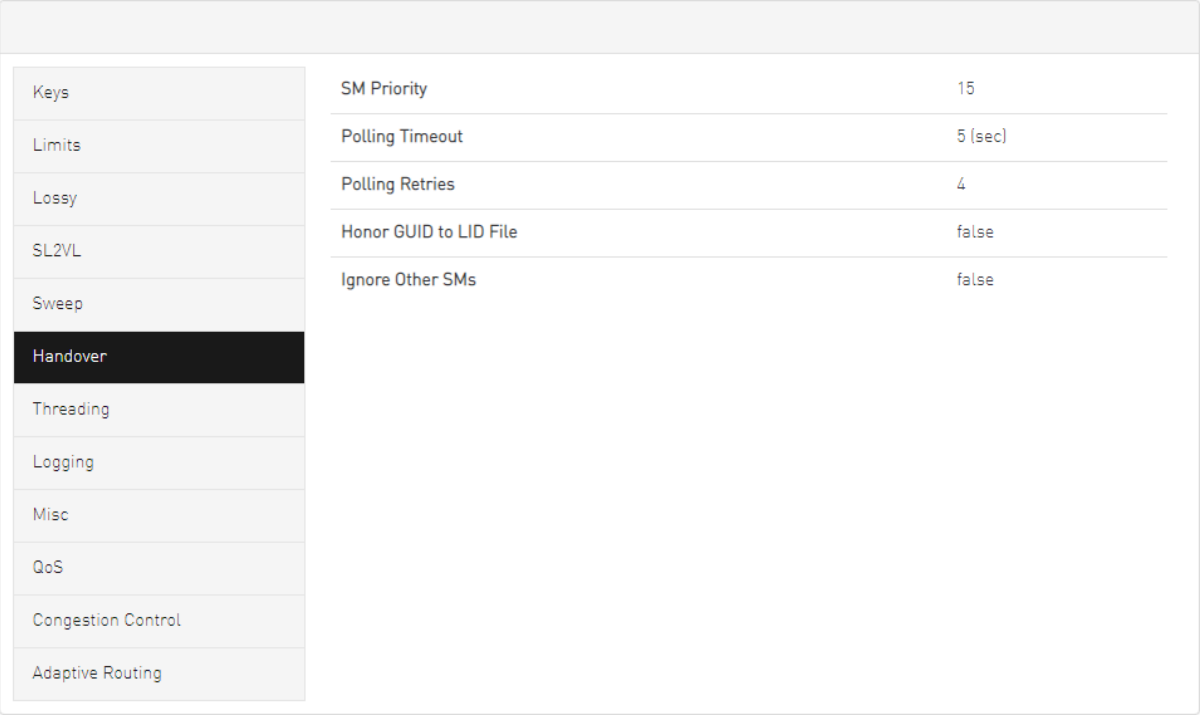


To configure SM Sweep, set the fields as described in the table below and click “Save."

Field/Box	Description	Default
Sweep Interval	A field that allows you to view and/or edit the number of seconds between light sweeps (0 disables it).	10
Reassign LIDs	If enabled, causes all LIDs to be reassigned.	Disabled
Sweep on Trap	If enabled, traps 128 and 144 will cause a heavy sweep.	Enabled
Force Heavy Sweep	If enabled, forces every sweep to be a heavy sweep.	Disabled

SM Handover Configuration

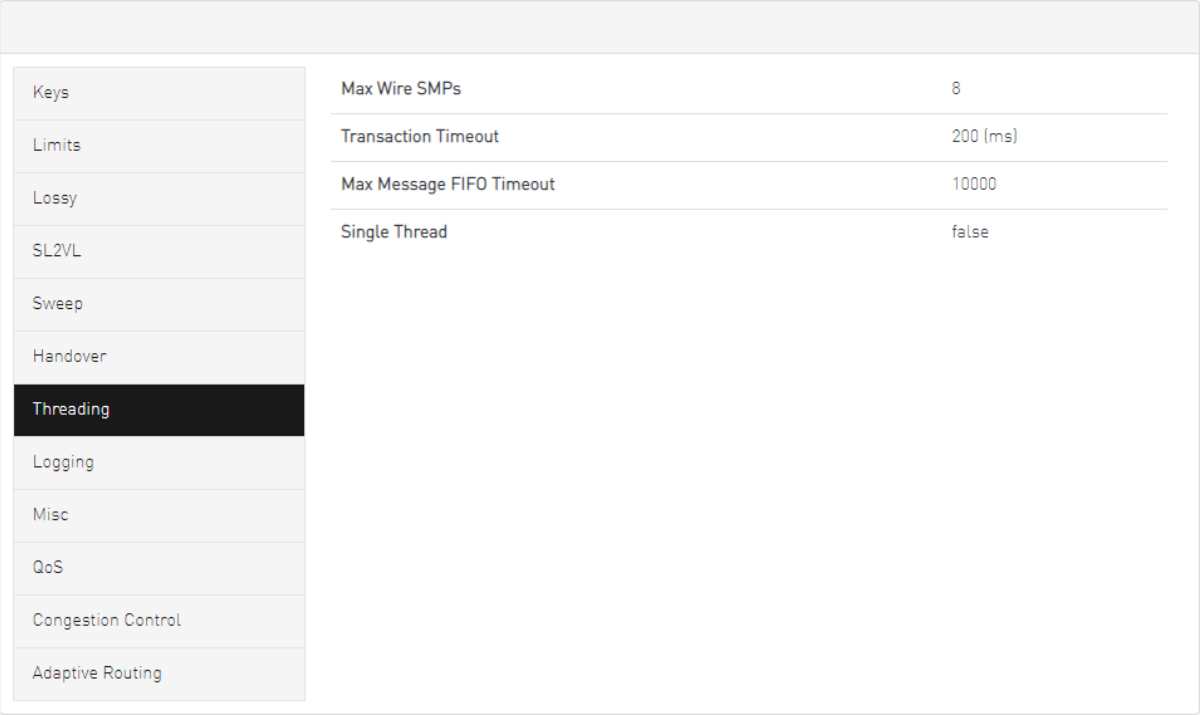
The SM Handover tab enables you to view the Subnet Manager Handover Configuration parameters. You cannot change the configuration in this tab.



Field/Box	Description	Default
SM Priority	A field that shows the SM priority used for determining the master. Range is 0 (lowest priority) to 15 (highest). Note: Currently, these settings may not be changed.	15
Polling Timeout	A field that shows the timeout in [sec] between two polls of active master SM.	Range=10000
Polling Retries	Number of failing polls of remote SM that declares it "not operational."	4
Honor GUID to LID File	If enabled, honor the guid2lid file when coming out of standby state, if the file exists and is valid.	Disabled
Ignore other SMs	If enabled, other SMs on the subnet are ignored.	Disabled

SM Threading Configuration

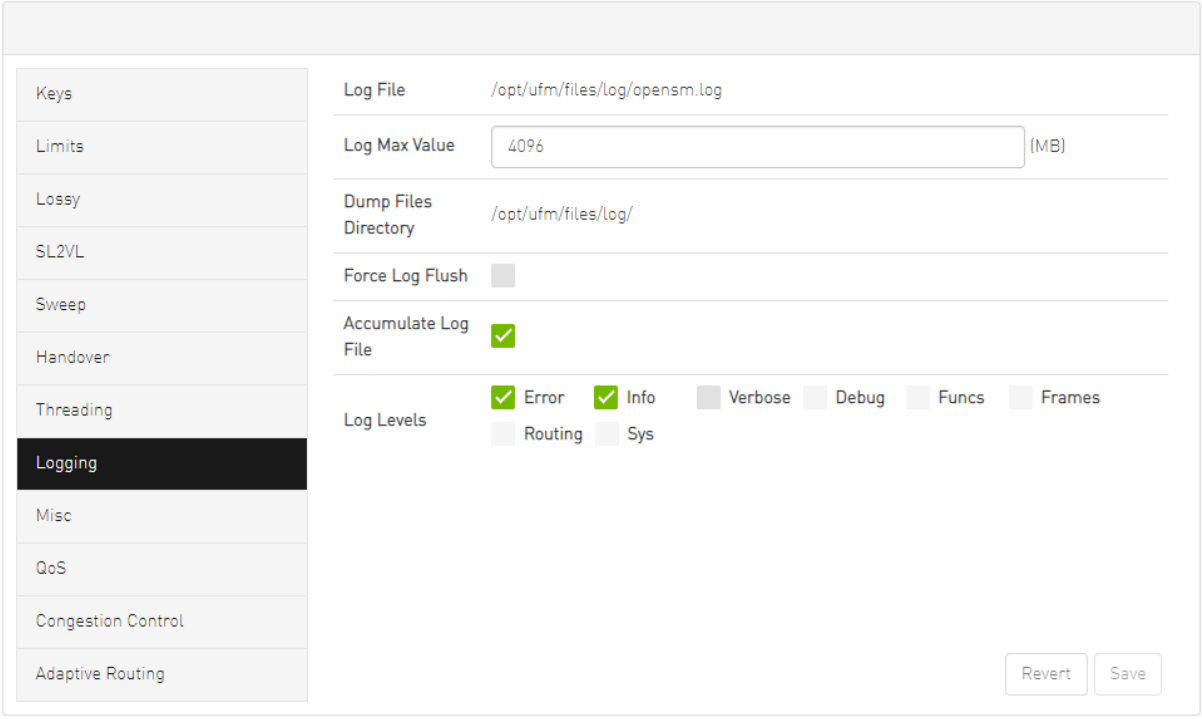
The SM Threading tab enables you to view the Subnet Manager Timing and Threading Configuration parameters. You cannot change the configuration in this tab.



Field/Box	Description	Default
Max Wire SMPs	A field that shows the maximum number of SMPs sent in parallel.	4
Transaction Timeout	A field that shows the maximum time in [msec] allowed for a transaction to complete.	200
Max Message FIFO Timeout	A field that shows the maximum time in [msec] a message can stay in the incoming message queue.	10000
Single Thread	When enabled, a single thread is used for handling SA queries.	Disabled

SM Logging Configuration

The SM Logging tab enables you to view and/or set the **Subnet Manager Logging Configuration** parameters.

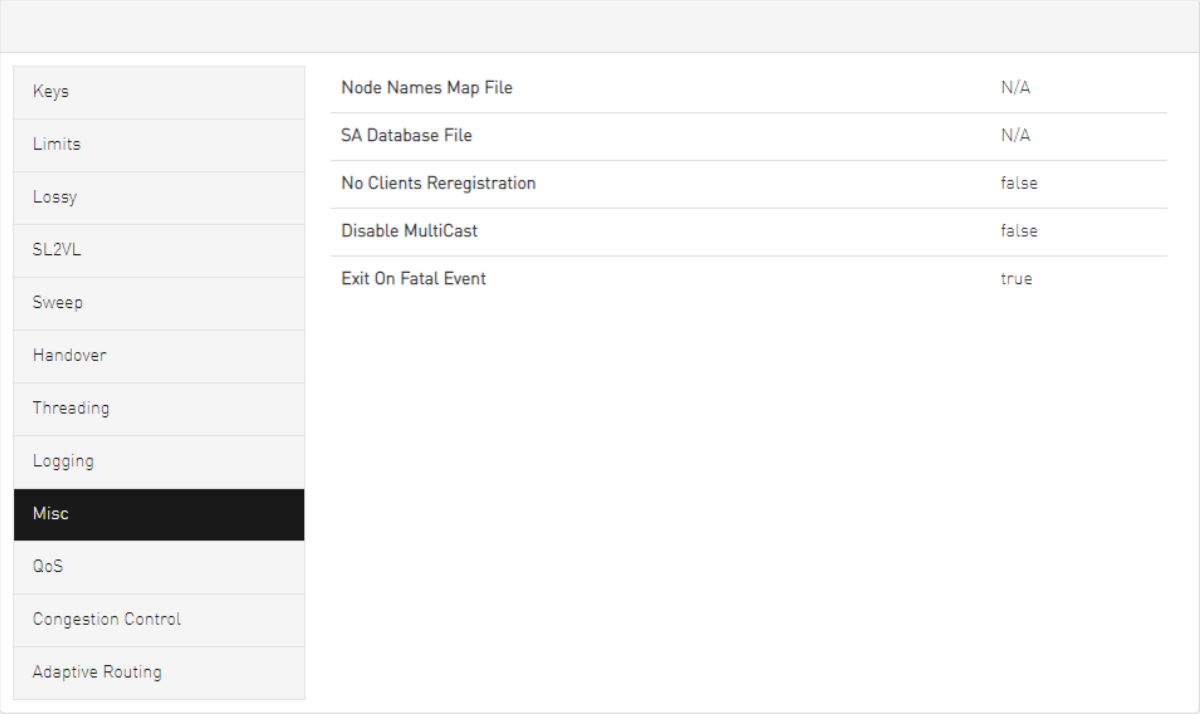


To configure SM Logging, set the fields as described in the table below and click “Save."

Field/Box	Description	Default
Log File	Path of the Log file to be used.	`cond/opt/ufm/files/log/opensm.log`
Log Max Size	A field that allows you to view and/or edit the size limit of the log file in MB. If overrun, the log is restarted.	4096
Dump Files Directory	The directory that holds the SM dump file.	`/opt/ufm/files/log`
Force Log Flush	Force flush to the log file for each log message.	Disabled
Accumulate Log File	If enabled, the log accumulates over multiple SM sessions.	Enabled
Log Levels	Available log levels: Error, Info, Verbose, Debug, Funcs, Frames, Routing, and Sys.	Error and Info

SM Miscellaneous Settings

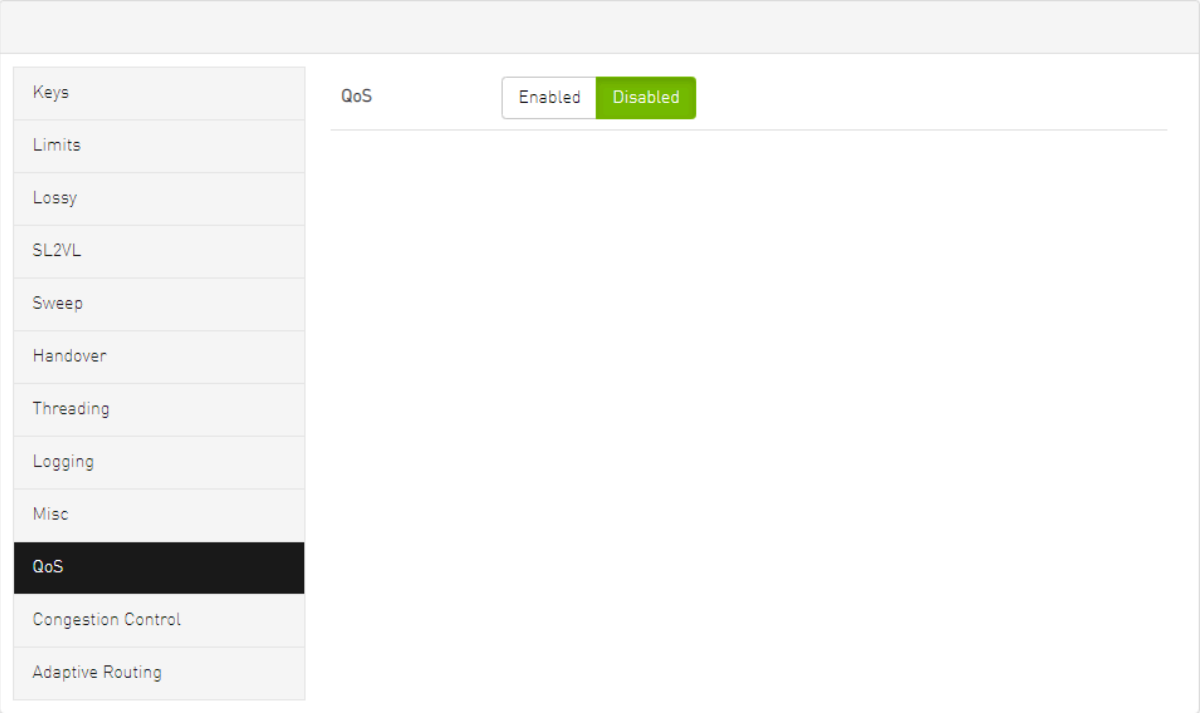
The Misc tab enables you to view additional **Subnet Manager Configuration** parameters. You cannot change the configuration in this tab.



Field/Box	Description	Default
Node Names Map File	A field that allows you to view and/or set the node name map for mapping nodes to more descriptive node descriptions.	None
SA Database File	SA database file name	None
No Clients Reregistration	If enabled, disables client re-registration.	Disabled
Disable Multicast	If enabled, the SM disables multicast support and no multicast routing is performed.	Disabled
Exit on Fatal Event	If enabled, the SM exits on fatal initialization issues.	Enabled

SM QoS Configuration

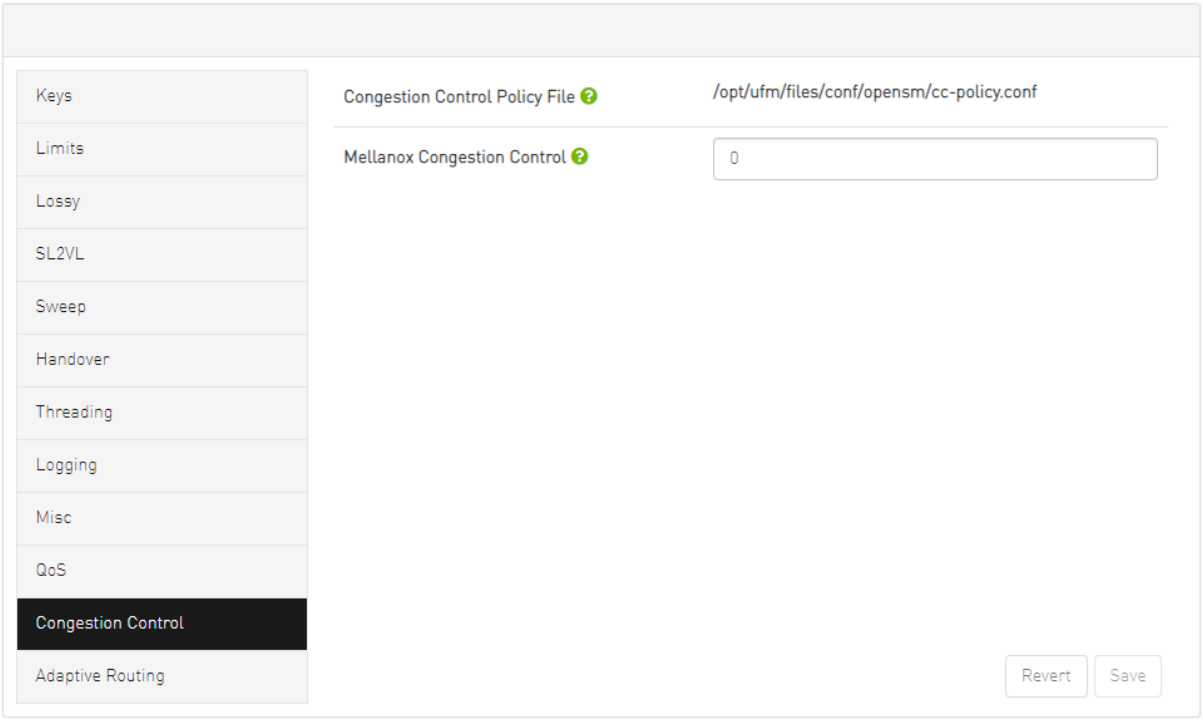
The QoS tab allows you to enable or disable QoS functionality. QoS is disabled by default.



SM Congestion Control Configuration

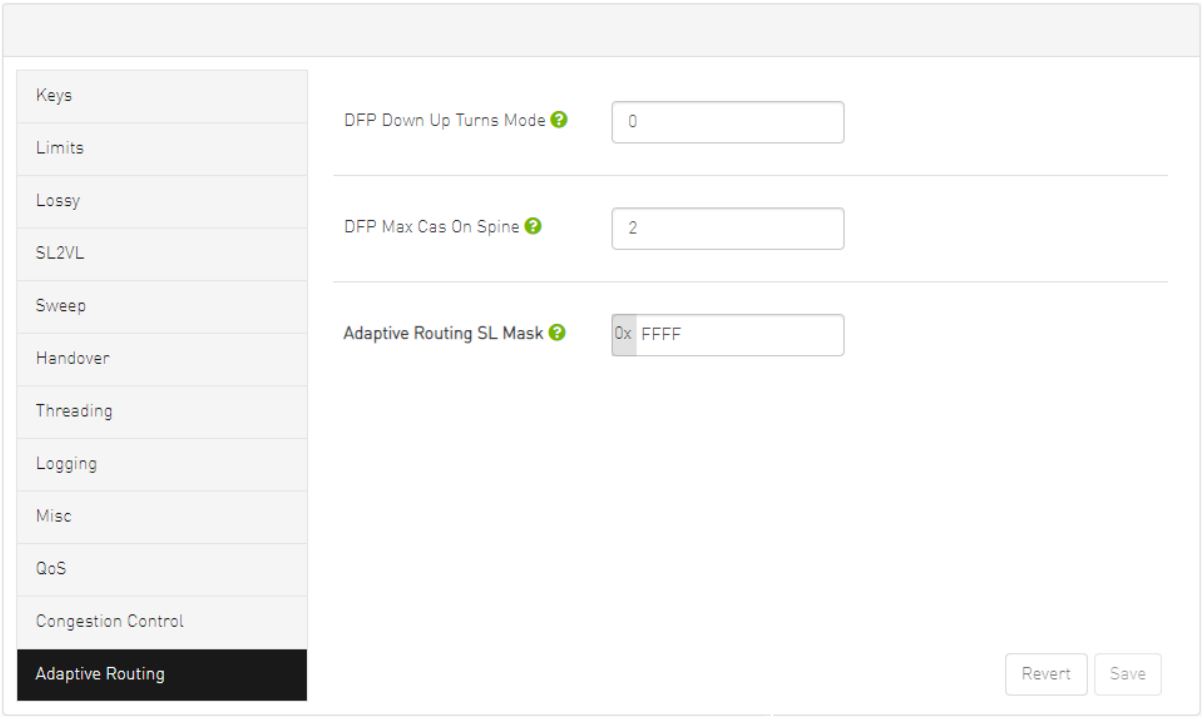
The Congestion Control tab allows you to enable, disable, or ignore congestion control.

- 0 – Ignore (default)
- 1 – Enable
- 2 – Disable



SM Adaptive Routing Configuration

The Adaptive Routing tab allows you to configure adaptive routing parameters.



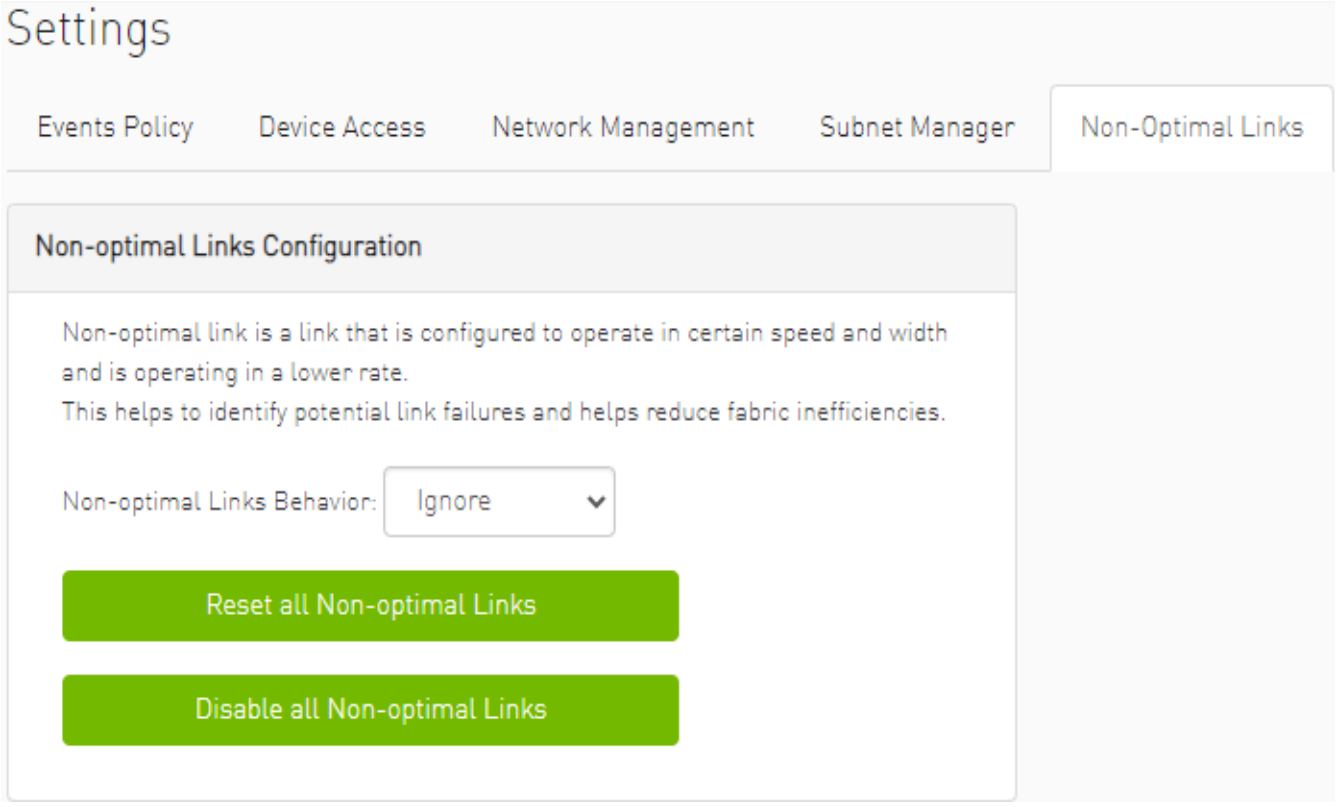
Non-Optimal Links

A non-optimal link is a link between two ports that is configured to operate at a certain speed and width and is operating at a lower rate. The Non-optimal links feature helps you identify potential link failures and reduce fabric inefficiencies.

Non-optimal links can be any of the following:

- NDR links that operate in HDR, EDR, FDR, QDR, DDR or SDR mode
- HDR links that operate in EDR, FDR, QDR, DDR or SDR mode
- EDR links that operate in FDR, QDR, DDR or SDR mode
- FDR links that operate in QDR, DDR or SDR mode
- QDR links that operate in DDR or SDR mode
- 4X links that operate in 1X mode

The Non-Optimal Links window allows you to set the preferred action for non-optimal links.



To set the non-optimal links policy:

From the drop-down menu, select the action for Non-optimal Links behavior.

The drop-down menu defines the default behavior. Options are: **Ignore** (default), **Disable**, and **Reset**.

Option	Description
Ignore	Ignore the non-optimal links
Reset	Reset all non-optimal links ports
Disable	Disable all non-optimal links ports

Reset all Non-Optimal Links allows users to reset all current non-optimal links ports on-demand.

Disable all Non-Optimal Links allows users to disable all current non-optimal links ports on-demand.

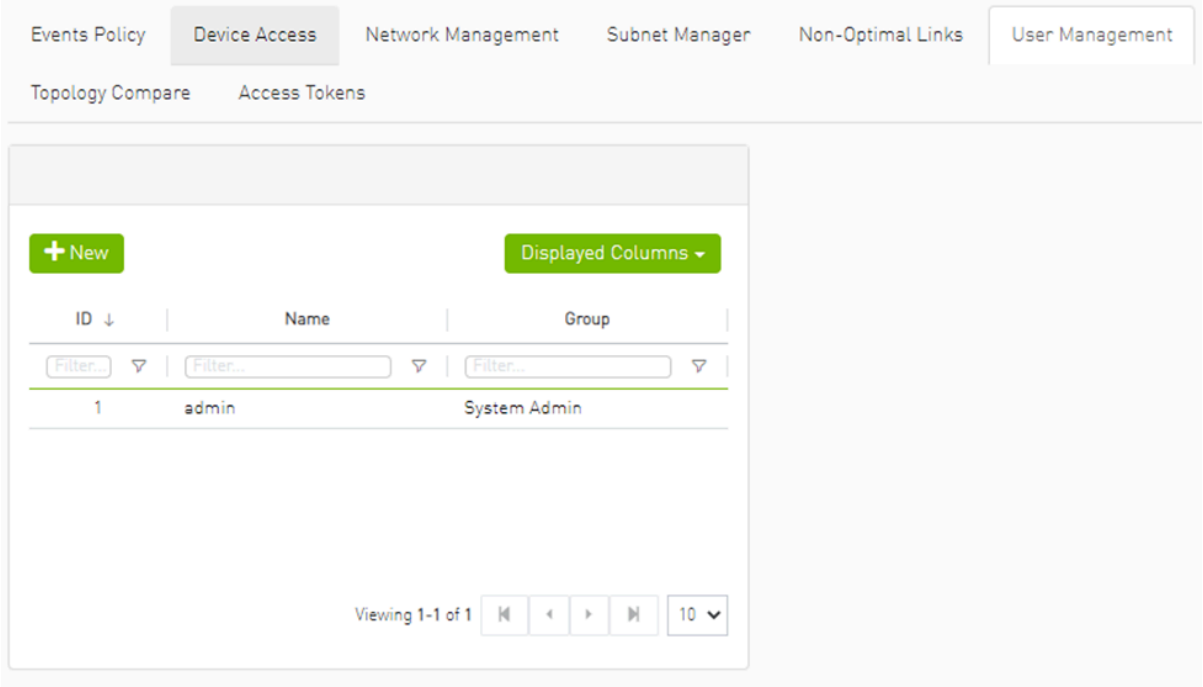
User Management Tab

UFM User Authentication is based on standard Apache User Authentication. Each Web Service client application must authenticate against the UFM Server to gain access to the system. UFM implements any kind of third-party authentication supported by the Apache Web Server.

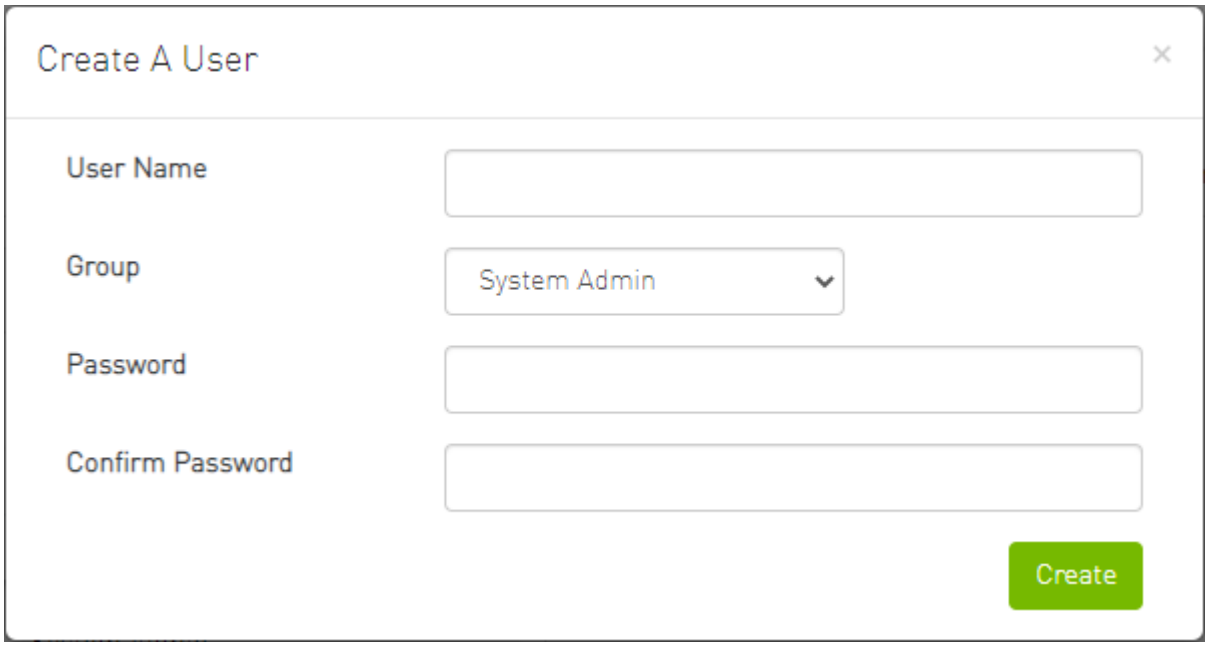
The default user (admin) has System Administration rights. A user with system Administration rights can manage other users' accounts, including creation, deletion, and modification of accounts. The system's default user is the **admin** user.

To add a new user account, do the following:

1. Click the "New" button.



2. Fill in the required fields in the dialog box.



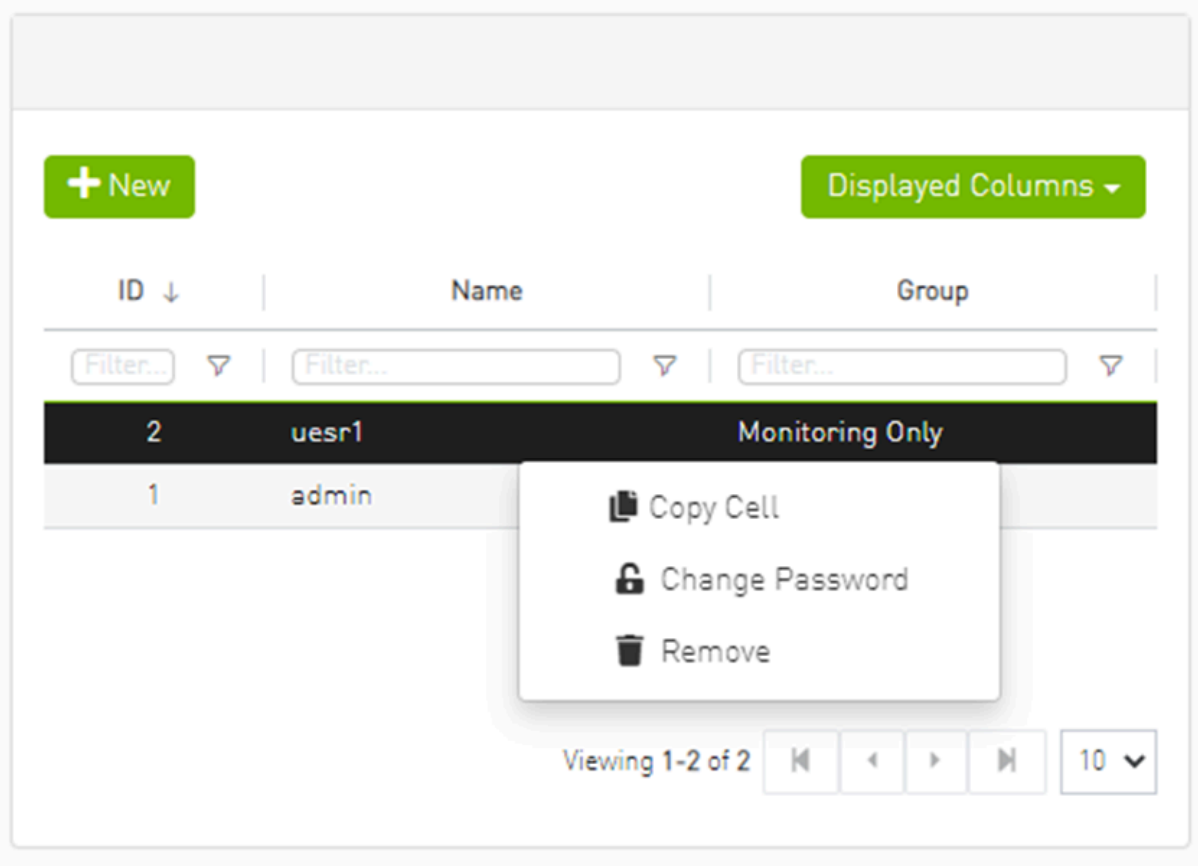
Each user can be assigned to one of the following Group (role) options:

- **System Admin** – users can perform all operations including managing other users accounts.
- **Fabric Admin** – users can perform fabric administrator actions such as update SM configuration, update global credentials, manage reports, managing unhealthy ports, and manage PKeys, etc.

- **Fabric Operator** – users can perform fabric operator actions such as device management actions (enable/disable port, add/remove devices to/from groups, reboot device, upgrade software, etc.)

- **Monitoring Only** – users can perform monitoring actions such as view the fabric configuration, open monitoring sessions, define monitoring templates, and export monitoring data to CSV files, etc.

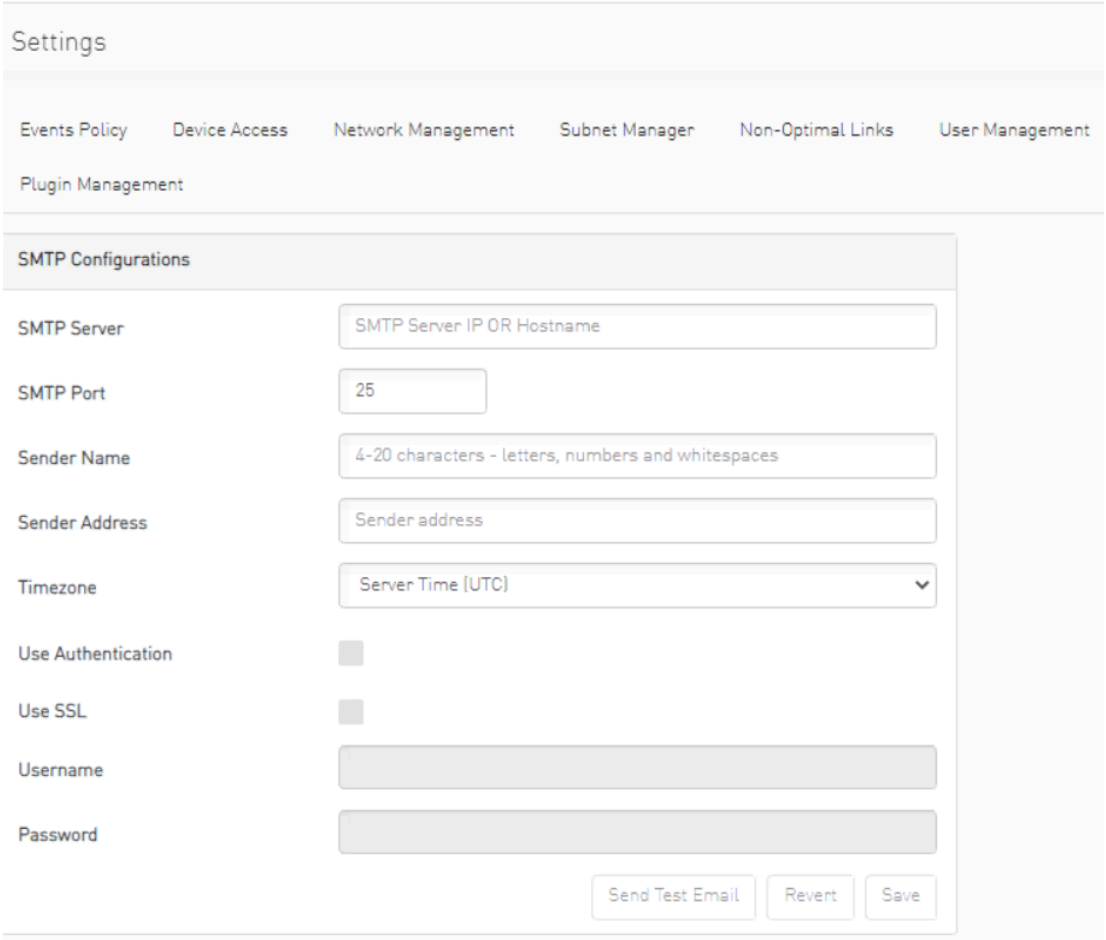
To edit existing users accounts, right-click the account from the list of user accounts and perform the desired action (Change Password/Remove).



Email

SMTP configuration is required to set both the Daily Reports Tab and the Email-on-Events features.

1. In the SMTP Configuration dialogue window, enter the following information:

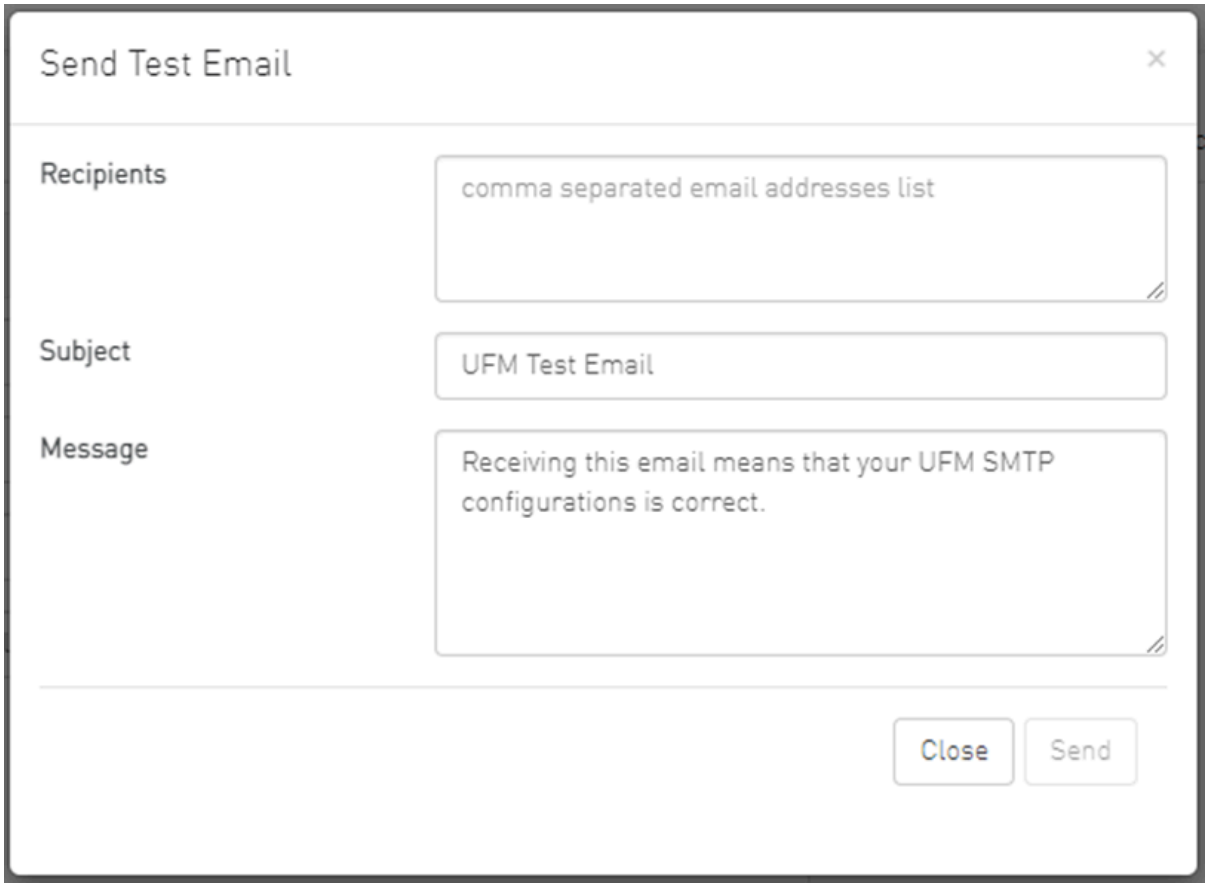


Attribute	Description
SMTP Server	The IP or host name of the SMTP server. **Examples**: ◦ If mail service is installed, localhost is a valid value for this field, but usually it cannot send mails outside the local domain. ◦ smtp.gmail.com
SMTP Port	Default value – 25
Sender Name	The name that will be displayed in the email header
Sender Address	A valid email address that will be displayed in the email header
Time Zone	The default time zone for receiving sent emails is the server time zone. Users have the option to specify a different preferable time zone
Use Authentication	By default, this field is unchecked. If checked, you must supply a username and password in the respective fields

Attribute	Description
Use SSL	Default value is false – not using SSL
Username	SMTP account username
Password	SMTP account password

2. Click “Save." All configuration of the SMTP server will be saved in the UFM Database.

   Click “Send Test Email” to test the configuration and the following model will appear:



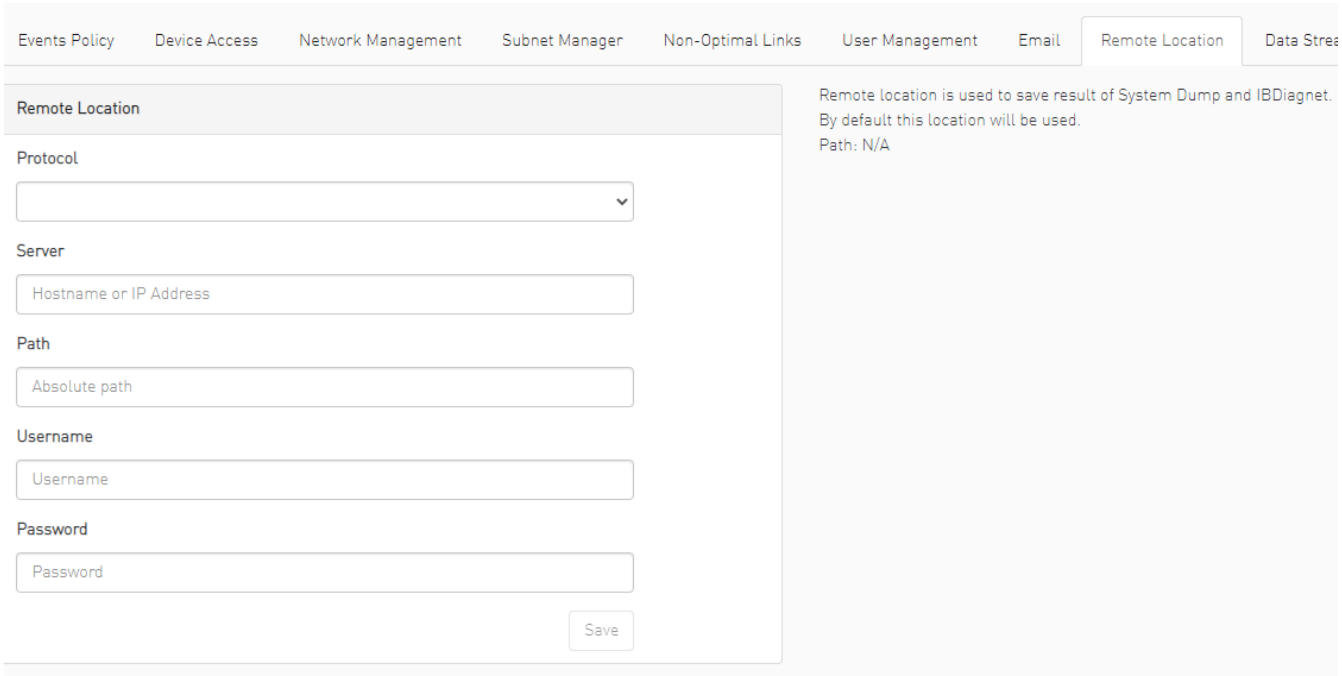
Attribute	Description
Recipients	User can choose email from event policy and daily report recipients or enter any email
Subject	Email subject
Message	Email message

The System Health window enables running and viewing reports and logs for monitoring and analyzing UFM server and fabric health through the following tabs:

UFM Health, UFM Logs, UFM Snapshot, Fabric Health, Daily Reports and Topology Compare.

Remote Location

Remote location tab is used to set a predefined remote location for the results of System Dump action on switches and hosts and for IBDiagnet executions.

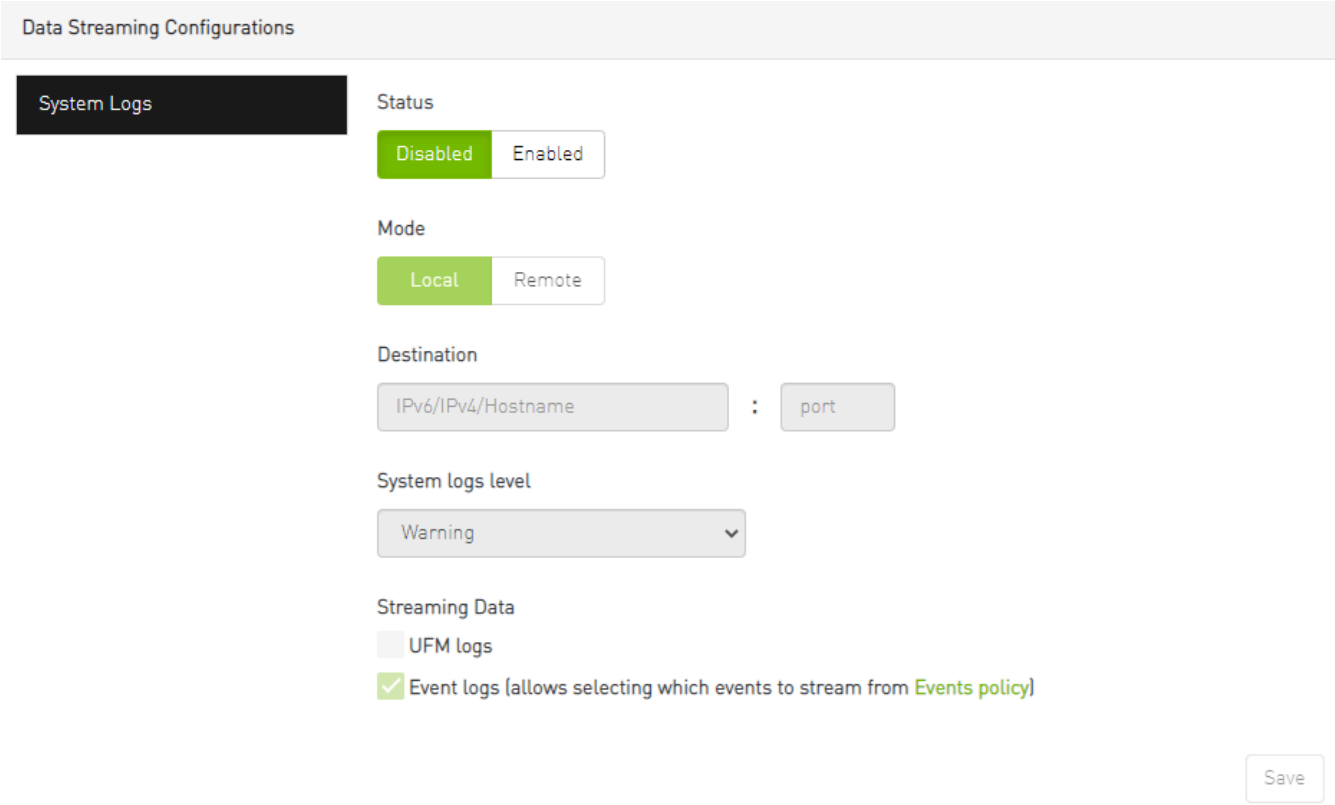


Field	Description
Protocol	The protocol to use to move the dump file to the external storage (scp/sftp)
Server	Hostname or IP address of the server
Path	The path where dump files are saved
Username	Username for the server
Password	Respective password

After configuring these parameters, it would be possible for users to collect sysdumps for specific devices, groups, or links (through Network Map/Cables Window) by right-clicking the item and selecting System Dump.

Data Streaming

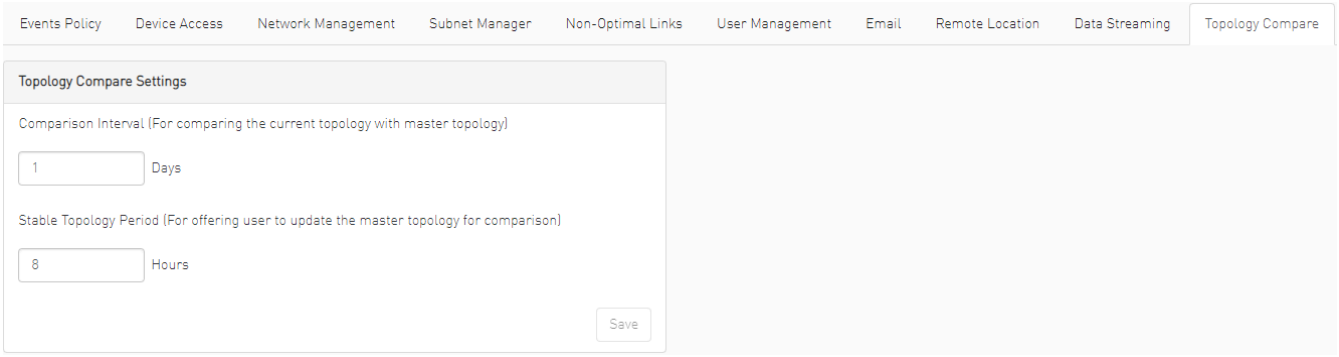
This section allows users to configure System Logs settings via web UI.



Field	Description
Status	Enable/disable exporting UFM logs to system logs
Mode	Export logs to local or remote system logs
Destination	Remote server IP/hostname and port
System Logs Level	Log level to export
Streaming Data	Logs to export to system logs. **Note** Events logs are selected one by one from Events Policy settings when the system logs feature is enabled.

Topology Compare

This tab controls the settings for the Periodic Topology Comparison feature.



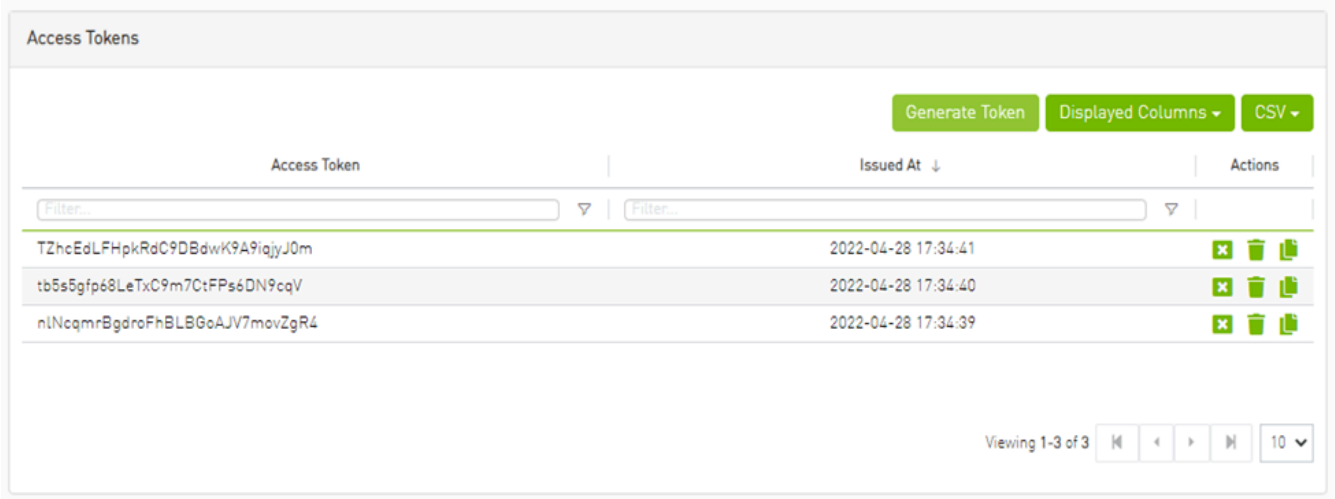
- Comparison Interval – determines how often the current topology is compared against the master topology
- Stable Topology Period – determines how long a topology must be stable before it is designated the new master topology

Token-based Authentication

Token-based authentication is a protocol which allows users to verify their identity, and in return receive a unique access token. During the life of the token, users then access the UFM APIs that the token has been issued for, rather than having to re-enter credentials each time they need to use any UFM API.

> **Note**
>
> Under the Settings section there is a tab titled called “Access Tokens”.

The functionality of the added tab is to give the user the ability to create new tokens & manage the existing ones (list, copy, revoke, delete):



Actions:

Name	Icon	Description
Revoke		Revoke a specific token. **Note** The revoked token will no longer be valid.
Delete		Delete a specific token.
Copy		Copy specific token into the clipboard.

Note

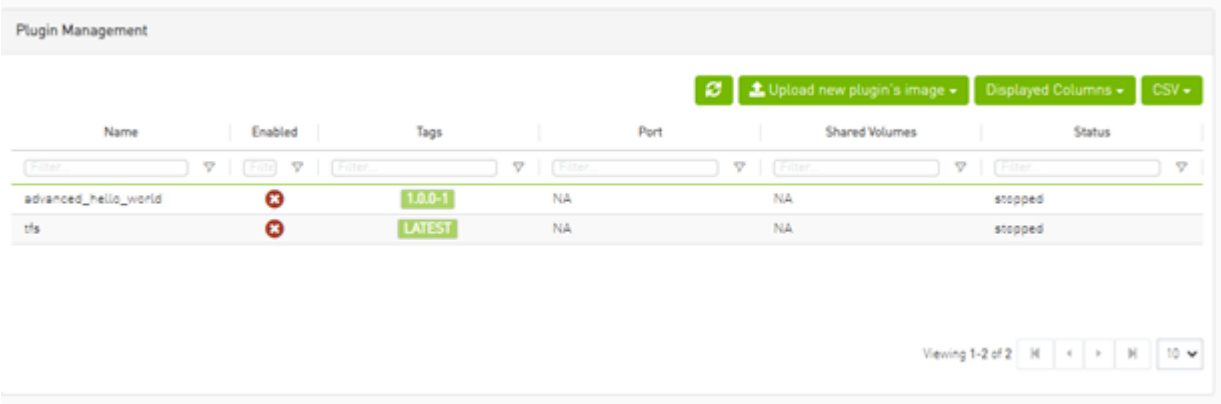
Each user is able to list and manage only the tokens that have been created by themselves. Only the users with system_admin role will be able to create tokens.

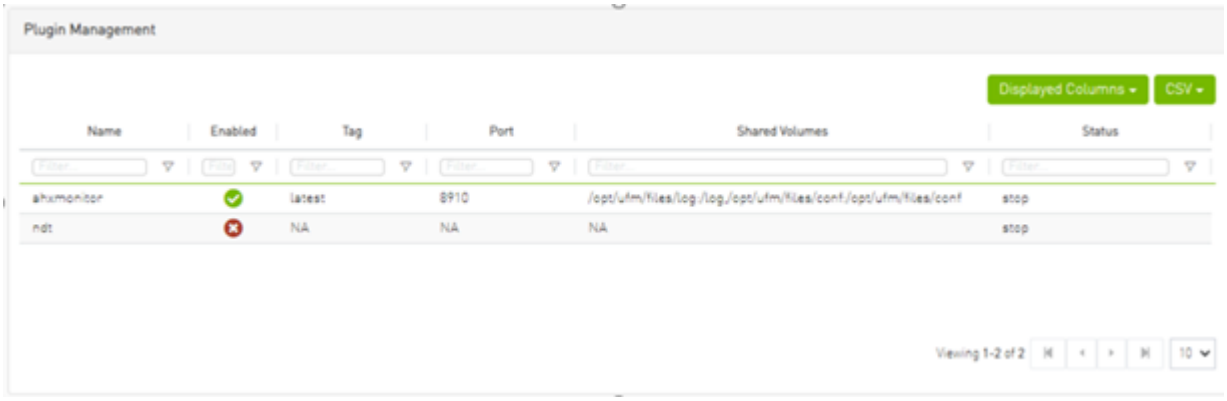
Plugin Management

Plugin management allows users to manage UFM plugins without using CLI commands. Under "Settings", there is a tab titled “Plugin Management”.

The functionality of the "Plugin Management" tab is to give the user the ability to add, remove, disable and enable plugins.

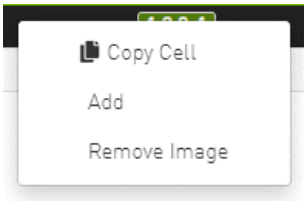
Furthermore, the plugin management feature allows loading a plugin's image in two ways: either by remotely pulling it from a Docker Hub repository or by directly uploading the image file from the user's local machine.



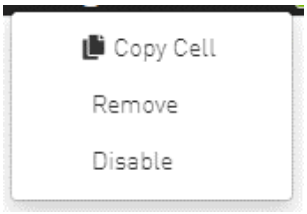


Actions:

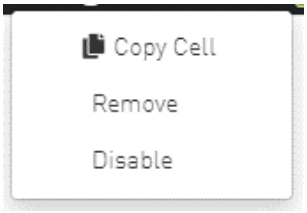
- Add – Used to add a selected plugin, opens a model to select the needed tag.



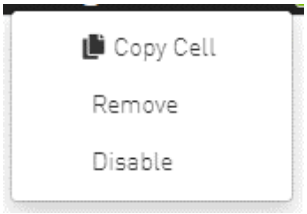
- Remove – Used to remove a selected plugin.



- Disable – Used to disable a selected plugin, so the plugin is disabled once the UFM is disabled.

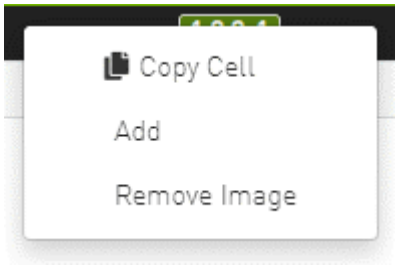


- Enable – Used to enable a selected plugin, so the plugin is enabled once the UFM is enabled.

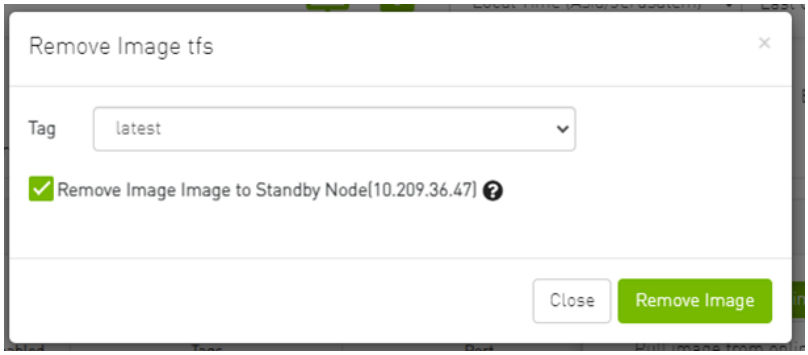


- Add ahxmonitor – Used to add a selected plugin; the action opens a modal to select the requested tag.

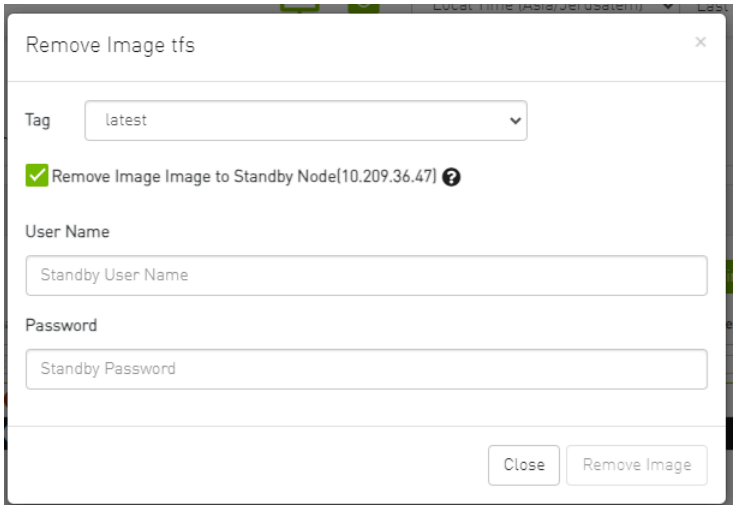
- Remove plugin Image – Used to remove plugin image



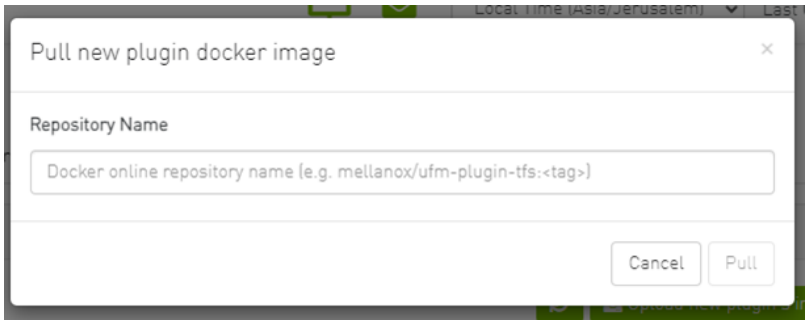
If the high availability (HA) mode is enabled, the user will see the option to remove the image from the standby node as well.



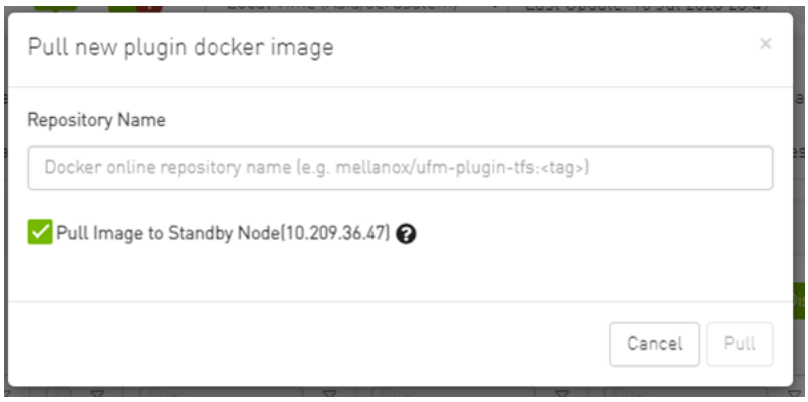
In cases where there is no established trust communication between the master and standby nodes, the user will be required to provide a username and password to establish an SSH connection between them.



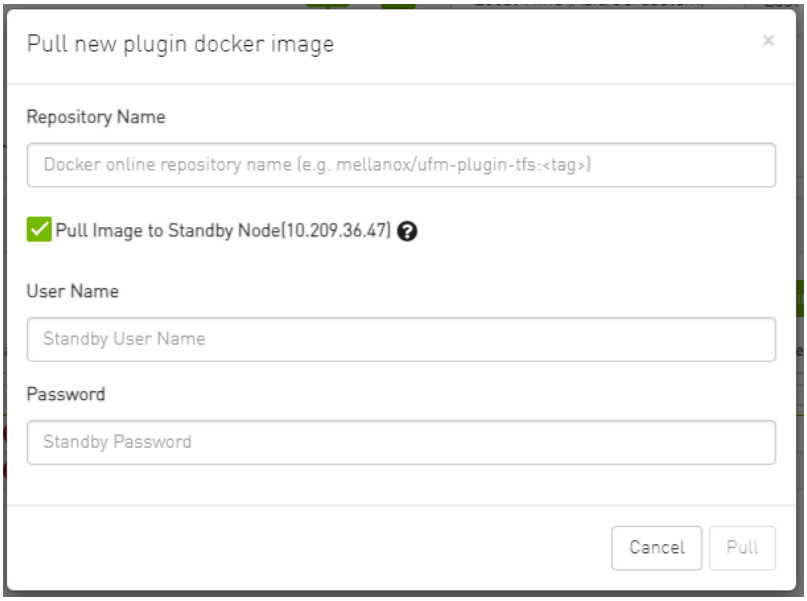
- Pull plugin Image – Used to pull plugin image remotely (e.g. from a Docker Hub repository) or by loading it from user local machine by uploading the image file itself.



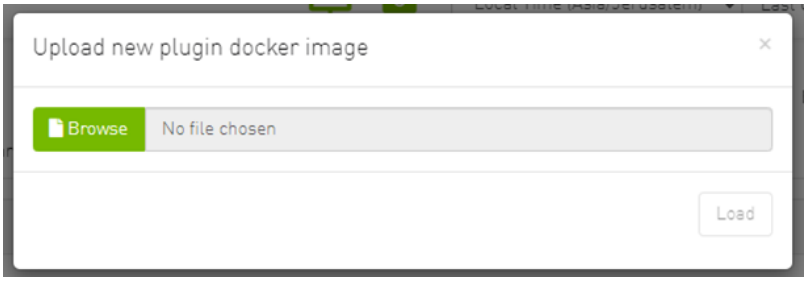
If the high availability (HA) mode is active, the user will be presented with the choice to pull the image to the standby node as well.



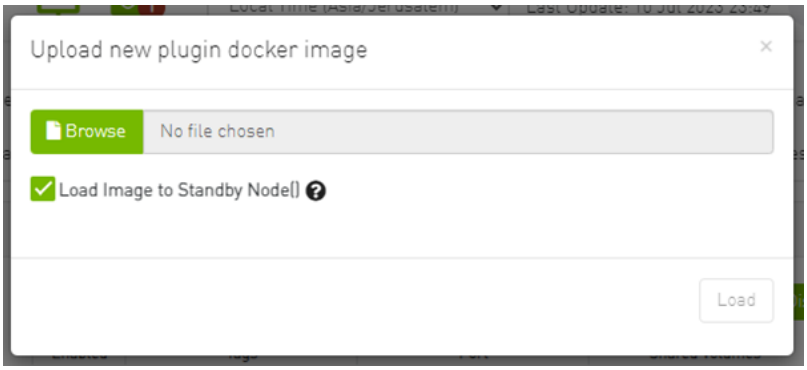
Once again, in the absence of trusted communication between the master and standby nodes, the user will need to input a username and password to create an SSH connection between the nodes.



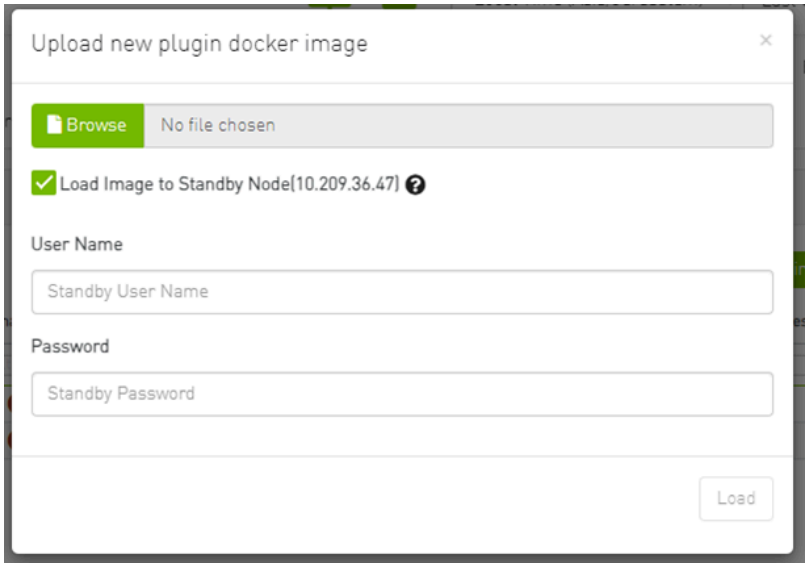
- Load plugin Image: this feature allows the user to upload the image file from their local machine directly.



Similarly, if the high availability (HA) mode is enabled, the user will have the option to load the image to the standby node too.



And, as mentioned earlier, if there is no trusted communication between the master and standby node, the user will need to provide a username and password to establish an SSH connection between the nodes.



Rest Roles Access Control

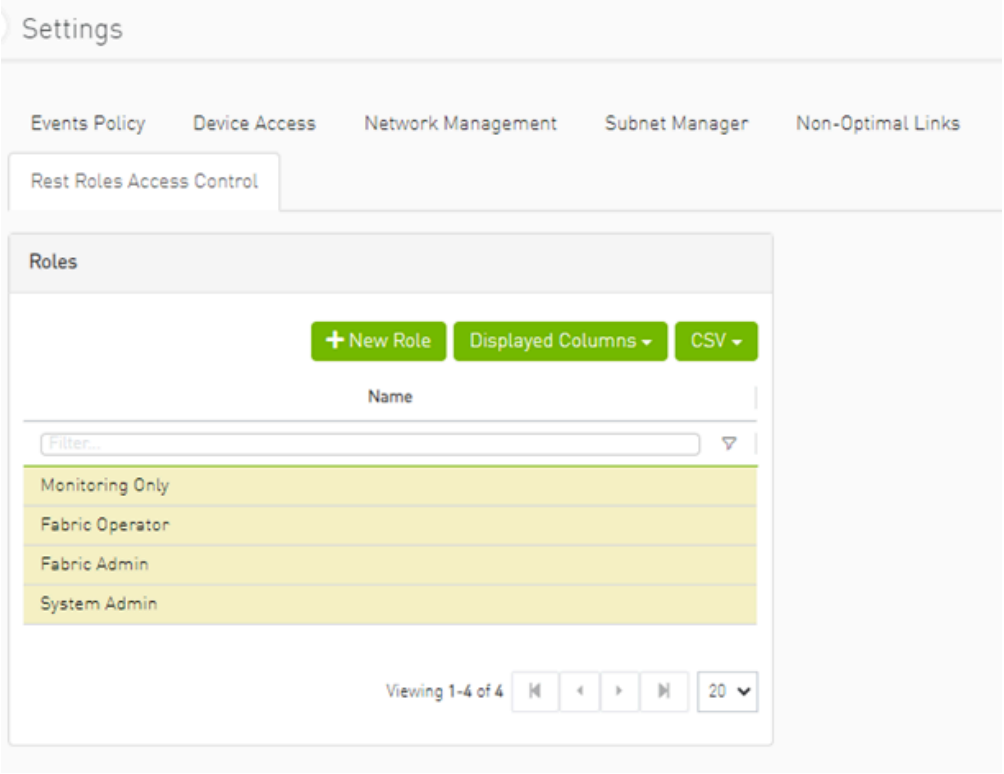
In UFM, there are four predefined roles with the following corresponding values:

1. System Admin (Role value: 5)
2. Fabric Admin (Role value: 4)
3. Fabric Operator (Role value: 3)
4. Monitoring Only (Role value: 2)

For more information, refer to the User Management Tab.

The "Rest Roles Access Control" tab empowers Admin users to design their custom roles alongside the existing predefined roles. Admins can set permissions and access levels for these custom roles, defining which APIs the roles can access.

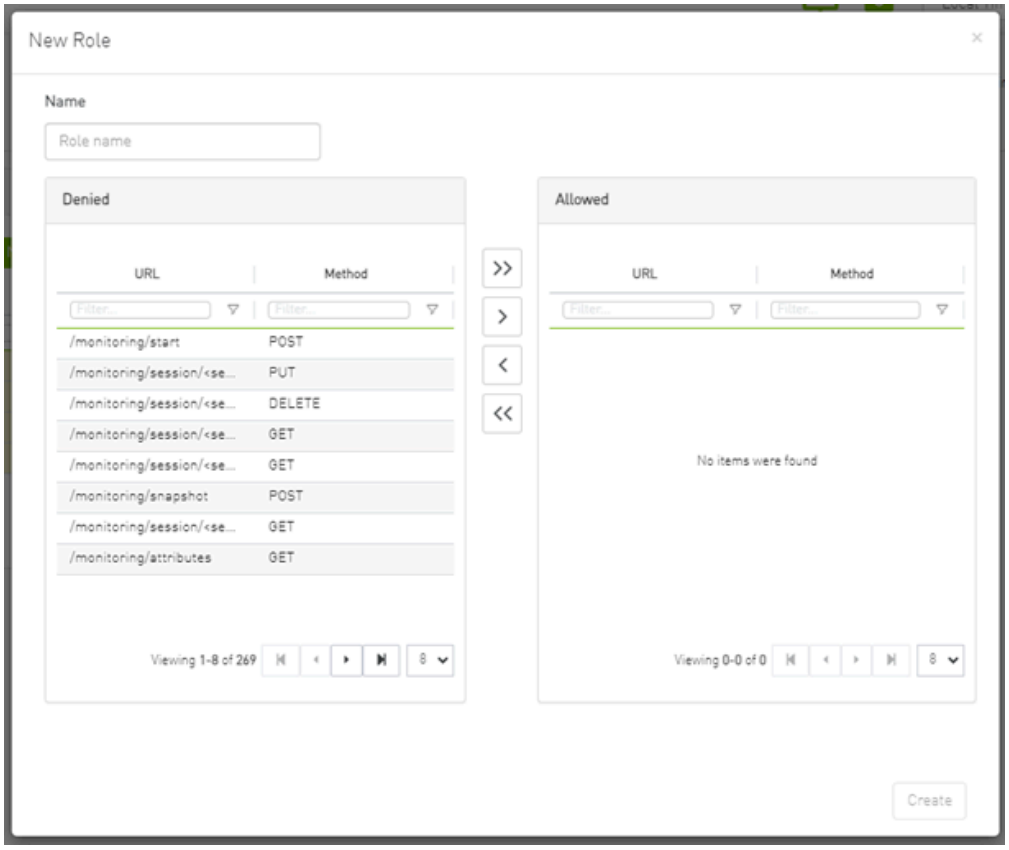
Roles are presented in a table format, with the predefined roles highlighted in yellow.



This tab is exclusively available to System_Admin users and can be enabled or disabled through the `gv.cfg` file. By default, it is enabled.

Adding a New Role

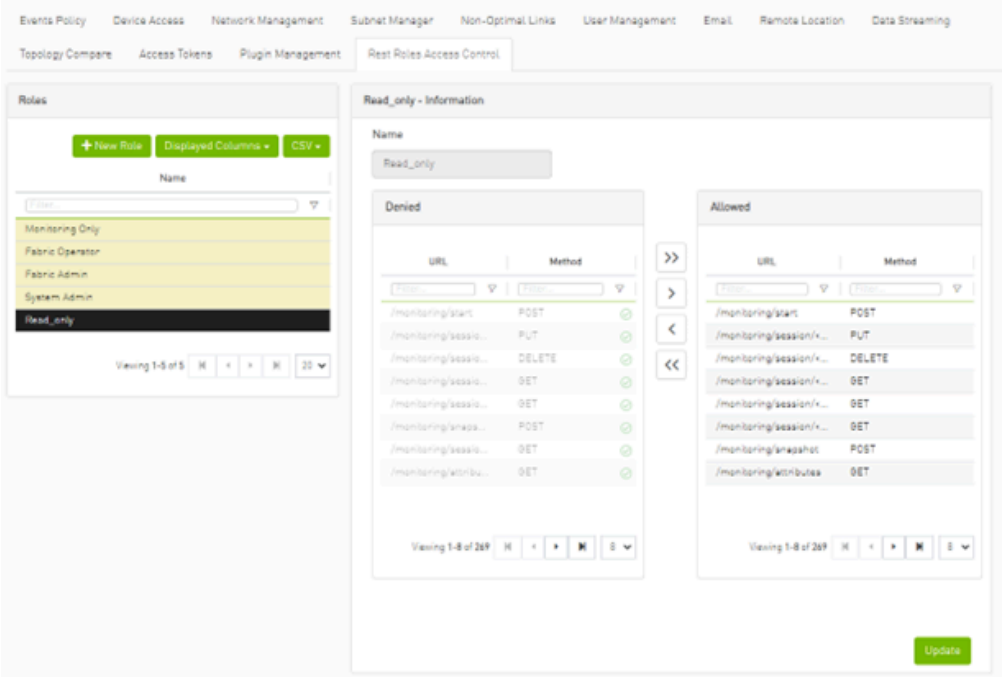
1. Click the + New Role button.
2. Fill in the necessary details in the dialog box.



By default, all URLs are denied. To allow specific URLs for this role, move them to the "allowed" category.

Updating Custom Roles

1. Select the role that requires updating.

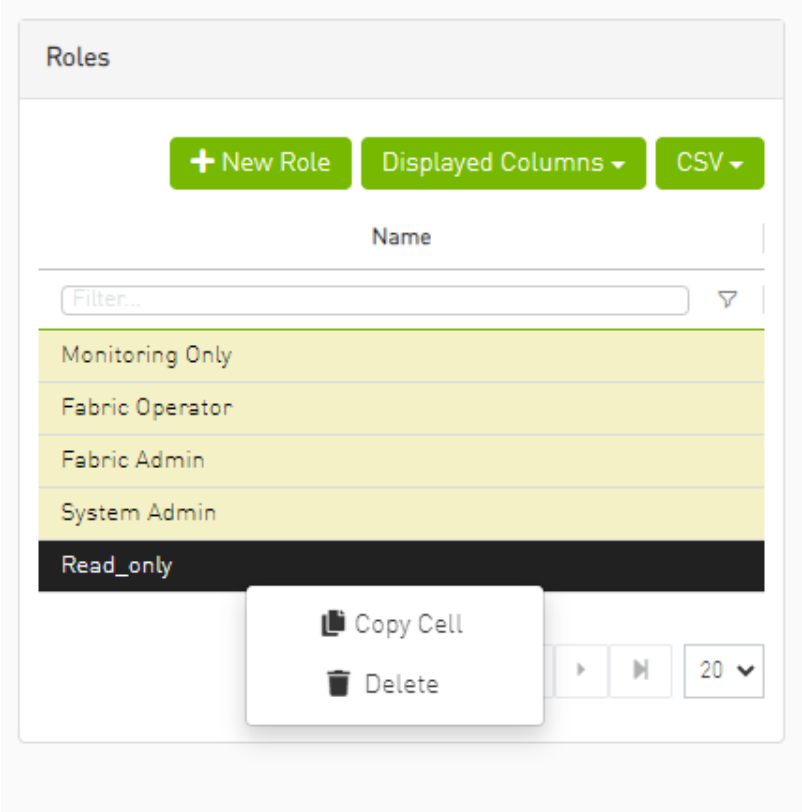


2. Modify the allowed list from the role information section.

Deleting Custom Roles

1. Right-click on the role that needs deletion.

2. Choose the "Delete" option from the context menu.

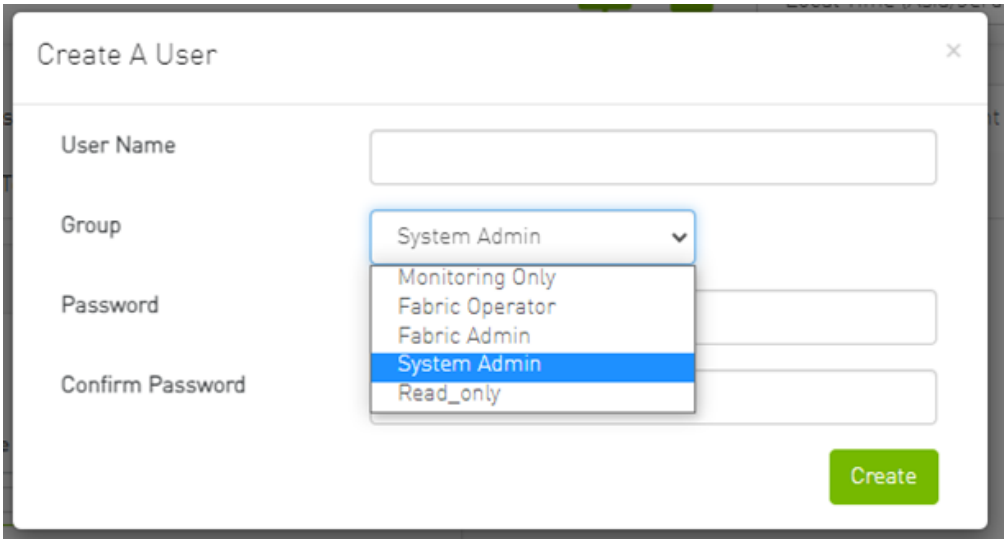


Note

Deleting and updating predefined roles is not permitted.

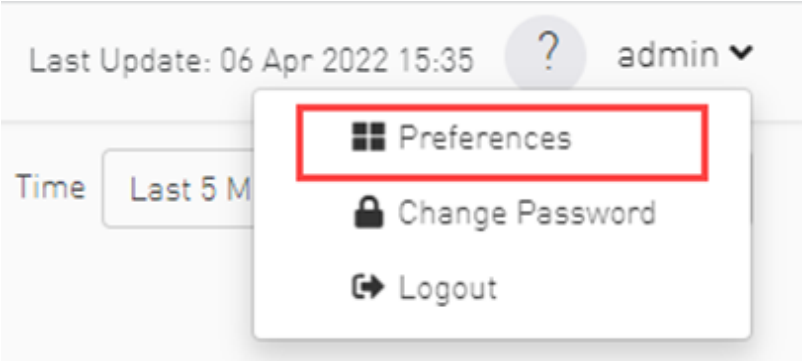
Creating a User with a Custom Role

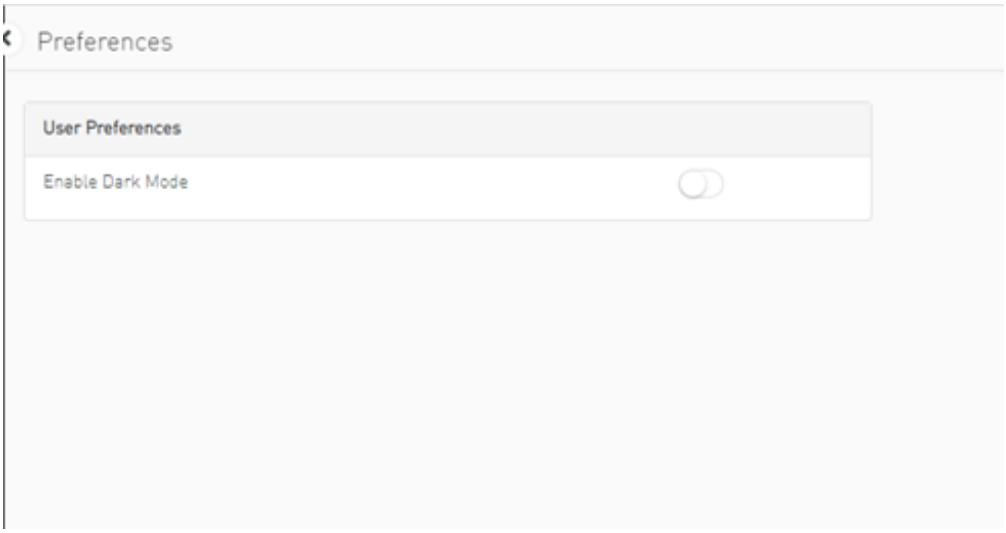
1. Navigate to the Users Management tab.
2. Create a new user, and you will find all roles (both custom and predefined) listed under the group list.



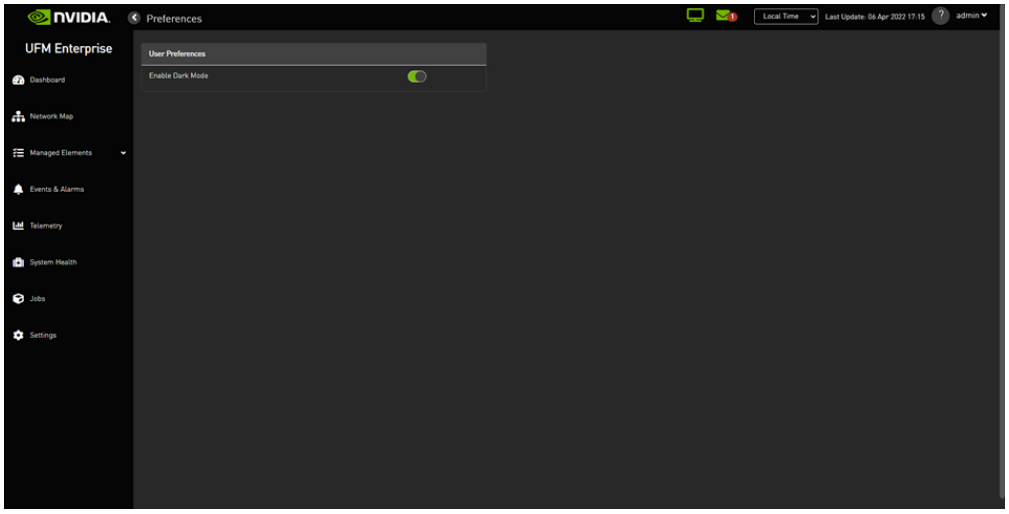
User Preferences

This page allows user to change UI preferences in general.





When user enables dark mode, the UFM is presented in dark theme.


NVIDIA
UFM Enterprise
Preferences
Local Time
Last Update: 06 Apr 2022 17:15
admin
Dashboard
Network Map
Managed Elements
Events & Alarms
Telemetry
System Health
Jobs
Settings
User Preferences
Enable Dark Mode

Multi-Subnet UFM

Overview

The Multi-Subnet UFM feature allows for the management of large fabrics, consisting of multiple sites, within a single product, namely Multi-Subnet UFM.

This feature is comprised of two layers: UFM Multi-Subnet Provider and UFM Multi-Subnet Consumer.

The UFM Provider functions as a Multi-Subnet Provider, exposing all local InfiniBand fabric information to the UFM consumer. On the other hand, the UFM Consumer acts as a Multi-Subnet Consumer, collecting and aggregating data from currently configured UFM Providers, enabling users to manage multiple sites in one place. While UFM Consumer offers similar functionality to regular UFM, there are several behavioral differences related to aggregation.

Setting Up Multi-Subnet UFM

In `/opt/ufm/files/conf/gv.cfg`, fill in the section named [`Multisubnet`] for UFM Multi-Subnet Provider and Consumer.

To set up UFM as a MultI-Subnet Provider, perform the following:

- Set `multisubnet_enabled` to `true`
- Set `multisubnet_role` to `provider`
- Set `multisubnet_site_name` (optional, if not set, it will be randomly generated); e.g., *`provider_1`*
- Start UFM

To set up UFM as a Multi-Subnet Consumer, perform the following:

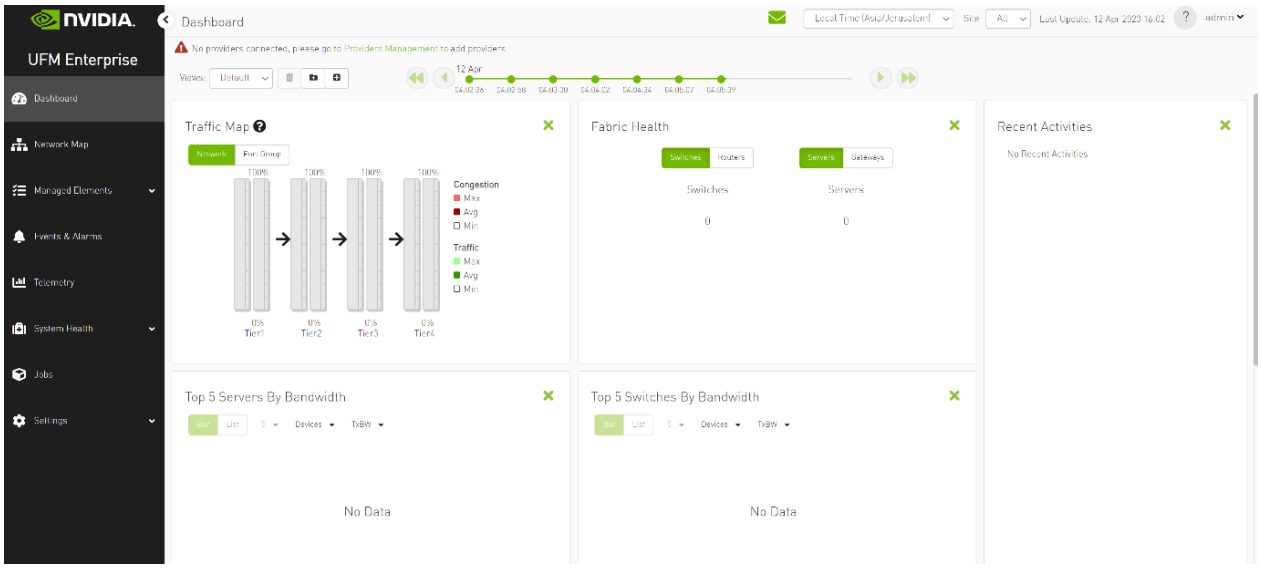
- Set `multisubnet_enabled` to `True`
- Set `multisubnet_role` to `consumer`

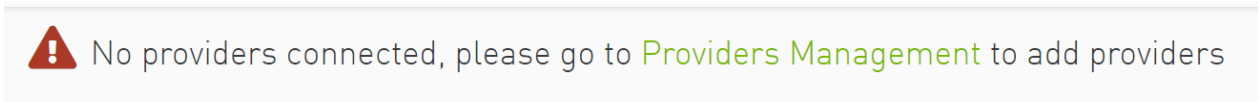
- Start UFM

It is important to note that UFM Multi-Subnet Consumer can be configured on a machine or VM without an established InfiniBand connectivity. Additionally, users may customize UFM Provider and Consumer using optional configuration parameters found in the [`Multisubnet`] section of `/opt/ufm/files/conf/gv.cfg`.

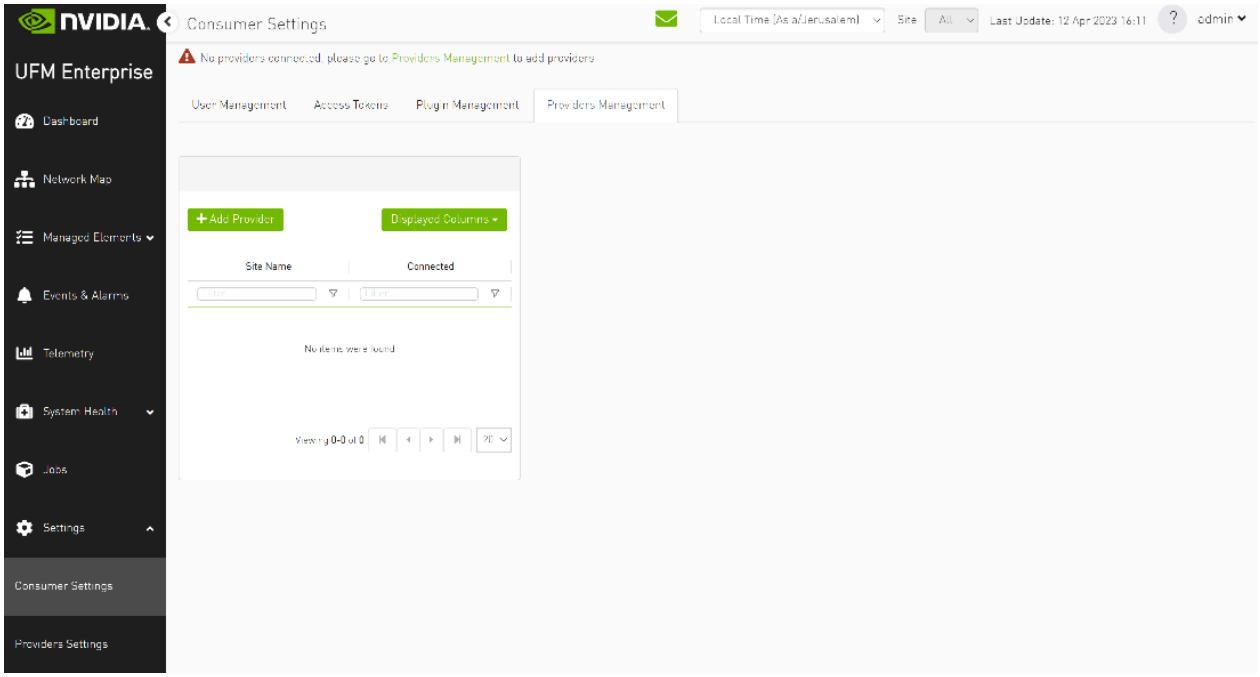
Functionality

1. Following the initial launch of the Consumer, the Dashboard view is devoid of data, and a message containing a hyperlink leading to the Provider Management section is displayed.

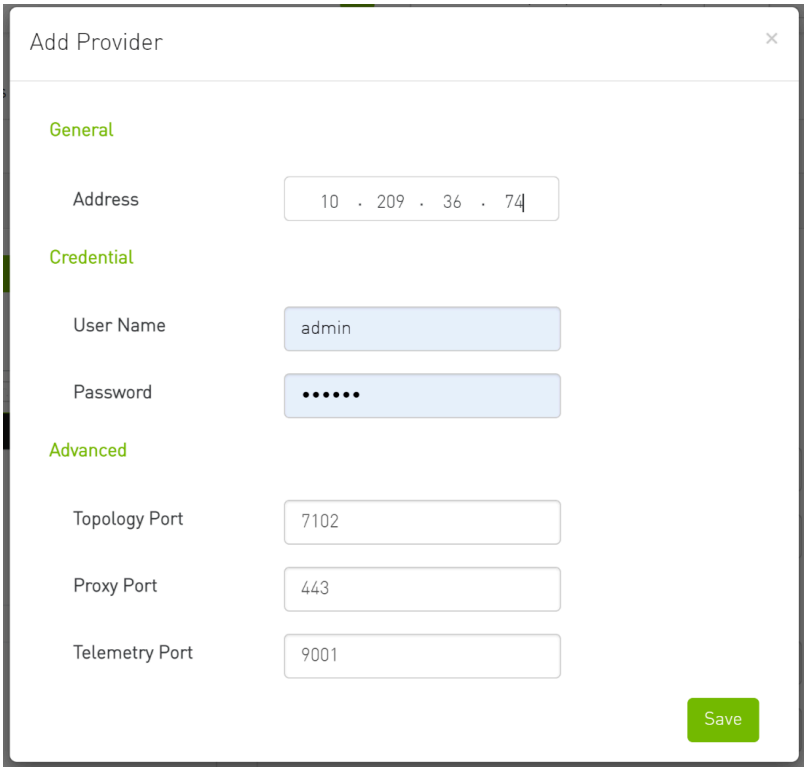




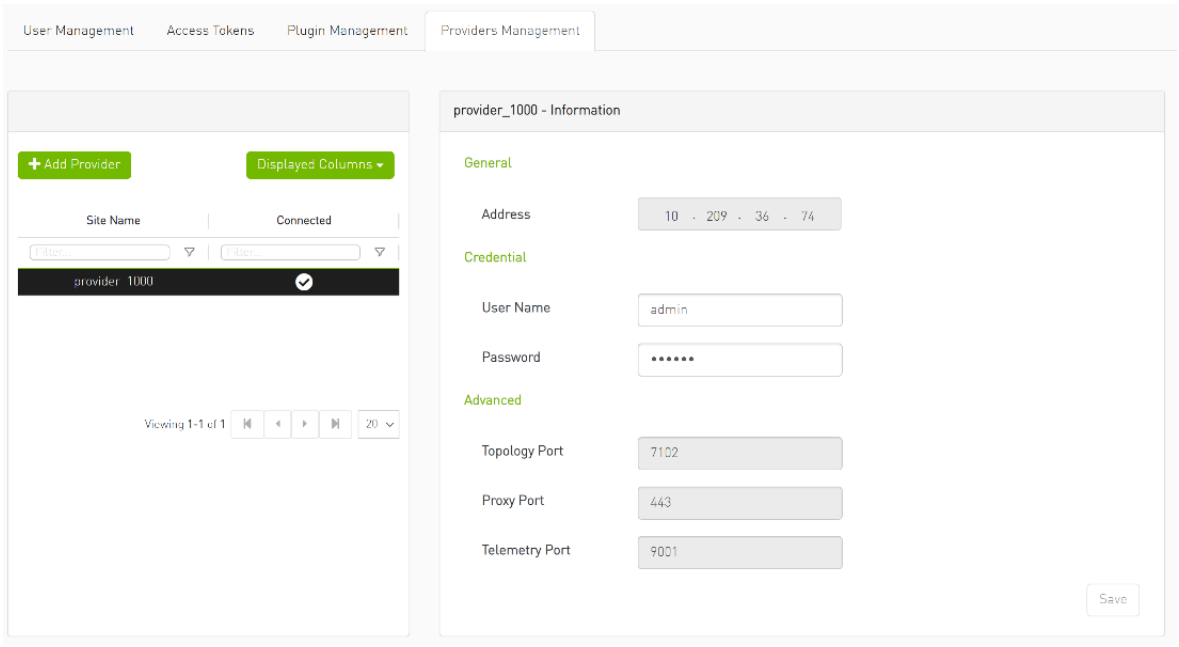
2. As shown in the below snapshot, a new section for Provider Management has been added, enabling users to configure UFM Providers.



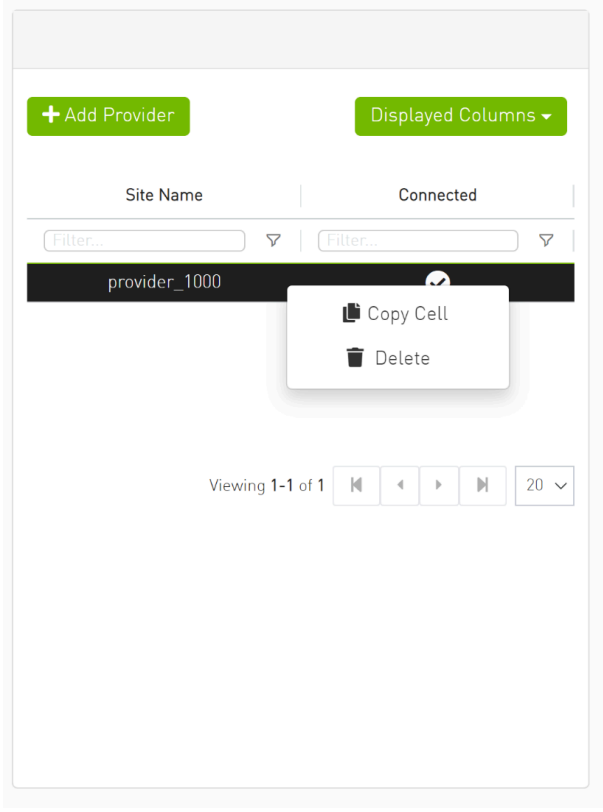
1. To add a provider, the user is required to enter its IP address and credentials. Unless there are multiple instances of UFM providers on a single machine, the advanced section parameters should be set with default values. However, if there are multiple instances, the advanced parameters may be set per Provider and then be configured in the Providers Management view.



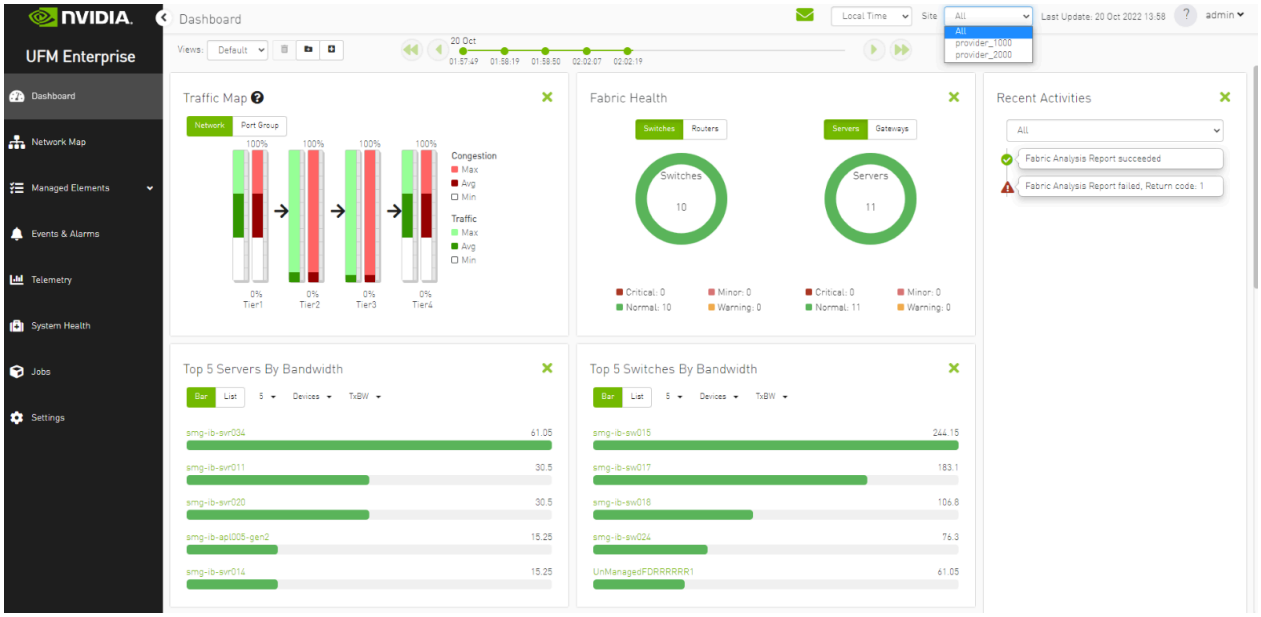
2. By editing the Provider view, you can change Provider's credentials.

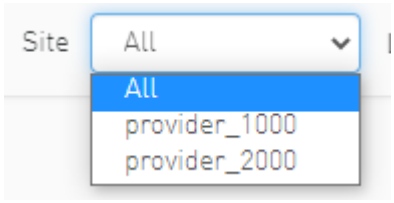


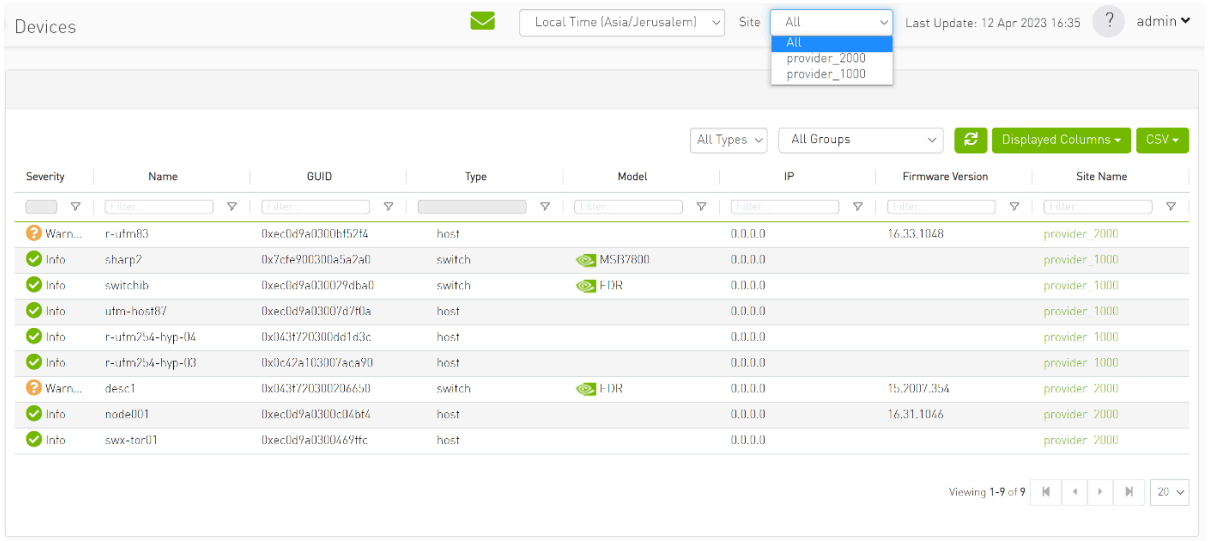
3. The "Delete Provider" function removes the selected Provider from the Consumer. Please note that this action may take some time to complete, and changes may only be reflected in the view after approximately 30 seconds.

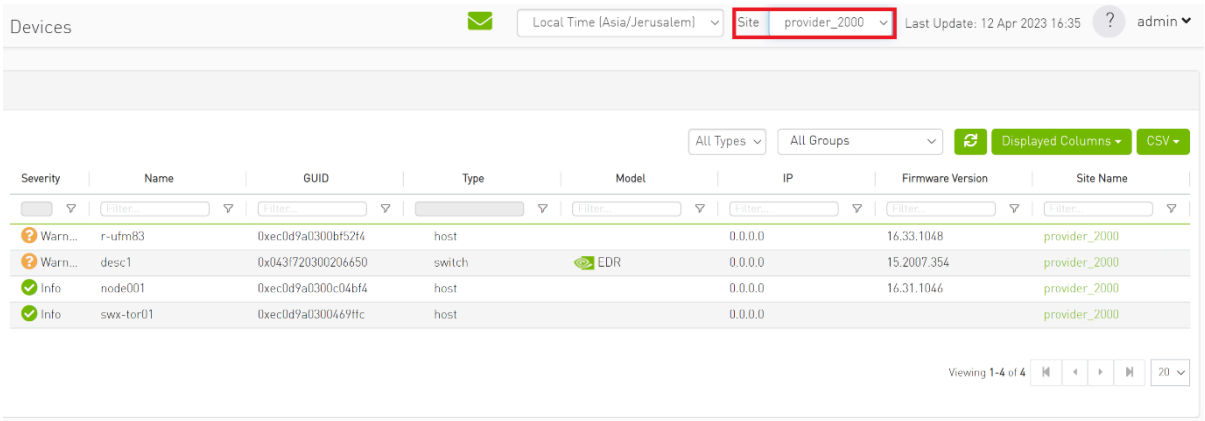


3. A general filter has been added to the top right corner of the page, enabling users to filter displayed data by site.

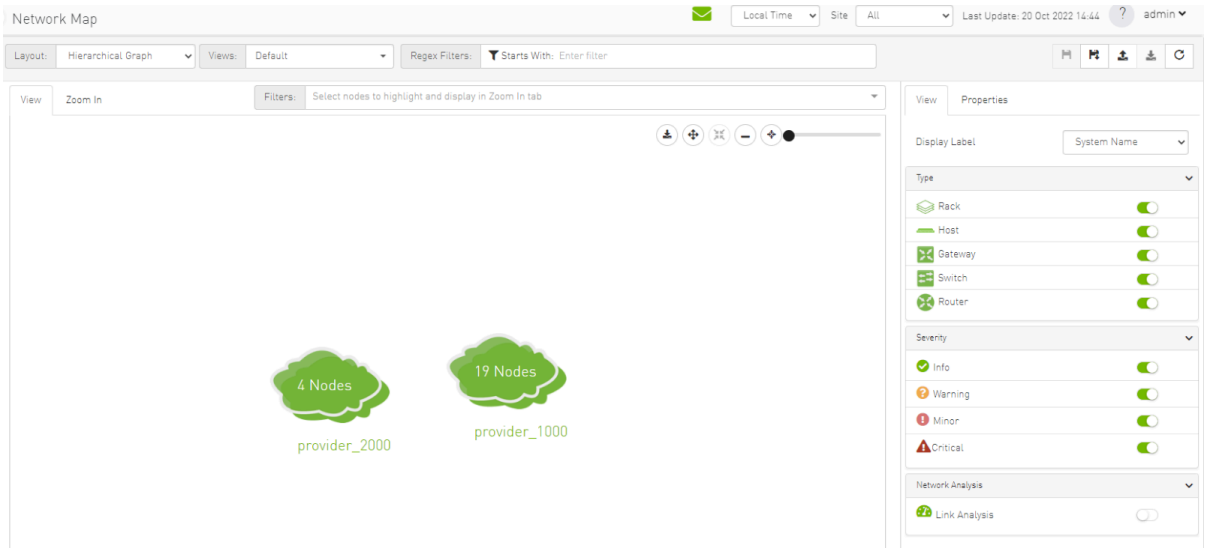

NVIDIA
UFM Enterprise
Dashboard
Network Map
Managed Elements
Events & Alarms
Telemetry
System Health
Jobs
Settings
Traffic Map
Fabric Health
Recent Activities
Top 5 Servers By Bandwidth
Top 5 Switches By Bandwidth

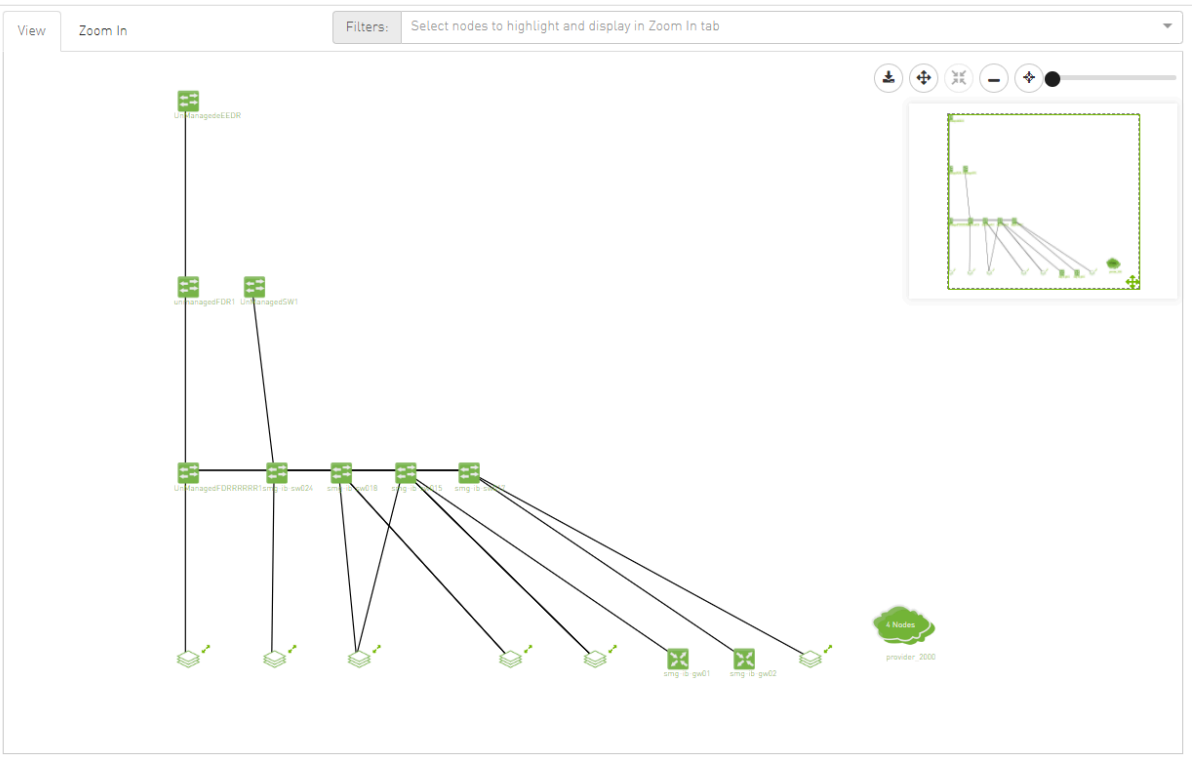

Site
All
All
provider_1000
provider_2000


Devices
Local Time (Asia/Jerusalem)
Site
All
Last Update: 12 Apr 2023 16:35
admin
All
provider_2000
provider_1000
All Types
All Groups
Displayed Columns
CSV
Severity
Name
GUID
Type
Model
IP
Firmware Version
Site Name

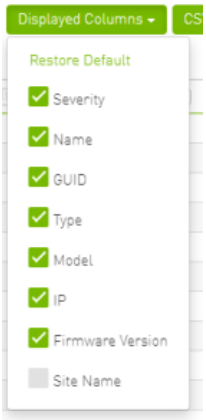


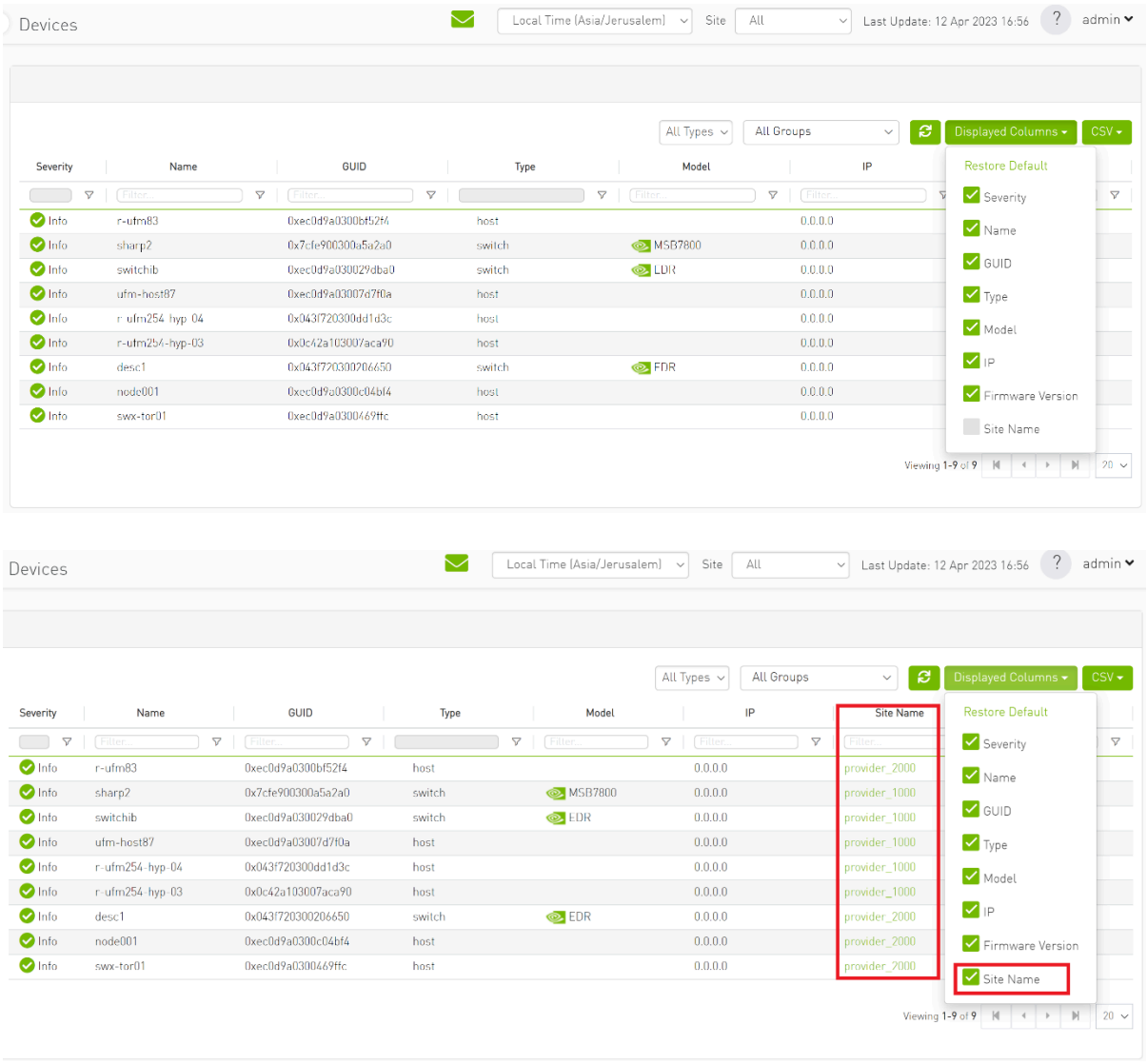
4. Network map contains “clouds” for each provider.



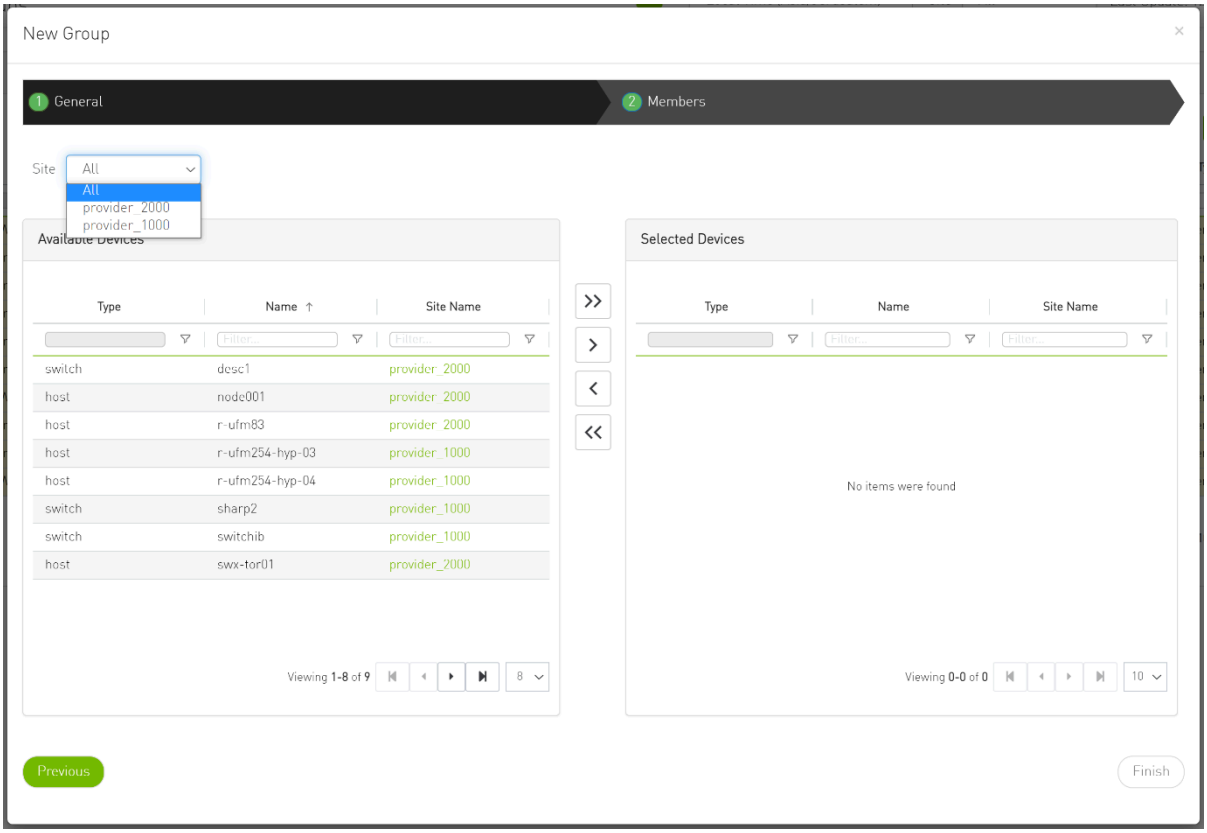


5. A "Site Name" column is present in all Managed Elements sections. The column is disabled (hidden) by default.

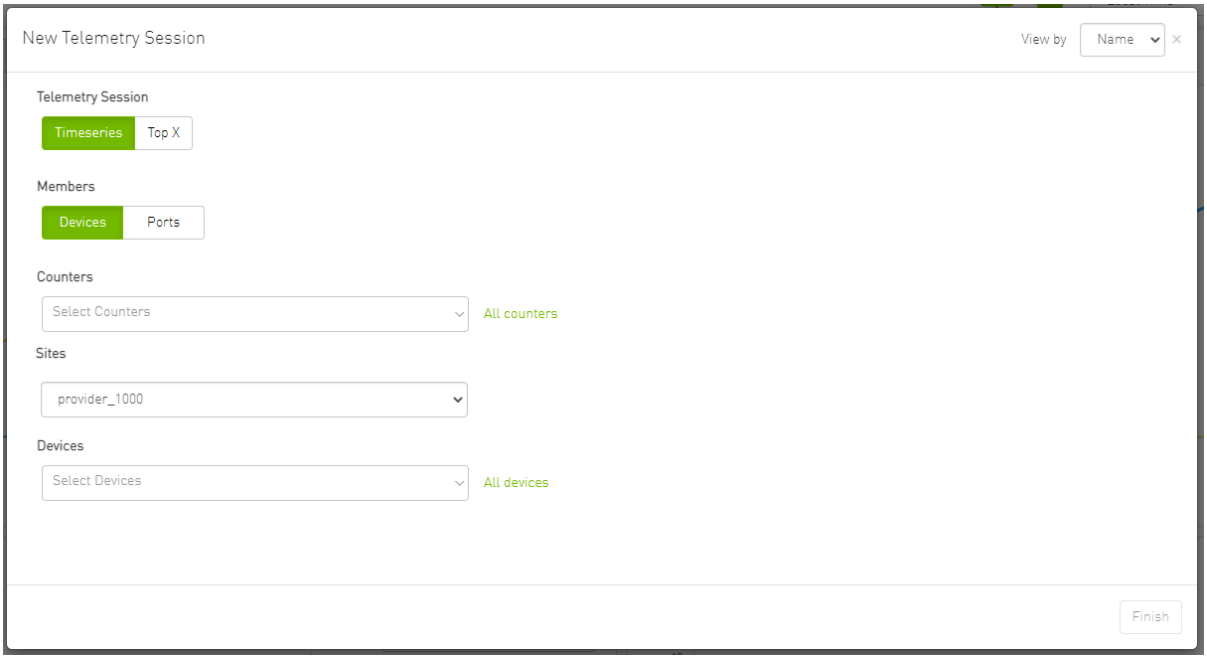


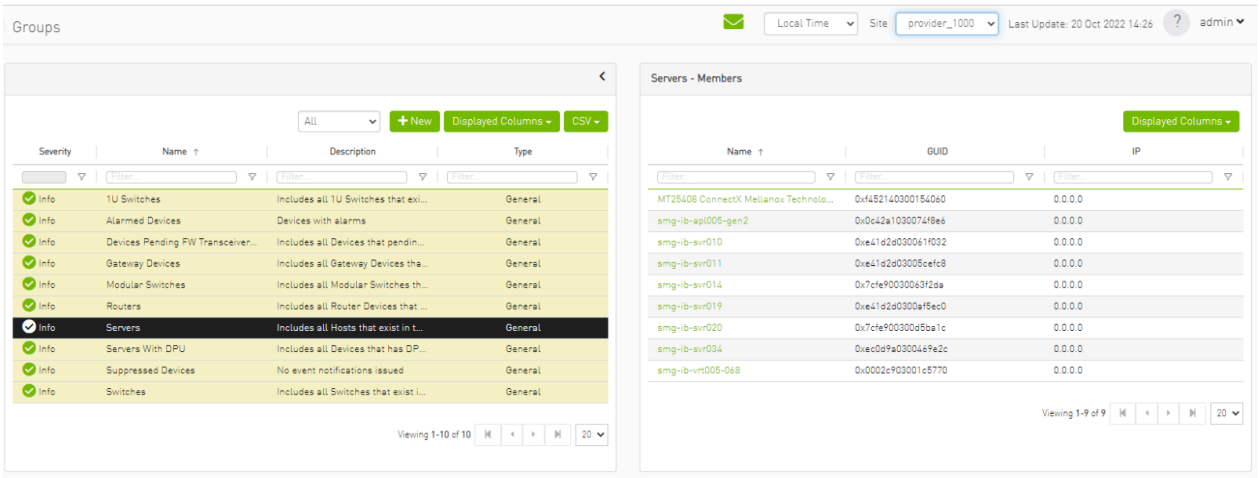


6. The "Group" and "Telemetry" sections include "Site" filters.



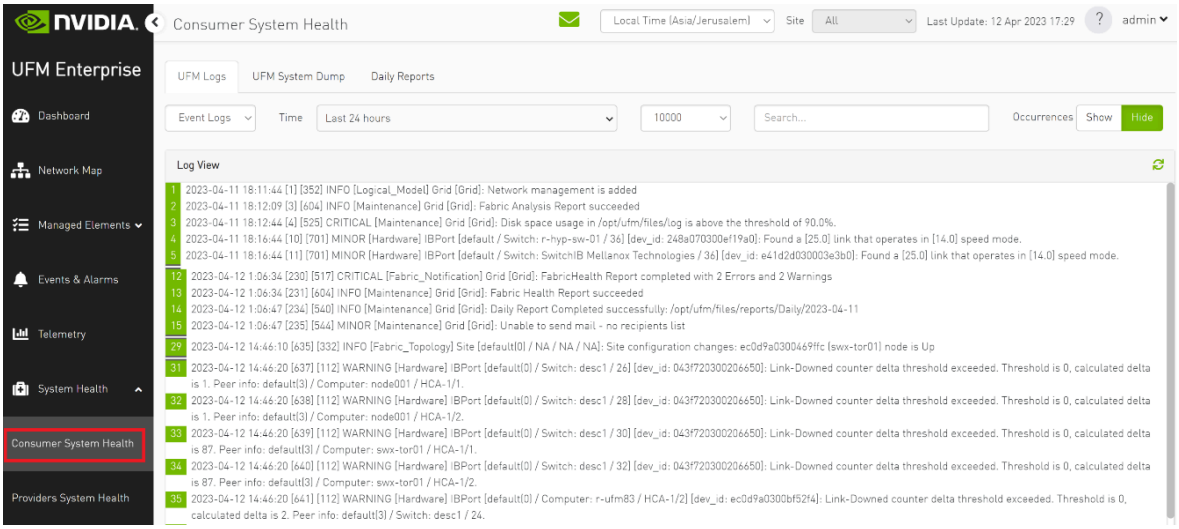
7. The filter in "Groups" impacts the Members table only.



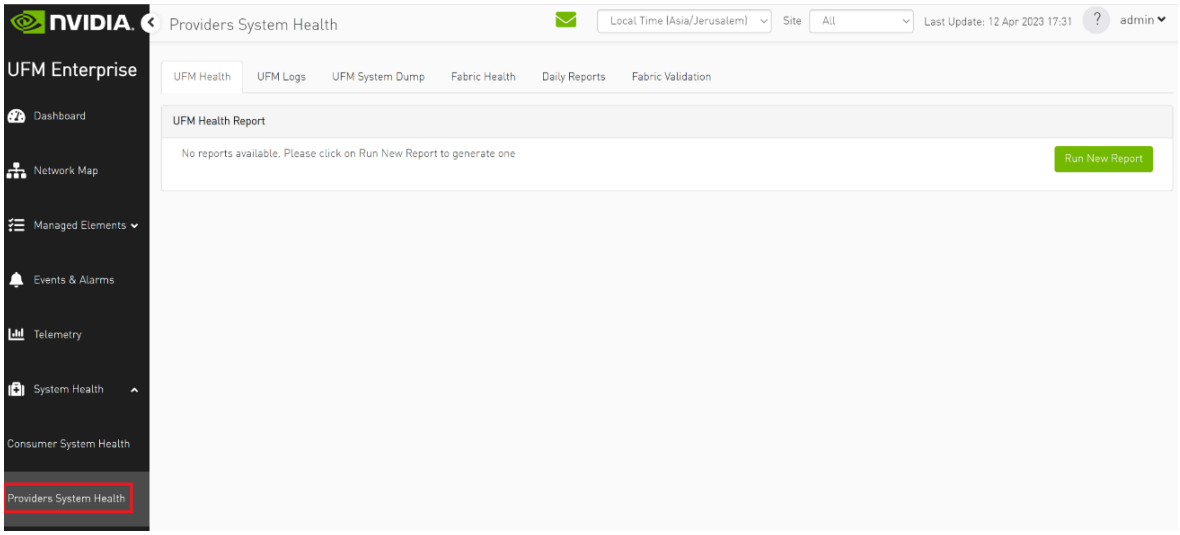


8. In the System Health tab, subsections for Consumer and Provider are available.

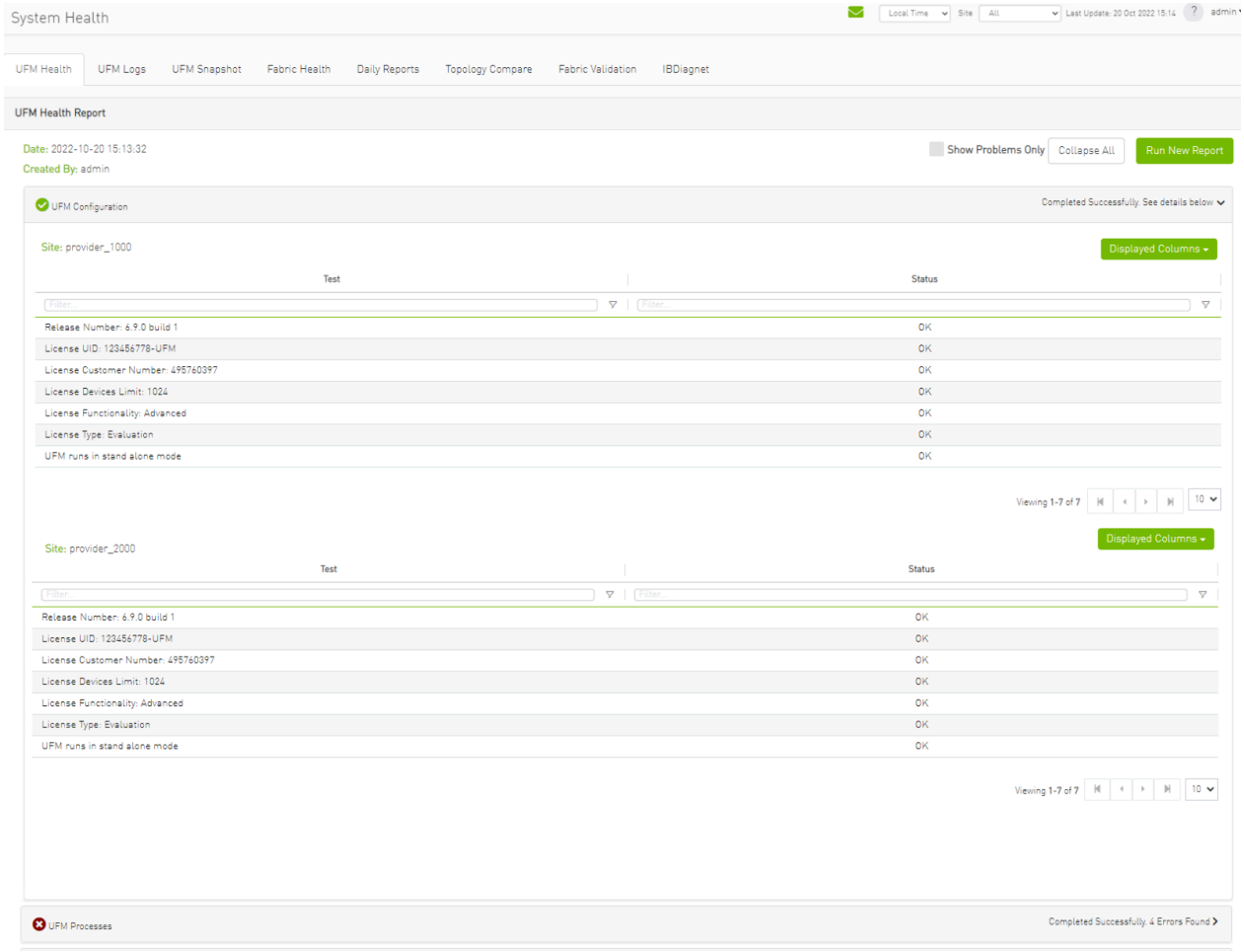
    1. Consumer System Health tab contains sections applicable to Consumer UFM specifically (e.g., logs from Consumer UFM).



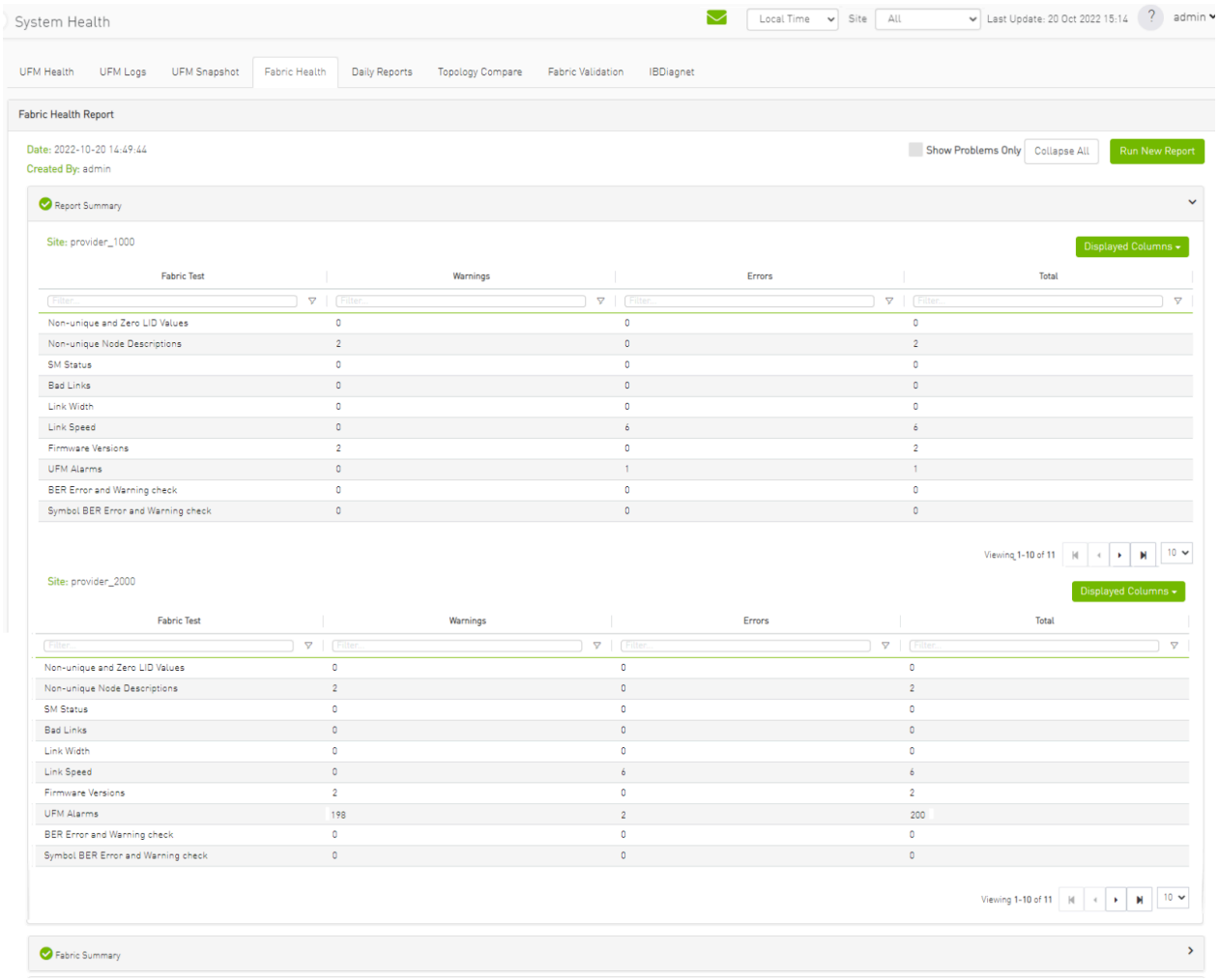
    2. Provider System Health contains sections applicable to one or multiple providers (e.g., Fabric Health Report can be triggered on multiple Providers from the Consumer).



9. UFM Health tab contains sub report tables for each provider.

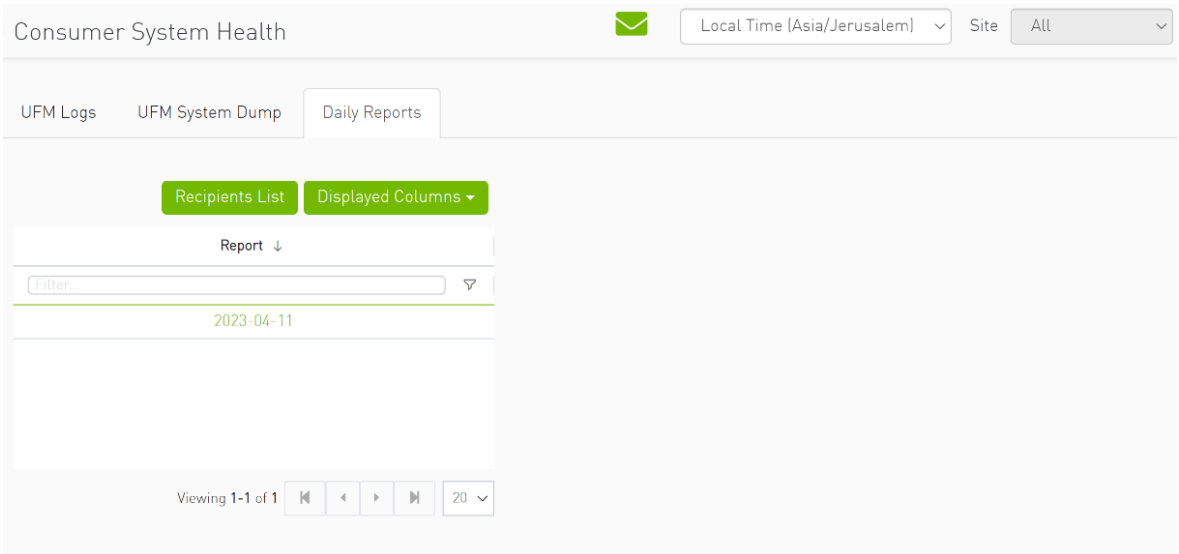


10. Fabric Health contains sub report tables for each provider.

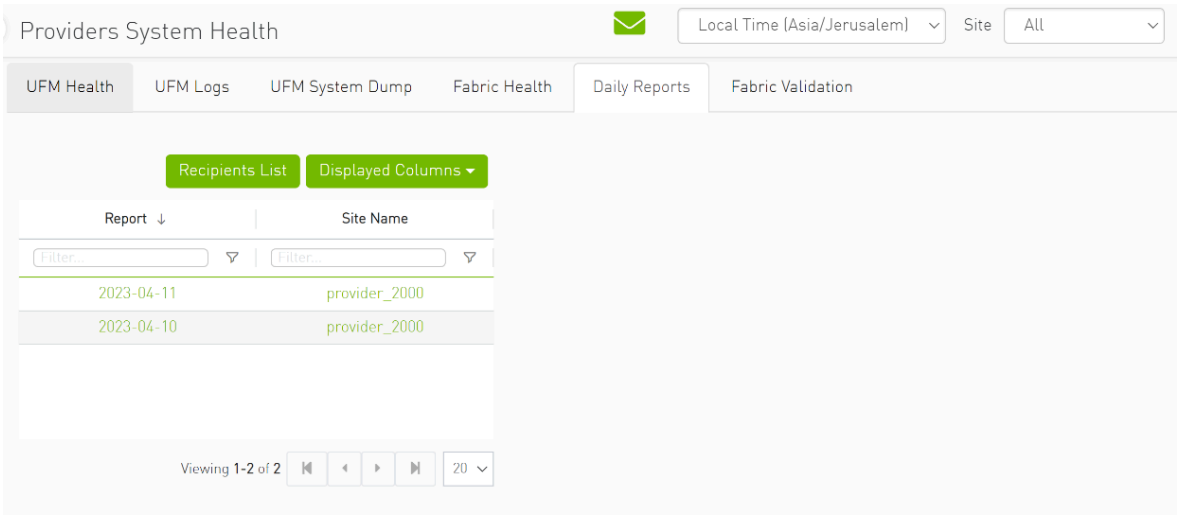


11. Daily Reports:

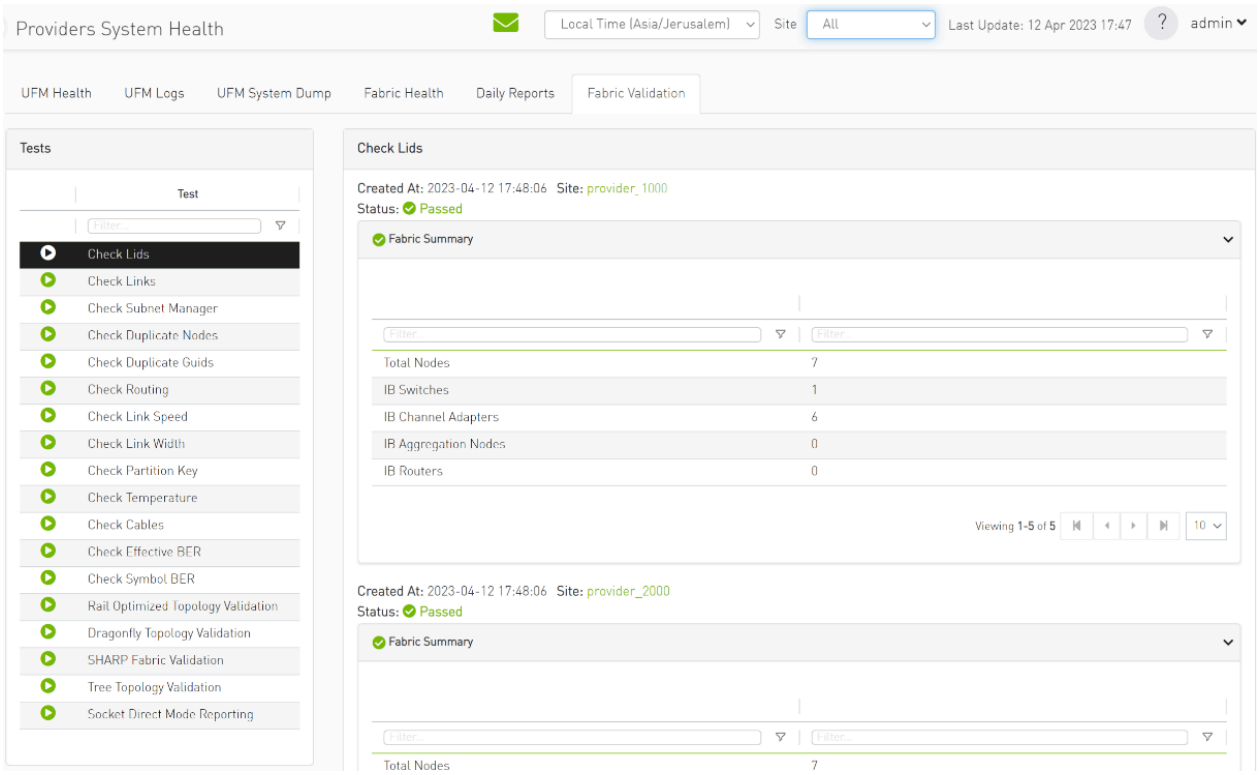
    1. Consumer Daily reports display consumer reports.



2. Providers Daily reports display reports from all providers.

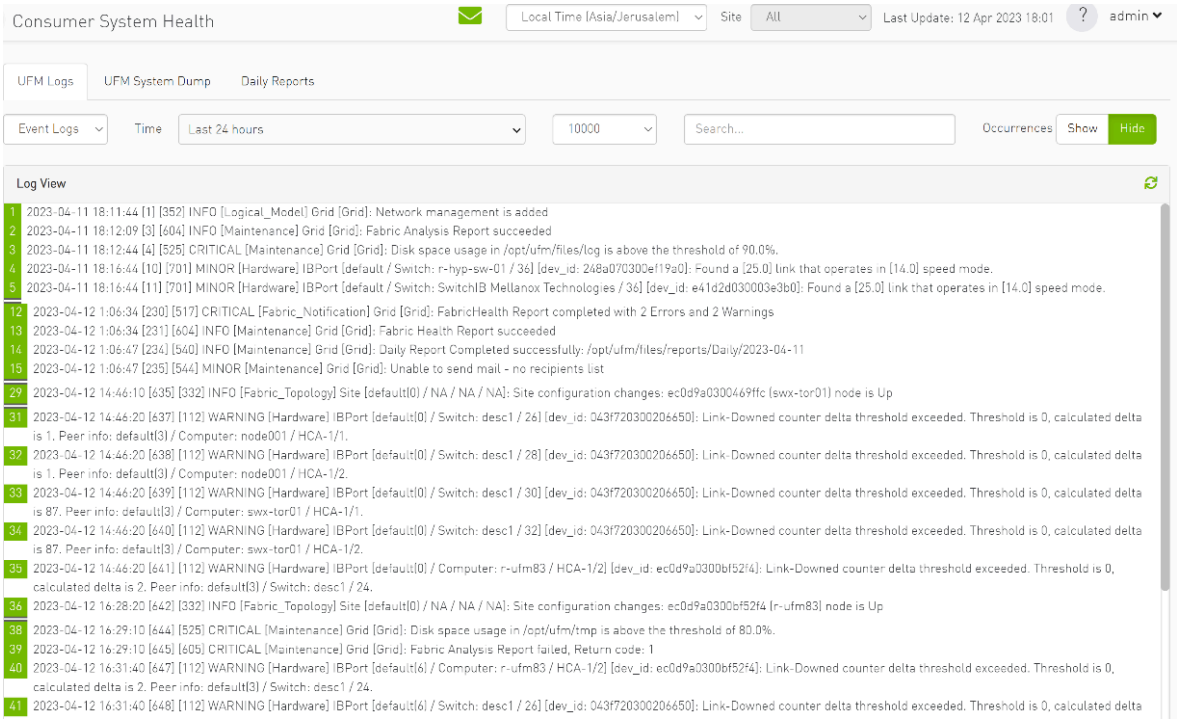


12. The "Fabric Validation" tab contains sub report tables for each provider.

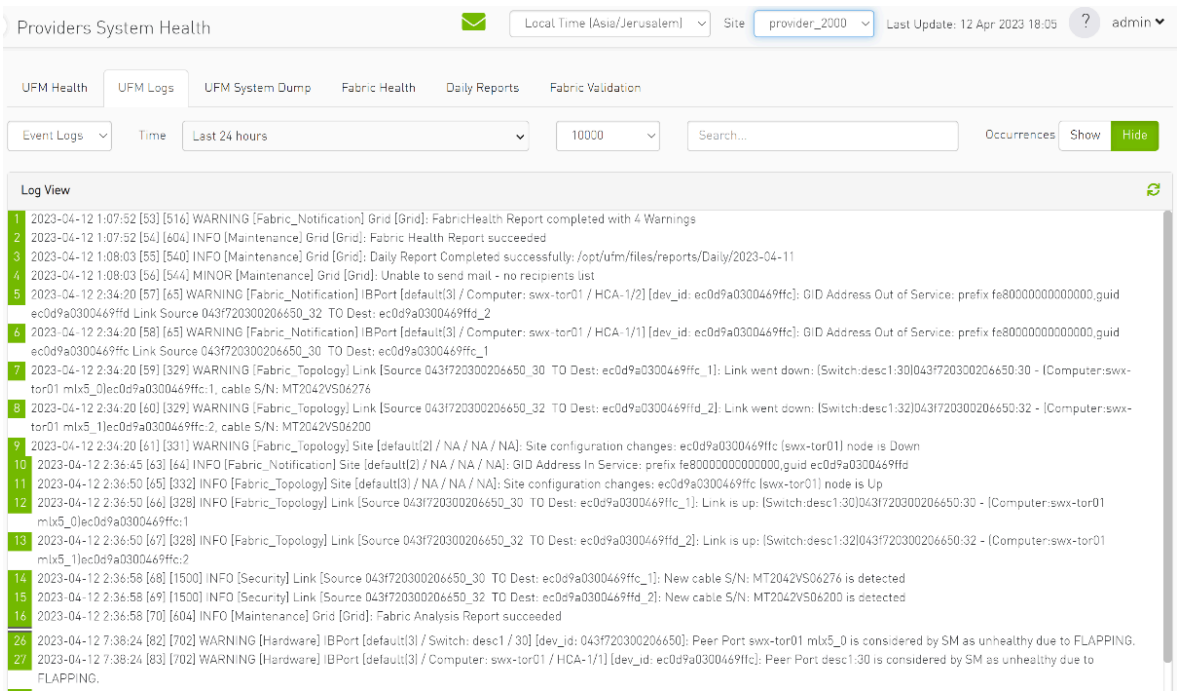


13. In "UFM Logs" Tab:

    1. Consumer logs:

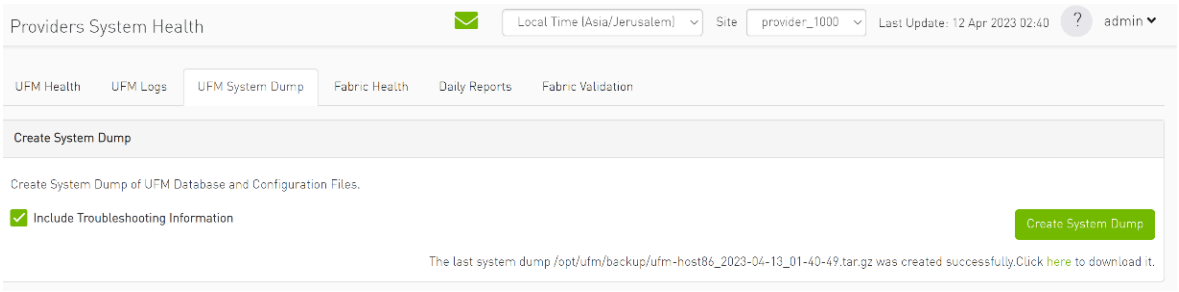


2. Providers logs display providers log separately, displaying logs for all providers is not supported.

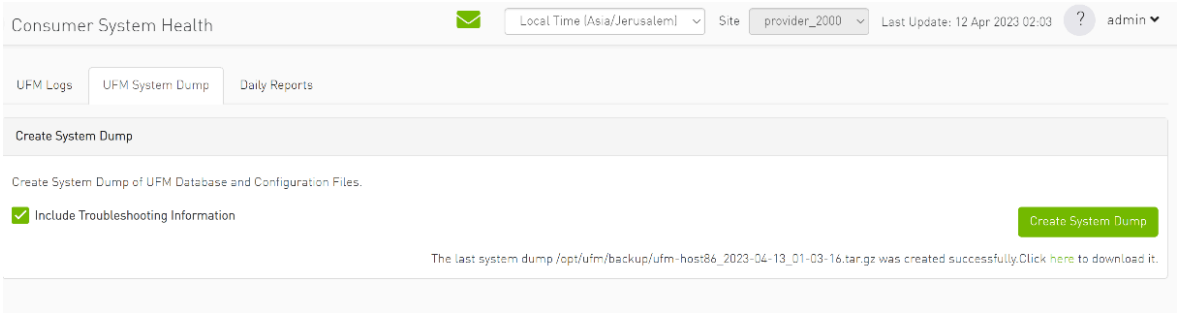


14. In the "System Dump" tab:

    1. "Consumer System Dump" collects system dump for consumer

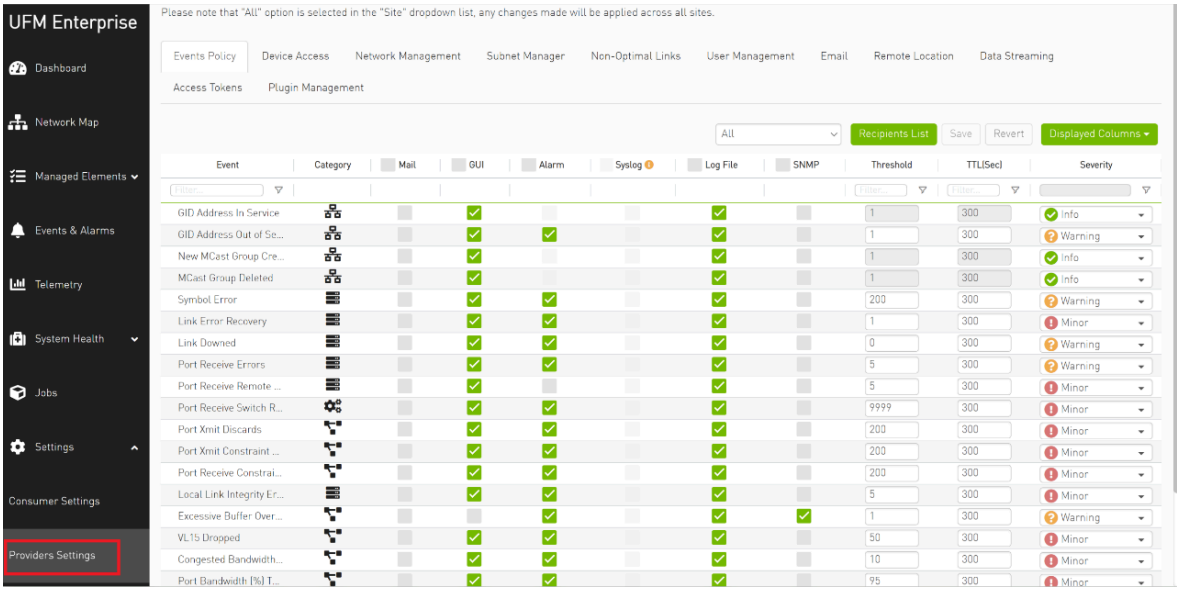


2. "Providers System Dump" collect system dumps for one or all providers and mergeS them into one folder

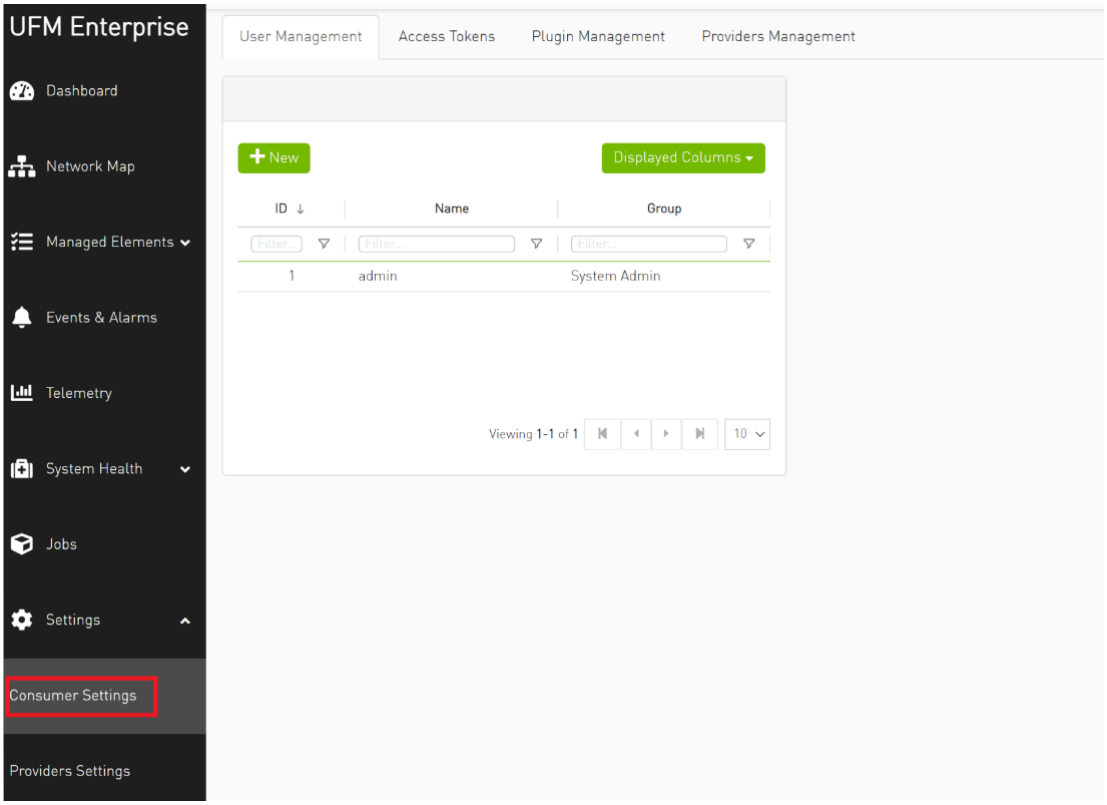


15. Under "Settings", subsections for Consumer and Provider are available.

    1. "Consumer Settings" contain sections applicable to Consumer UFM specifically (e.g., creation of access tokens for UFM consumer authentication);



    2. "Provider Settings" contain sections applicable to one or multiple providers (e.g., Event Policies can be changed for multiple Providers at once from the Consumer).


UFM Enterprise
Dashboard
Network Map
Managed Elements
Events & Alarms
Telemetry
System Health
Jobs
Settings
Consumer Settings
Providers Settings
User Management
Access Tokens
Plugin Management
Providers Management
New
Displayed Columns
ID
Name
Group
1
admin
System Admin
Viewing 1-1 of 1
10

UFM Plugins

rest-rdma Plugin

rest-rdma is a tool designed for sending requests over InfiniBand to the UFM server. These REST requests can fall into three categories:

1. UFM REST API requests
2. ibdiagnet requests

3. Telemetry requests

The rest-rdma utility is distributed as a Docker container, capable of functioning both as a server and a client.

Deployment Server

Deploy Plugin on UFM Appliance

1. Log into your UFM as admin.
2. Enter config mode. Run:

```
enable
config terminal
```

> **Note**
>
> Make sure that UFM is running with `show ufm status`. If UFM is down, then run with `ufm start`.

3. Ensure that rest-rdma plugin is disabled with the `show ufm plugin` command.
4. Pull the plugin container with `docker pull mellanox/ufm-plugin-rest-rdma:[version].`
5. Run `ufm plugin rest-rdma add tag [version]` to enable the plugin.
6. Check that plugin is up and running with `docker pull mellanox/ufm-plugin-rest-rdma:[version]`

Deploy Plugin on Bare Metal Server

1. Verify that UFM is installed and running.
2. Pull image from docker hub:

   `docker pull mellanox/ufm-plugin-rest-rdma:[version]`
3. To load image run:

   `/opt/ufm/scripts/manage_ufm_plugins.py add -p rest-rdma`

Deployment Client

Run the following command to pull the image from the docker hub:

```
docker pull mellanox/ufm-plugin-rest-rdma:[version]
```

Verify that the `/tmp/ibdiagnet` directory exists on the client's computer. If not – create it.

To start container as client (on any host in the same fabric as UFM server) run:

```
docker run -d --network=host --privileged --name=ufm-plugin-rest-rdma --rm -v /tmp/ibdiagnet:/tmp/ibdiagnet mellanox/ufm-plugin-rest-rdma:[version] client
```

To check that plugin is up and running, run:

```
docker ps
```

How to Run

Server

In server mode `ufm_rdma.py` is started automatically and is restarted if exited. If the `ufm_rdma.py` server is not running – enter to the docker and run the following commands to start the server:

```
cd /opt/ufm/src/ufm-plugin-ufm-rest
./ufm_rdma.py -r server
```

Client

There are three options to run client. Running the client from inside the Docker container, using a custom script from the hosting server to execute the client or using the "docker exec" command from the hosting server.

1. **Option 1:** Run the client from inside the Docker container
    1. Enter the docker container using `docker exec -it ufm-plugin-rest-rdma bash`
    2. Then, run `cd /opt/ufm/src/ufm-plugin-rest-rdma`
    3. Use the `-h` help option to see the available parameters

       `./ufm_rdma.py -h`
2. **Option 2:** From the host server, the scripts can be located at `/opt/ufm/ufm-plugin-ufm-rest/ directory` inside the docker container. They can copied using the following command:

> **Note**
>
> cp <containerId>:/opt/ufm/ufm-plugin-ufm-rest/[script name] /host/path/target

Example:

> **Note**
>
> cp <containerId>:/opt/ufm/ufm-plugin-ufm-rest/ufm-rest-rdma_client.sh /host/path/target

1. To see the available options, run:

   `./ufm-rest-rdma_client.sh -h`

3. **Option 3:** From hosting server, use the `docker exec` command.

> **Note**
>
> To run from inside docker, run:
>
> `docker exec ufm-plugin-rest-rdma` prior to the command.
>
> For example:
>
> ```
> docker exec ufm-plugin-rest-rdma /opt/ufm/ufm-plugin-ufm-rest/src/ufm_rdma.py -r client -u admin -p password -t simple -a GET -w ufmRest/app/ufm_version
> ```

Authentication Configuration

Telemetry and ibdiagnet request authentication options could be enabled or disabled (enabled by default – set to True) in `ufm_rdma.ini file` in [Server] section on the server. The `rest_rdma` server performs simple requests to UFM server, using supplied credentials to verify that the user is allowed to run telemetry or ibdiagnet requests.

```
[Server]
use_ufm_authentication=True
```

Remote ibdiagnet Request

The following two user scripts can run on the hosting server.

- remote_ibdiagnet_auth.sh
- remote_ibdiagnet.sh

These scripts should be copied from the container to the hosting server using the following command:

```
cp <containerId>:/opt/ufm/ufm-plugin-ufm-rest/[script name]
/host/path/target
```

Example :

```
cp <containerId>:/opt/ufm/ufm-plugin-ufm-
rest/remote_ibdiagnet_auth.sh /host/path/target
```

The `remote_ibdiagnet.sh` script does not require authentication as the server side can run on a machine which does not run UFM (which is responsible for the authentication). This means it can run from the hosting server.

```
/remote_ibdiagnet.sh [options]
```

Authenticated Remote ibdiagnet Request

The `remote_ibdiagnet_auth.sh` script can receive parameters as credentials for authentication with UFM server.

```
/remote_ibdiagnet_auth.sh [options]
```

To get all the options, run the following command:

```
/remote_ibdiagnet_auth.sh -h
```

Note

Important Note:

When using remote_ibdiagnet.sh, authentication is not required and the the ibdiagnet parameters should be sent in ibdiagnet format.

Example: `./remote_ibdiagnet.sh --get_phy_info`

When using the remote_ibdiagnet**_auth**.sh, the ibdiagnet parameters should be sent using the `-l` key.

Example without credentials:
`./remote_ibdiagnet_auth.sh -l '--get_phy_info'`

Example with credentials:
`./remote_ibdiagnet_auth.sh -u username -p password -l '–get_phy_info'`

Please use the -h option to see the examples of credential usage.

Rest Request with Username/Password Authentication

To get the UFM version from inside the docker:

```
./ufm_rdma.py -r client -u admin -p admin_pwd -t simple -a GET -w
ufmRest/app/ufm_version
```

To get the UFM version from hosting server using script:

```
./ufm_rest_rdma_client.sh -u admin -p admin_pwd -t simple -a GET
-w ufmRest/app/ufm_version
```

For telemetry:

```
./ufm_rdma.py -r client -u admin -p admin_pwd -t telemetry -a GET
-g 9001 -w /csv/enterprise
```

To get ibdiagnet run result using UFM REST API from inside the docker:

```
./ufm_rdma.py -r client -u admin -p admin_pwd -t ibdiagnet -a
POST -w ufmRest/reports/ibdiagnetPeriodic -l '{"general": {"name":
"IBDiagnet_CMD_1234567890_199_88", "location": "local", "running_mode": "once"}, "command_flags": {"--
pc": ""}}'
```

Rest Request with Client Certificate Authentication

```
need to pass path to client certificate file and name of UFM
server machine:
6.  ./ufm_rdma.py -r client -t simple -a GET -w
ufmRest/resources/modules -d /path/to/certificate/file/ufm-
client.pfx -s ufm.azurehpc.core.azure-test.net
for telemetry if need authentication from inside the docker
./ufm_rdma.py -r client  -t telemetry -a GET -g 9001 -w
csv/enterprise -d /path/to/certificate/file/ufm-client.pfx -s
ufm.azurehpc.core.azure-test.net
```

Note

Client certificate file should be located INSIDE the docker container.

Rest Request with Token Authentication

```
need to pass token for authentication
./ufm_rdma.py -r client -k OGUY7TwLvTmFkXyTkcsEWD9KKNvq6f -t
simple -a GET -w ufmRestV3/app/ufm_version
for telemetry if need to perform authentication
./ufm_rdma.py -r client -k 4rQRf7i7wEeliuJEurGbeecc210V6G  -t
telemetry -a GET -g 9001 -w /csv/enterprise
```

Note

Token could be generated using UFM UI.

Note

If a token is used for client authentication, `ufmRestV3` must be used.

NDT Plugin

Overview

NDT plugin is a self-contained Docker container with REST API support managed by UFM. The NDT plugin introduces the following capabilities:

1.

    1. **NDT topology comparison:** Allows the user to compare InfiniBand fabric managed by the UFM and NDT files which are used for the description of InfiniBand clusters network topology.

        - Verifies the IB fabric connectivity during cluster bring-up.
        - Verifies the specific parts of IB fabric after component replacements.
        - Automatically detects any changes in topology.

    2. **Subnet Merger - Expansion of the fabric based on NDT topology files**

        Allows users to gradually extend the InfiniBand fabric without causing any disruption to the running fabric. The system administrator should prepare the

NDT topology files, which describe the InfiniBand fabric extensions. Then, an intuitive and user-friendly UI wizard facilitates the topology extension process with a step-by-step guidance for performing necessary actions.

- The Subnet Merger tool verifies the fabric topology within a predefined NDT file, and reports issues encountered for immediate resolution.
- Once the verification results are acceptable by the network administrator, the tool creates a topoconfig file to serve as input for OpenSM. This allows setting the physical port states of the designated boundary ports as desired (physical ports can be set as disabled or no-discover).
- Once the topoconfig file is deployed, the IB network can be extended and verified for the next IB extension.

Deployment

The following are the possible ways NDT plugin can be deployed:

1. On UFM Appliance
2. On UFM Software

For detailed instructions on how to deploy the NDT plugin refer to this page.

Authentication

Following authentication types are supported:

- basic (/ufmRest)
- client (/ufmRestV2**)**
- token (/ufmRestV3)

REST API

The following REST APIs are supported:

Topodiff

- GET /help

- GET /version
- POST /upload_metadata
- GET /list
- POST /compare
- POST /cancel
- GET /reports
- GET /reports/<report_id>
- POST /delete

Subnet Merger

- GET /merger_ndts_list
- GET /merger_ndts_list/<ndt_file_name>
- POST /merger_upload_ndt
- POST /merger_verify_ndt
- GET /merger_verify_ndt_reports
- GET /merger_verify_ndt_reports/<report_id>
- POST /merger_update_topoconfig
- POST /merger_deploy_ndt_config
- POST /merger_update_deploy_ndt_config
- POST /merger_delete_ndt
- GET /merger_deployed_ndt
- POST /merger_create_topoconfig

For detailed information on how to interact with NDT plugin, refer to the NVIDIA UFM Enterprise > Rest API > NDT Plugin REST API.

NDT Format – Topodiff

NDT is a CSV file containing data relevant to the IB fabric connectivity. The NDT plugin extracts the IB connectivity data based on the following fields:

1. Start device
2. Start port
3. End device
4. End port
5. Link type

Switch to Switch NDT

By default, IB links are filtered by:

- Link Type is Data
- Start Device and End Device end with IBn, where n is a numeric value.

For TOR switches, Start port/End port field should be in the format **Port N**, where **N** is a numeric value.

For Director switches, Start port/End port should be in the format **Blade N_Port i/j**, where **N** is a leaf number, **i** is an internal ASIC number and **j** is a port number.

Examples:

Start Device	Start Port	End Device	End Port	Link Type
DSM07-0101-0702-01IB0	Port 21	DSM07-0101-0702-01IB1	Blade 2_Port 1/1	Data
DSM07-0101-0702-01IB0	Port 22	DSM07-0101-0702-01IB1	Blade 2_Port 1/1	Data

Start Device	Start Port	End Device	End Port	Link Type
DSM07-0101-0702-01IB0	Port 23	DSM07-0101-0702-02IB1	Blade 3_Port 1/1	Data
DSM09-0101-0617-001IB2	Port 33	DSM09-0101-0721-001IB4	Port 1	Data
DSM09-0101-0617-001IB2	Port 34	DSM09-0101-0721-001IB4	Port 2	Data
DSM09-0101-0617-001IB2	Port 35	DSM09-0101-0721-001IB4	Port 3	Data

Switch to Host NDT

NDT is a CSV file containing data not only relevant to the IB connectivity.

Extracting the IB connectivity data is based on the following five fields:

1. Start device
2. Start port
3. End device
4. End port
5. Link type

IB links should be filtered by the following:

- Link type is "Data".
- "Start Device" or "End Device" end with **IBN**, where **N** is a numeric value.
    - The other Port should be based on persistent naming convention: **ibpXsYfZ**, where **X**, **Y** and **Z** are numeric values.

For TOR switches, Start port/End port field will be in the format Port n, where n is a numeric value.

For Director switches, Start port/End port will be in the format **Blade N_Port i/j**, where **N** is a leaf number, **i** is an internal ASIC number and **j** is a port number.

Examples:

Start Device	Start Port	End Device	End Port	Link Type
DSM071081704019	DSM071081704019 ibp11s0f0	DSM07-0101-0514-01IB0	Port 1	Data
DSM071081704019	DSM071081704019 ibp21s0f0	DSM07-0101-0514-01IB0	Port 2	Data
DSM071081704019	DSM071081704019 ibp75s0f0	DSM07-0101-0514-01IB0	Port 3	Data

Other

Comparison results are forwarded to syslog as events. Example of `/var/log/messages` content:

1. Dec 9 12:32:31 <server_ip> ad158f423225[4585]: NDT: missing in UFM "SAT111090310019/SAT111090310019 ibp203s0f0 - SAT11-0101-0903-19IB0/15"
2. Dec 9 12:32:31 <server_ip> ad158f423225[4585]: NDT: missing in UFM "SAT11-0101-0903-09IB0/27 - SAT11-0101-0905-01IB1-A/Blade 12_Port 1/9"
3. Dec 9 12:32:31 <server_ip> ad158f423225[4585]: NDT: missing in UFM "SAT11-0101-0901-13IB0/23 - SAT11-0101-0903-01IB1-A/Blade 08_Port 2/13"

For detailed information about how to check syslog, please refer to the NVIDIA UFM-SDN Appliance Command Reference Guide > UFM Commands > UFM Logs.

Minimal interval value for periodic comparison in five minutes.

In case of an error the clarification will be provided.

For example, the request "`POST /compare`" without NDTs uploaded will return the following:

- URL: https://<server_ip>/ufmRest/plugin/ndt/compare
- response code: 400
- Response:

```
{
  "error": [
    "No NDTs were uploaded for comparison"
  ]
}
```

Configurations could be found in "`ufm/conf/ndt.conf`"

- Log level (default: INFO)
- Log size (default: 10240000)
- Log file backup count (default: 5)
- Reports number to save (default: 10)
- NDT format check (default: enabled)
- Switch to switch and host to switch patterns (default: see NDT format section)

For detailed information on how to export or import the configuration, refer to the NVIDIA UFM-SDN Appliance Command Reference Guide > UFM Commands > UFM Configuration Management.

Logs could be found in "`ufm/logs/ndt.log`".

For detailed information on how to generate a debug dump, refer to the NVIDIA UFM-SDN Appliance Command Reference Guide > System Management > Configuration Management > File System.

NDT Format – Subnet Merger

The Subnet Merger tool facilitates the seamless expansion of the InfiniBand fabric based on Non-Disruptive Topology (NDT) files. This section outlines the process of extending the fabric while ensuring uninterrupted operation. The tool operates through an intuitive UI wizard, guiding users step-by-step in extending the fabric topology.

The Subnet Merger tool enables the gradual expansion of the InfiniBand fabric without causing disruptions to the existing network. To achieve this, system administrators need to prepare NDT topology files that describe the planned fabric extensions. The tool offers an intuitive UI wizard that simplifies the extension process.

Functionality

1. **NDT Topology File Verification:** The Subnet Merger tool verifies the InfiniBand fabric topology specified in a predefined NDT file. During this verification, any issues encountered are reported to the user for immediate resolution. This step ensures the integrity of the planned fabric extension.

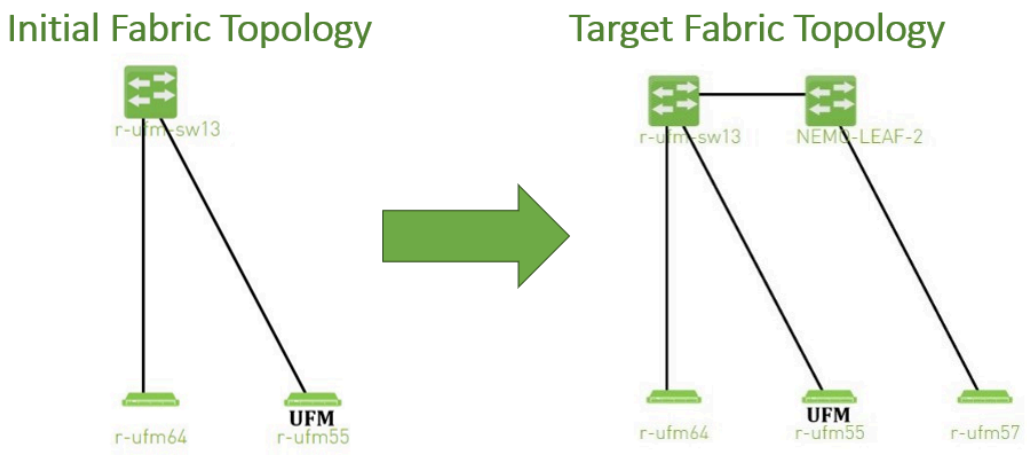
1. **Topology Extension Preparation:** Upon successful verification of the NDT topology file, the tool generates a comprehensive verification report. The network administrator reviews this report and ensures its acceptability.

1. **Topoconfig File Generation:** After obtaining acceptable verification results, the tool generates a topoconfig file. This file serves as input for OpenSM, the Subnet Manager for InfiniBand fabrics. The topoconfig file allows the network administrator to define the desired physical port states for designated boundary ports. These states include "disabled" or "no-discover."

1. **Fabric Extension and Verification:** With the topoconfig file prepared, the Subnet Merger tool initiates the deployment of the extended fabric configuration. The tool ensures that the defined physical port states are implemented. Once the extension is in place, the IB network can be extended further as needed. The fabric extension is executed while maintaining the operational stability of the existing network.

1. **Conclusion:** The Subnet Merger tool offers a reliable and user-friendly solution for expanding InfiniBand fabrics using NDT topology files. By following the steps provided in the intuitive UI wizard, system administrators can seamlessly extend the fabric while adhering to predefined physical port states. This tool ensures the smooth operation of the fabric throughout the expansion process, eliminating disruptions and enhancing network scalability.

Subnet Merger Flow



1. Create NDT, file that describes initial topology with definition of boundary ports. Boundary ports – switch ports that will be used for fabric extension. In our case it will be r-ufm-sw13 switch ports number 1 and 3. In NDT file those ports should be defined as boundary and disabled:

```
rack #,U
height,#Fields:StartDevice,StartPort,StartDeviceLocation,EndDe
height_1,LinkType,Speed,_2,Cable
Length,_3,_4,_5,_6,_7,State,Domain
,,MF0;r-ufm-sw13:MQM8700/U1,Port
1,,,,,,,,,,,,,,,Disabled,Boundary
,,MF0;r-ufm-sw13:MQM8700/U1,Port 30,,r-ufm55 mlx5_1,Port
1,,,,,,,,,,,,Active,In-Scope
,,MF0;r-ufm-sw13:MQM8700/U1,Port 29,,r-ufm55 mlx5_0,Port
1,,,,,,,,,,,,Active,In-Scope
,,MF0;r-ufm-sw13:MQM8700/U1,Port 26,,r-ufm64 mlx5_0,Port
1,,,,,,,,,,,,Active,In-Scope
,,MF0;r-ufm-sw13:MQM8700/U1,Port
3,,,,,,,,,,,,,,,Disabled,Boundary
```

2. Upload a new NDT topology file which describes the desired topology. Before deploying to UFM, the new NDT topology file should be verified against the existing topology – to find out mismatches and problems.

   After the verification, the plugin generates reports including information about:

-
    - Duplicated GUIDs
    - Misswired links
    - Non-existent links in the pre-defined NDT files

1.
    - Links that exist in the fabric and not in the NDT file

2. Following the issues detected in the plugin reports, the network administrator changes the NDT file or the fabric. The verification process can be repeated as many times as necessary until the network administrator is satisfied with the results.

3. If the NDT verification results are satisfactory, a topoconfig file is generated and can be deployed to the UFM server to be used as configuration input for OpenSM. Topoconfig file should be located at /opt/ufm/files/conf/opensm/topoconfig.cfg on UFM server. By sending SIGHUP signal to opensm it forced to read configuration and to deploy it. In topoconfig file at this stage boundary ports will be defined as **Disabled**.

   Example of topoconfig.cfg:

```
0xb83fd2030080302e,1,-,-,Any, Disabled
0xb83fd2030080302e,30,0xf452140300280081,1,Any,Active
0xb83fd2030080302e,29,0xf452140300280080,1,Any,Active
0xb83fd2030080302e,26,0xf452140300280040,1,Any,Active
0xb83fd2030080302e,3,-,-,Any, Disabled
```

4. Next stage is to extend the fabric. Prepare separately new subnet that will be added to the existing fabric and, once it is ready, connect to the boundary ports, that are defined as Disabled in configuration file, so newly added subnet will not be discovered by opensm and will not affect in any way current setup functionality.

5. Once new subnet connected to the fabric - prepare next NDT file, that contains setup, that describes current fabric with extended, when previously defined as boundary ports defined as Active and if planned to continue with extension new ports defined as boundary.

For example port number 9 of switch r-ufm-sw13:

```
rack #,U
height,#Fields:StartDevice,StartPort,StartDeviceLocation,EndDe
height_1,LinkType,Speed,_2,Cable
Length,_3,_4,_5,_6,_7,State,Domain
,,MF0;r-ufm-sw13:MQM8700/U1,Port 1,,NEMO-LEAF-2,Port
1,,,,,,,,,,,Active,In-Scope
,,MF0;r-ufm-sw13:MQM8700/U1,Port 30,,r-ufm55 mlx5_1,Port
1,,,,,,,,,,,Active,In-Scope
,,MF0;r-ufm-sw13:MQM8700/U1,Port 29,,r-ufm55 mlx5_0,Port
1,,,,,,,,,,,Active,In-Scope
,,NEMO-LEAF-2,Port 11,,r-ufm57 mlx5_0,Port
1,,,,,,,,,,,Active,In-Scope
,,MF0;r-ufm-sw13:MQM8700/U1,Port 26,,r-ufm64 mlx5_0,Port
1,,,,,,,,,,,Active,In-Scope
,,NEMO-LEAF-2,Port 1,,MF0;r-ufm-sw13,Port
1,,,,,,,,,,,Active,In-Scope
,,MF0;r-ufm-sw13:MQM8700/U1,Port 3,,NEMO-LEAF-2,Port
3,,,,,,,,,,,Active,In-Scope
,,NEMO-LEAF-2,Port 3,,MF0;r-ufm-sw13,Port
3,,,,,,,,,,,Active,In-Scope
,,MF0;r-ufm-sw13:MQM8700/U1,Port
9,,,,,,,,,,,,,Disabled,Boundary
```

6. After new subnet connected physically to the fabric, in opensm configuration file (topoconfig.cfg) boundary ports previously defined as Disabled should be set as No-discover. Example:

```
0xb83fd2030080302e,1,-,-,Any,No-discover
0xb83fd2030080302e,30,0xf452140300280081,1,Any,Active
0xb83fd2030080302e,29,0xf452140300280080,1,Any,Active
0xb83fd2030080302e,26,0xf452140300280040,1,Any,Active
0xb83fd2030080302e,3,-,-,Any,No-discover
```

7. Updated file should be deployed to UFM. In case boundary ports will be defined as No-discover – fabric, connected beyond those ports will not be discovered by opensm, but all the ibutils (ibdiagnet…) could send mads beyond those ports to newly added subnet - so NDT file verification for extended setup could be performed.

8. Upload new NDT file and run verification for this file. Fix problems detected by verification. Once satisfied with results – deploy configuration to UFM.

   Example of topoconfig file for extended setup:

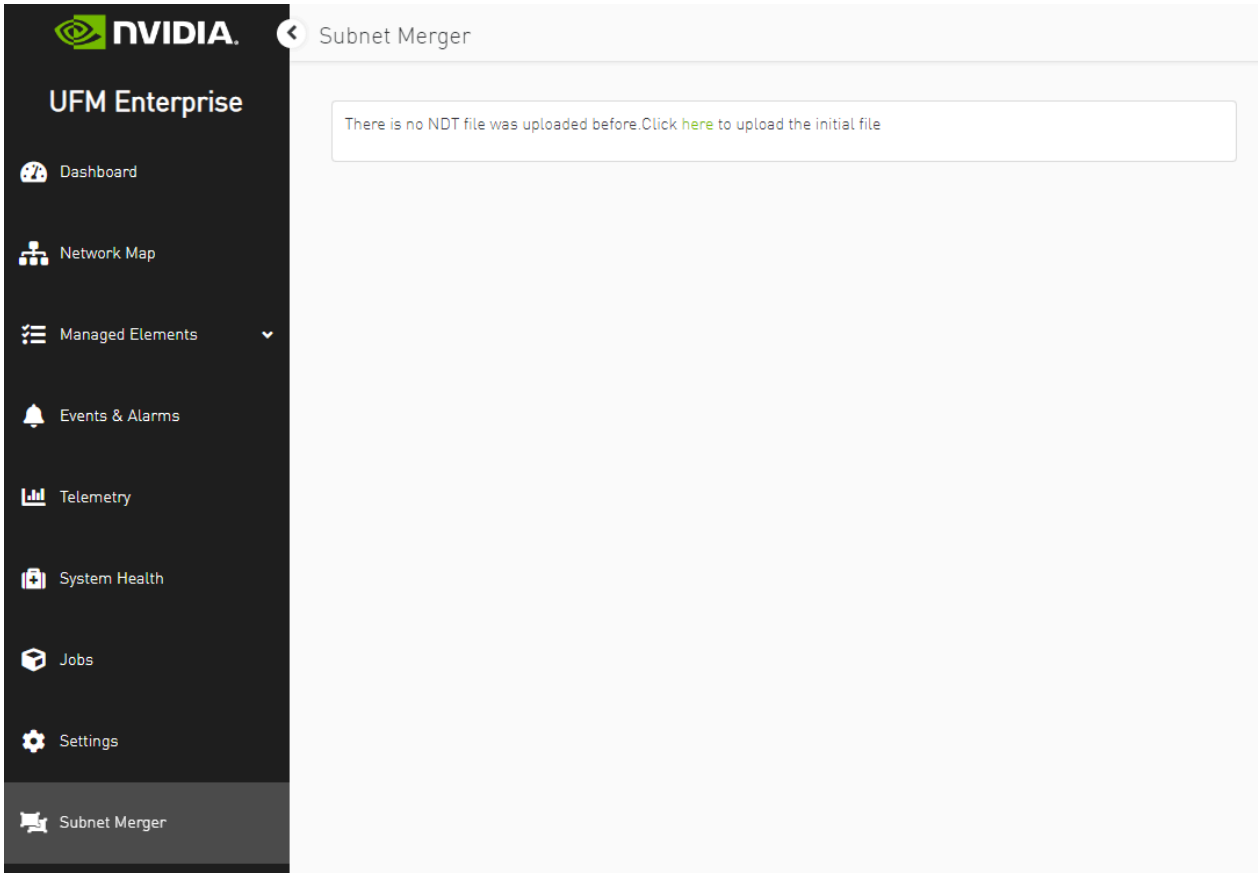
```
0xb83fd2030080302e,1,0x98039b0300867bba,1,Any,Active
0xb83fd2030080302e,30,0xf452140300280081,1,Any,Active
0xb83fd2030080302e,29,0xf452140300280080,1,Any,Active
0x98039b0300867bba,11,0x248a0703009c0066,1,Any,Active
0xb83fd2030080302e,26,0xf452140300280040,1,Any,Active
0x98039b0300867bba,1,0xb83fd2030080302e,1,Any,Active
0xb83fd2030080302e,3,0x98039b0300867bba,3,Any,Active
0x98039b0300867bba,3,0xb83fd2030080302e,3,Any,Active
0xb83fd2030080302e,9,-,-,Any,Disabled
```

9. Repeat previous steps if need to perform additional setup extension.

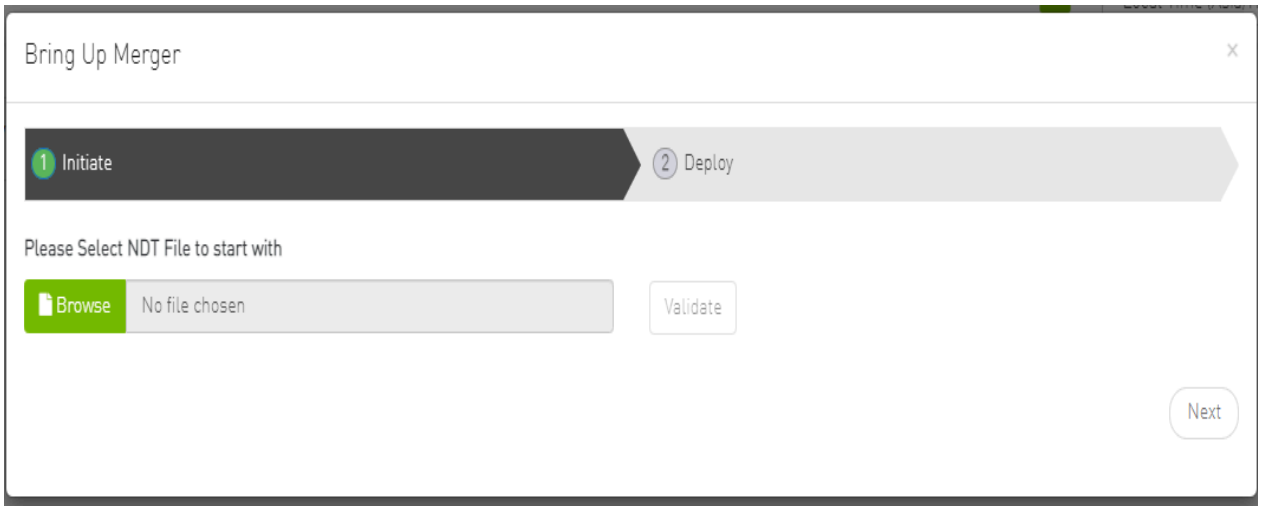
Subnet Merger UI

Bring-Up Merger Wizard

1. Add the NDT plugin to UFM by loading the plugin's image through Settings->Plugins Management. A new item will appear in the main left navigator menu of the UFM labeled "Subnet Merger".

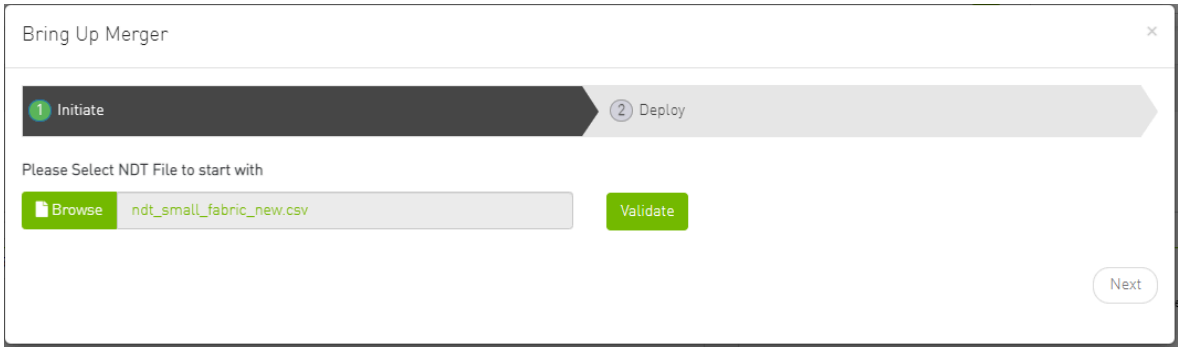


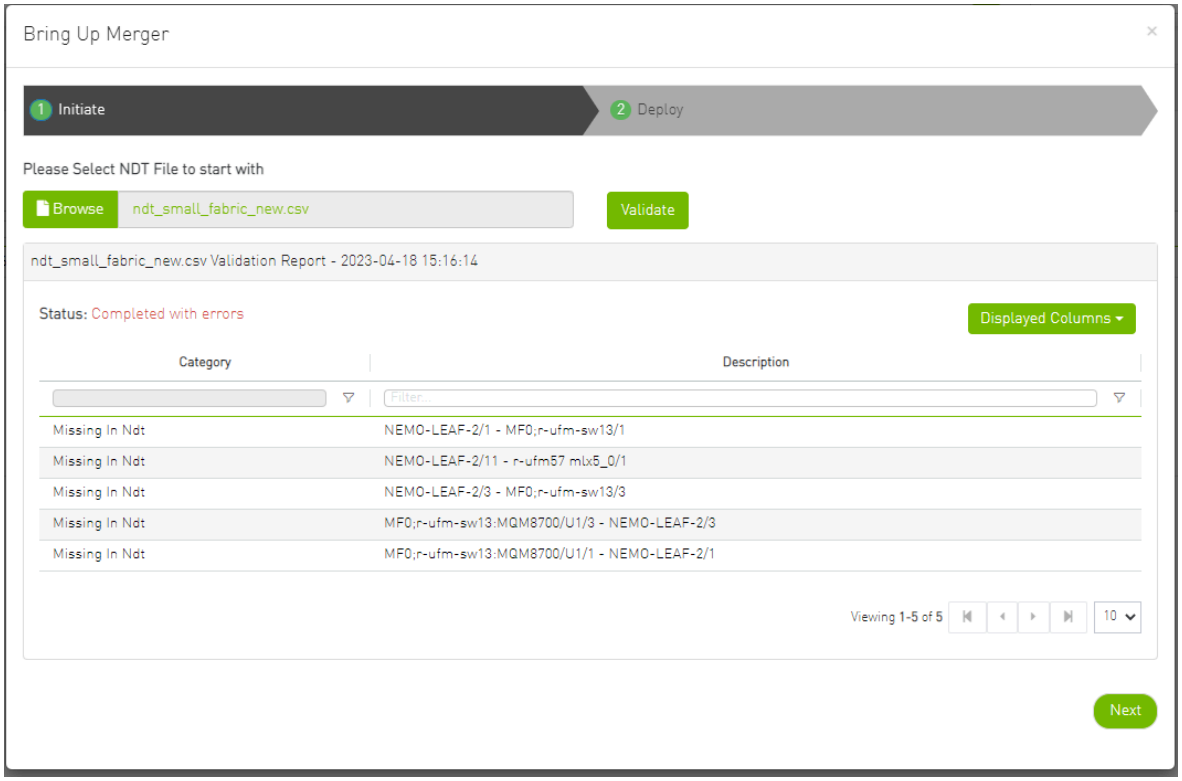
2. Access "Subnet Merger" to initiate the bring-up wizard.



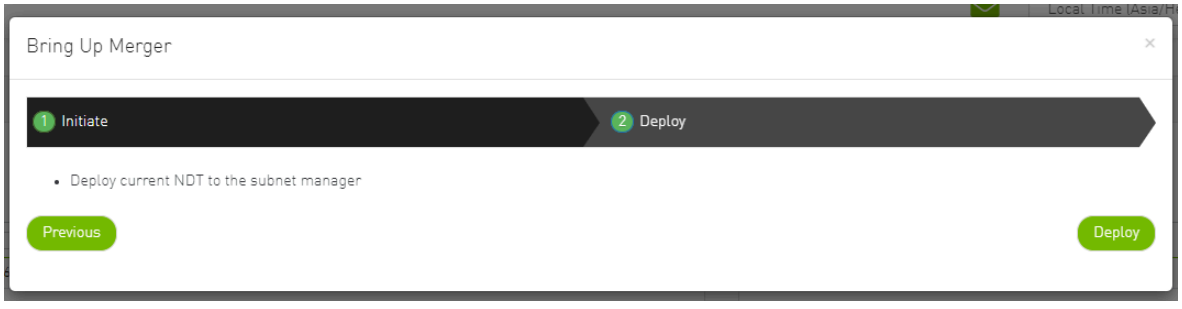
3. The wizard will guide you through the process, containing the following steps:

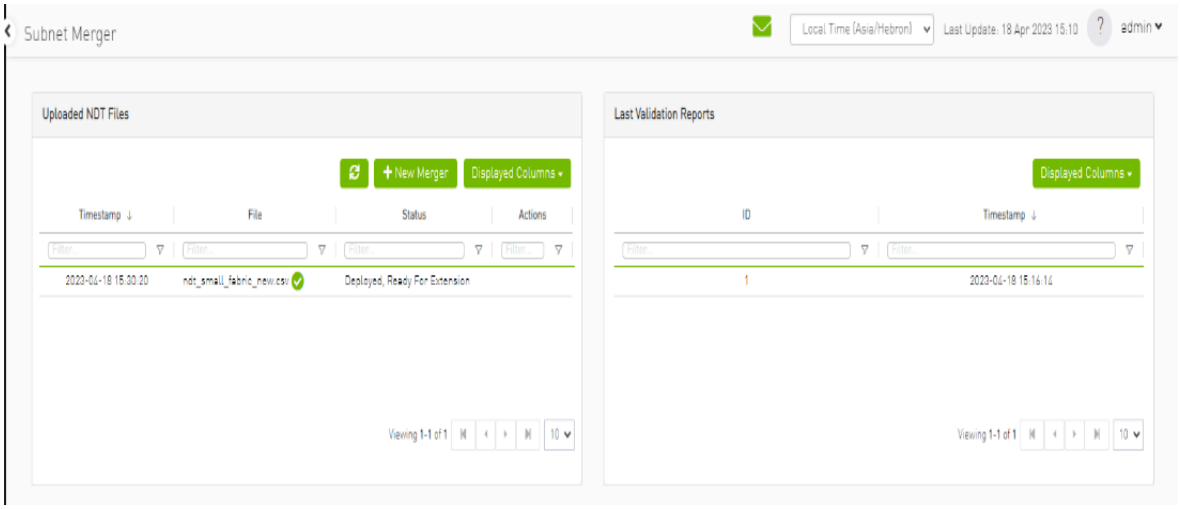
    1. Upload the initial NDT tab and validate it.





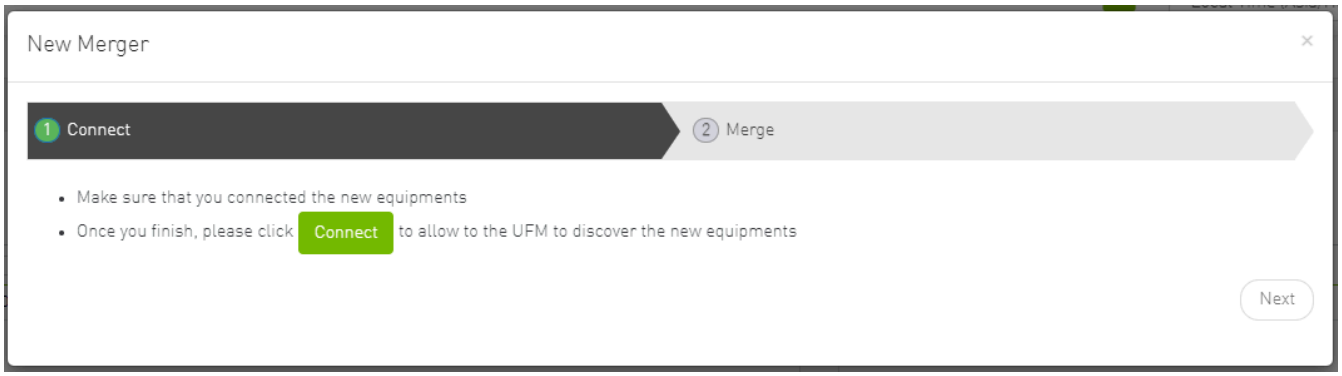
2. Once you are satisfied with the results of the validation in the previous tab, you can proceed to deploy the file.



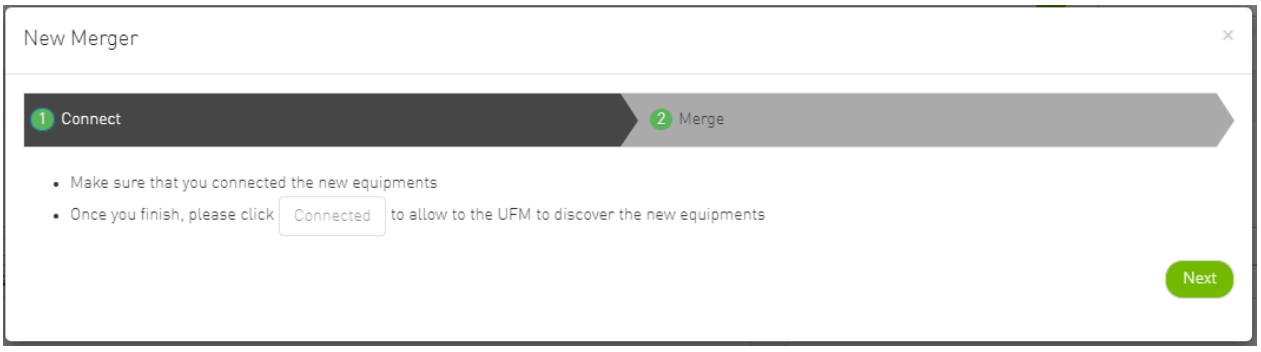


New Subnet Merger

Once you have successfully deployed the initial NDT file, you can initiate a new merger process by clicking the "New Merger" button.

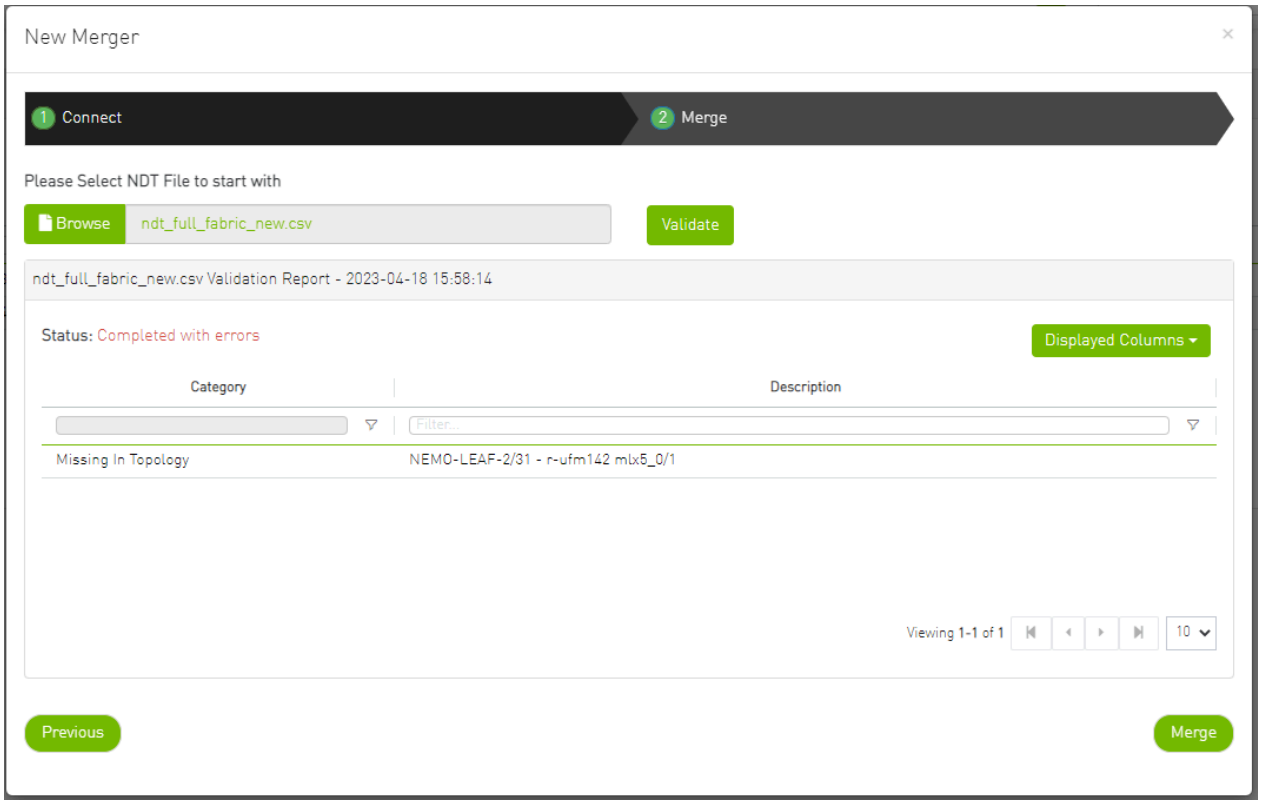


1. "Connect" Tab, it is important to physically connect the new equipment and confirm the connection. Then, click on a button which will open the boundary ports, change their state from Disabled to No-discover, and then deploy the active file again.

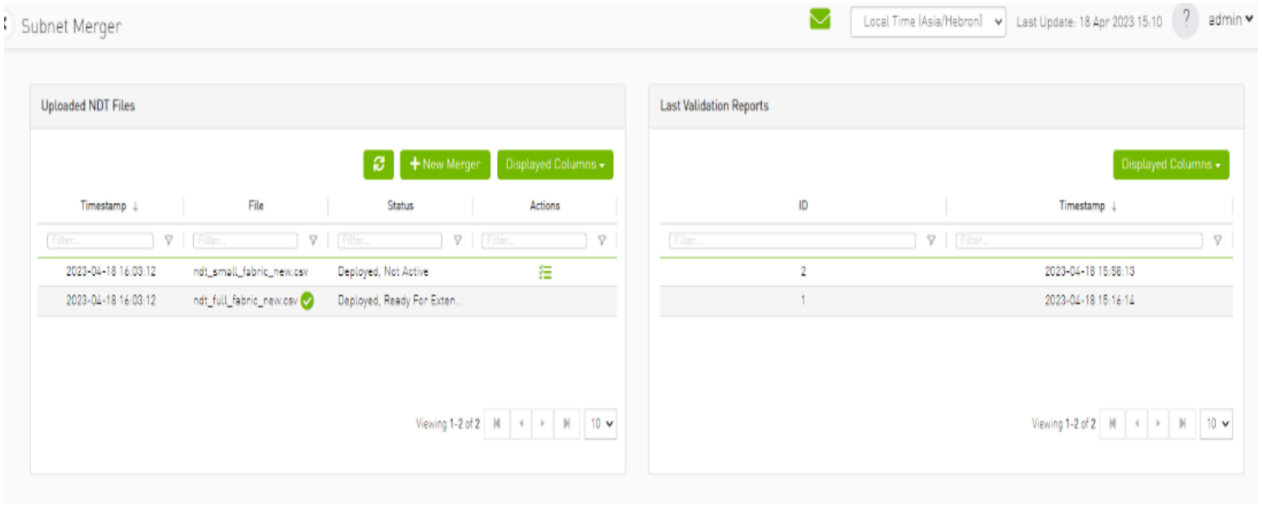


2. "Merge" Tab: Once the new equipment is connected and the boundary ports are updated, upload a new NDT file that includes both the current and newly added

equipment, along with their boundary ports for future merges. Please note that you cannot merge the file if there are duplicate GUIDs in the report's results.



3. After completing the merge wizard, and if necessary, you can further proceed to extend the IB fabric.



Extending the InfiniBand Setup via Subnet Merger

The following instructions outline the necessary steps for expanding the InfiniBand setup or fabric using subnet merging.

1. **Step 1: NDT File Upload (Repeatable)**

   Upload the NDT file, performing this action as many times as required, especially when addressing file-related issues.

2. **Step 2: NDT File Validation and Verification (Repeatable)**

   Validate the NDT file, a process that can be repeated multiple times, particularly after fixing fabric topology or NDT file errors. After initiating this call, you will obtain a validation report ID. The progress of this process is asynchronous, with the report's status initially indicated as "running." Once the report is completed, the status will change to either "Successfully completed" or "Completed with errors."

3. **Step 3: Retrieving and Monitoring the Validation Report**

   Retrieve the validation report by its corresponding ID, running this step through continuous polling until the report reaches completion.

4. **Step 4: Review and Potential Fixes**

   Inspect the report and address any necessary fixes to either the NDT file or the topology. Should changes be made to the file, upload the corrected NDT file anew. Alternatively, in case of topology has changed, repeat the verification process.

5. **Step 5: Topology Deployment to UFM**

   Deploy the verified topology to UFM once you are satisfied with the verification outcomes.

6. **Step 6: Adjusting Boundary Ports and Deployment**

   Following the physical connection of the setup extension, change the boundary ports' state from "Disabled" to "No-discover."

7. **Step 7: Uploading Updated Topoconfig File**

   Deploy the updated topoconfig file to the UFM server.

8. **Step 8: Next NDT File Upload (Combined Fabric and Extension)**

   Upload the next NDT file, which consolidates the current fabric and extension components.

9. **Step 9: NDT File Verification**

   Conduct the NDT file verification process.

10. **Step 10: Reviewing Verification Report**

    Review the verification report.

11. **Step 11: Addressing Setup or NDT File Issues**

    If necessary, make necessary adjustments to the setup or NDT file.

12. **Step 12: Final Configuration Deployment**

    Once content with the modifications, proceed to deploy the configuration to UFM.

13. **Step 13: Iterative Workflow**

    Repeat this flow as many times as needed to further the expansion process.

UFM Telemetry FluentD Streaming (TFS) Plugin

Overview

TFS plugin is a self-contained Docker container with REST API support managed by UFM. TFS plugin provides Telemetry counters streaming to FluentD capability. As a fabric manager, the UFM Telemetry holds real-time network telemetry information of the network topology. This information changes over time and is reflected to the telemetry console. In order to do so, we present a stream of the UFM Telemetry data to the FluentD plugin.

Deployment

The following are the possible ways the TFS plugin can be deployed:

1. On UFM Appliance

2. On UFM Software

For complete instructions on deploying the TFS plugin, refer to UFM Telemetry endpoint stream To Fluentd endpoint (TFS).

Authentication

The following authentication types are supported:

- basic (/ufmRest)
- client (/ufmRestV2)
- token (/ufmRestV3)

Rest API

The following REST APIs are supported:

- POST /plugin/tfs/conf
- GET /plugin/tfs/conf
- POST /plugin/tfs/conf/attributes
- GET /plugin/tfs/conf/attributes

For detailed information on interacting with TFS plugin, refer to the NVIDIA UFM Enterprise > Rest API > TFS Plugin REST API.

UFM Events Fluent Streaming (EFS) Plugin

Overview

EFS plugin is a self-contained Docker container with REST API support managed by UFM. EFS plugin extracts the UFM events from UFM Syslog and streams them to a remote FluentD destination. It also has the option to duplicate current UFM Syslog messages and forward them to a remote Syslog destination. As a fabric manager, it will be useful to

collect the UFM Enterprise events/logs, stream them to the destination endpoint and monitor them.

Deployment

The following are the ways EFS plugin can be deployed:

1. On UFM Appliance
2. On UFM Software

For detailed instructions on how to deploy EFS plugin, refer to UFM Event Stream to FluentBit endpoint (EFS).

Authentication

The following authentication types are supported:

- basic (/ufmRest)
- client (/ufmRestV2**)**
- token (/ufmRestV3)

Rest API

The following REST APIs are supported:

- PUT /plugin/efs/conf
- GET /plugin/efs/conf

For detailed information on how to interact with EFS plugin, refer to the NVIDIA UFM Enterprise > Rest API > EFS Plugin REST API.

UFM Bright Cluster Integration Plugin

Overview

The Bright Cluster Integration plugin is a self-contained docker container managed by UFM and is managed by the REST APIs. It enables integrating data from Bright Cluster

Manager (BCM) into UFM, providing a more comprehensive network perspective. This integration improves network-centered Root Cause Analysis (RCA) tasks and enables better scoping of workload failure domains.

Deployment

The Bright Cluster Integration plugin can be deployed either on the UFM Appliance or on UFM Software.

For detailed instructions on Bright Cluster Integration plugin deployment, refer to UFM Bright Cluster Integration Plugin.

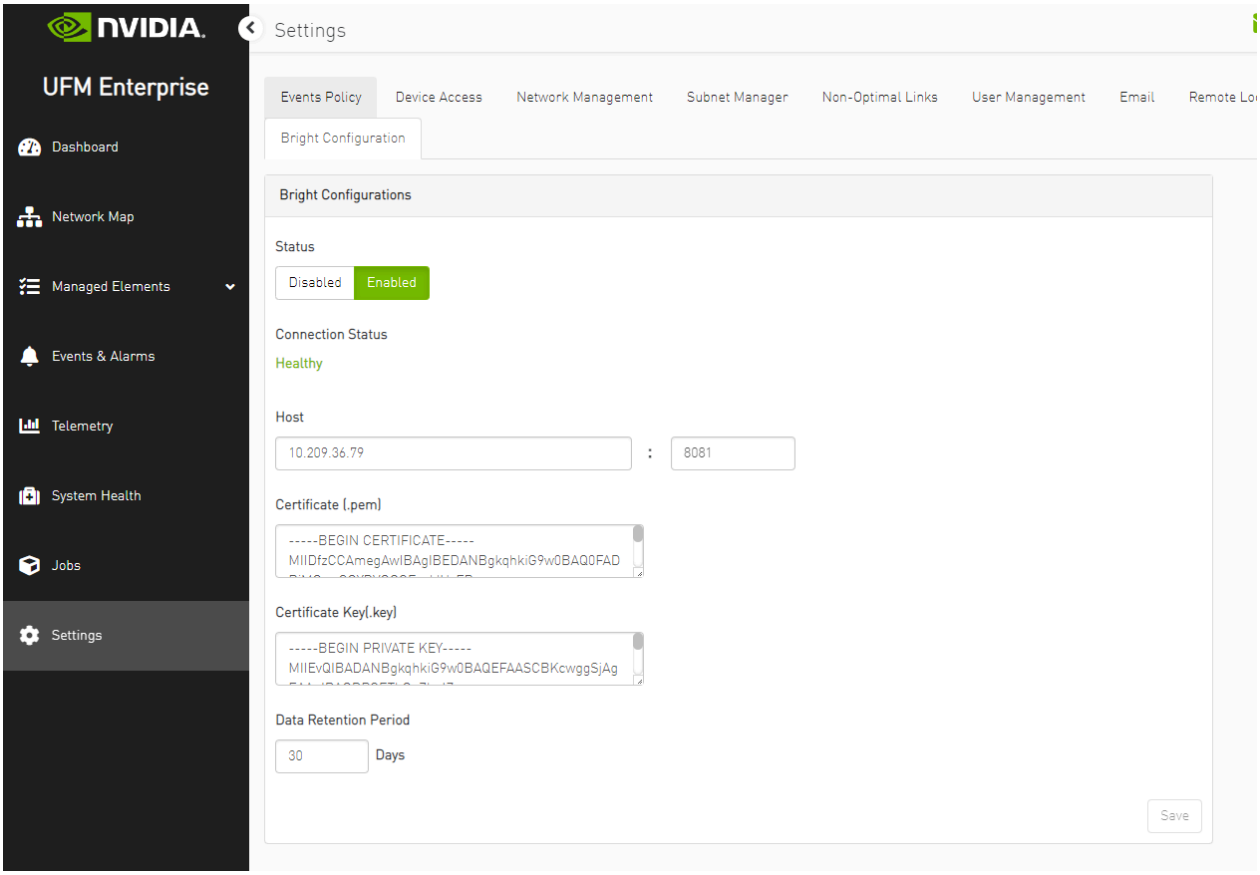
Authentication

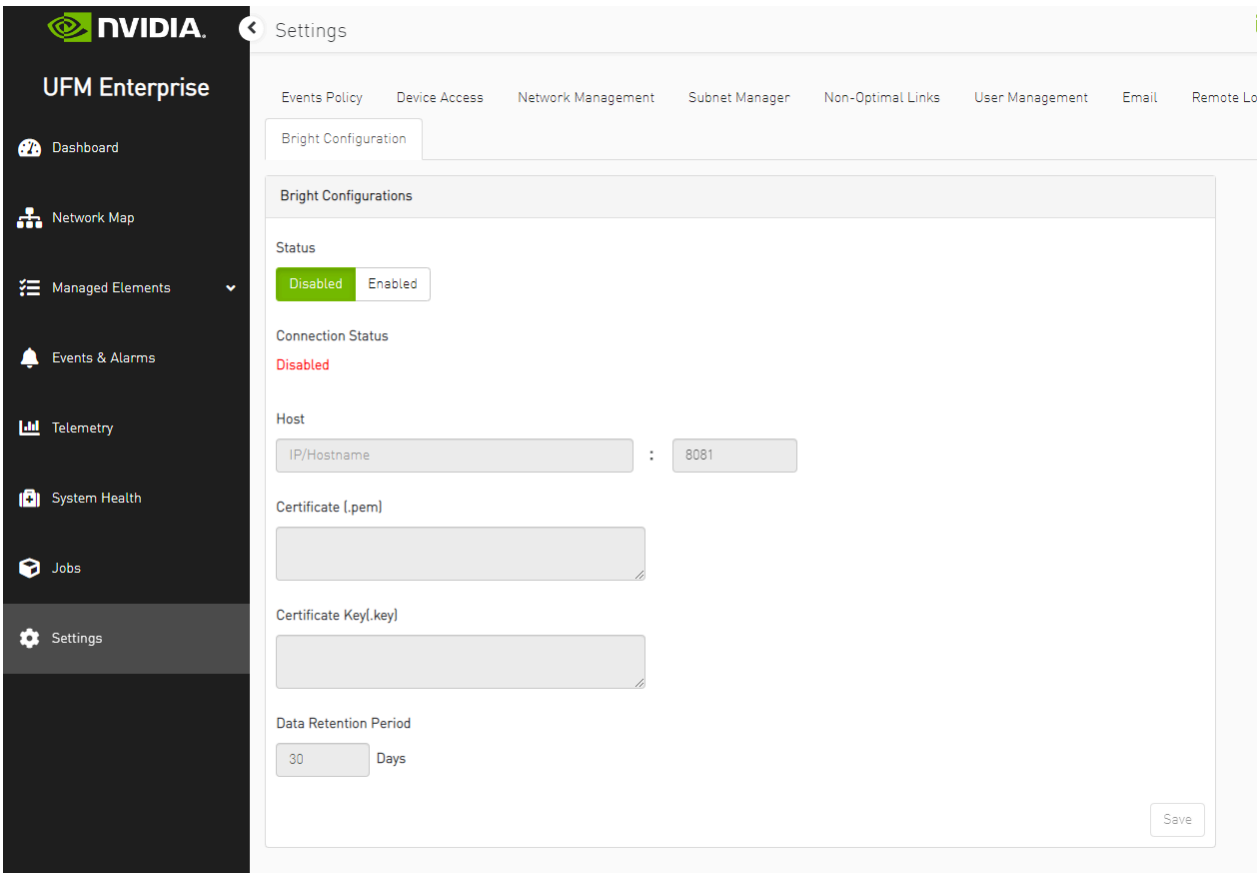
The following authentication types are supported:

- basic (/ufmRest)
- client (/ufmRestV2**)**
- token (/ufmRestV3)

GUI Screens

1. After the successful deployment of the plugin, a new tab is shown under the UFM settings section for bright configurations management:

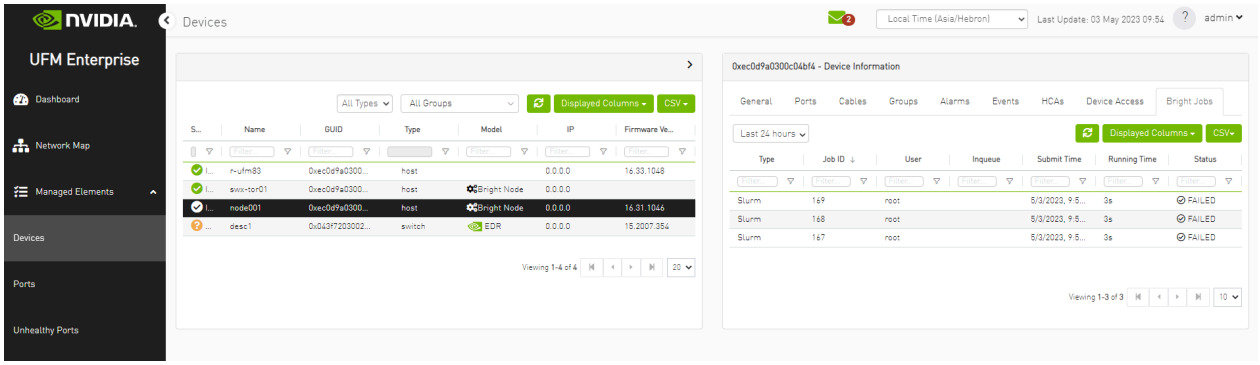

NVIDIA
UFM Enterprise
Dashboard
Network Map
Managed Elements
Events & Alarms
Telemetry
System Health
Jobs
Settings
Settings
Events Policy
Device Access
Network Management
Subnet Manager
Non-Optimal Links
User Management
Email
Bright Configuration
Bright Configurations
Status
Disabled
Enabled
Connection Status
Healthy
Host
10.209.36.79
8081
Certificate (.pem)
-----BEGIN CERTIFICATE-----
Certificate Key(.key)
-----BEGIN PRIVATE KEY-----
Data Retention Period
30
Days
Save


NVIDIA
UFM Enterprise
Dashboard
Network Map
Managed Elements
Events & Alarms
Telemetry
System Health
Jobs
Settings
Settings
Events Policy
Device Access
Network Management
Subnet Manager
Non-Optimal Links
User Management
Email
Bright Configuration
Bright Configurations
Status
Disabled
Enabled
Connection Status
Disabled
Host
IP/Hostname
8081
Certificate (.pem)
Certificate Key(.key)
Data Retention Period
30
Days
Save

Fill the below required configurations:

Parameter	Description
Host	Hostname or IP of the BCM server
Port	Port of the BCM server, is typically 8081
Certificate	BMC client certificate content that could be located in the BMC server machine under `.cm/XXX.pem`
Certificate key	BMC client certificate key that could be located in the BMC server machine under .`cm/XXX.key`
Data retention period	UFM erases the data gathered in the database after the configured retention period. By default, after 30 days.

2. After you ensure you have successfully completed the plugin configuration, and that you have established a healthy connection with the BMC, navigate to the UFM Web GUI -> Managed Elements -> Devices



Rest API

The following REST APIs are supported:

- PUT plugin/bright/conf
- GET plugin/bright/conf
- GET plugin/bright/data/nodes
- GET plugin/bright/data/jobs

For detailed information on how to interact with bright plugin APIs, refer to NVIDIA UFM Enterprise > Rest API > UFM Bright Cluster Integration Plugin REST API.

UFM Cyber-AI Plugin

Overview

The primary objective of this plugin is to integrate the UFM CyberAI product into the UFM Enterprise WEB GUI. This integration would result in both products being available within a single application.

Deployment

The following are the ways UFM CyberAI plugin can be deployed:

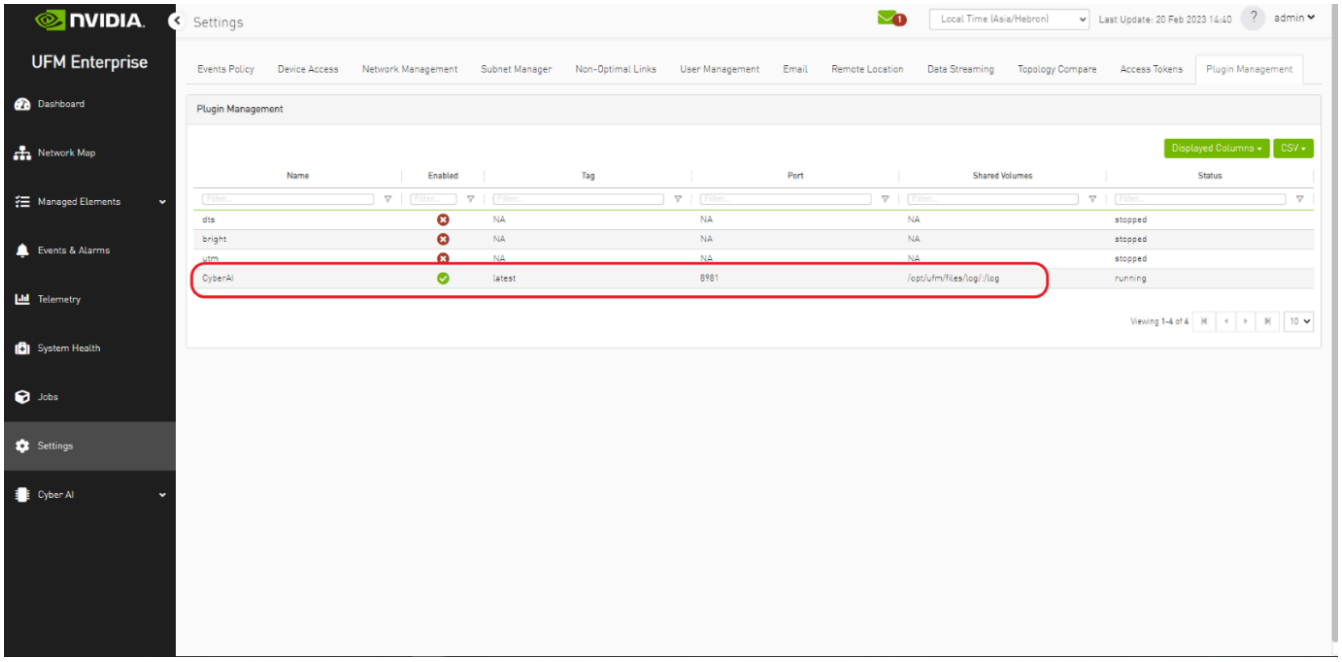
1. On UFM Appliance
2. On UFM Software

First, download the `ufm-plugin-cyberai-image` from the NVIDIA License Portal (NLP), then load the image on the UFM server, using the UFM GUI -> Settings -> Plugins Management tab or by loading the image via the following command:

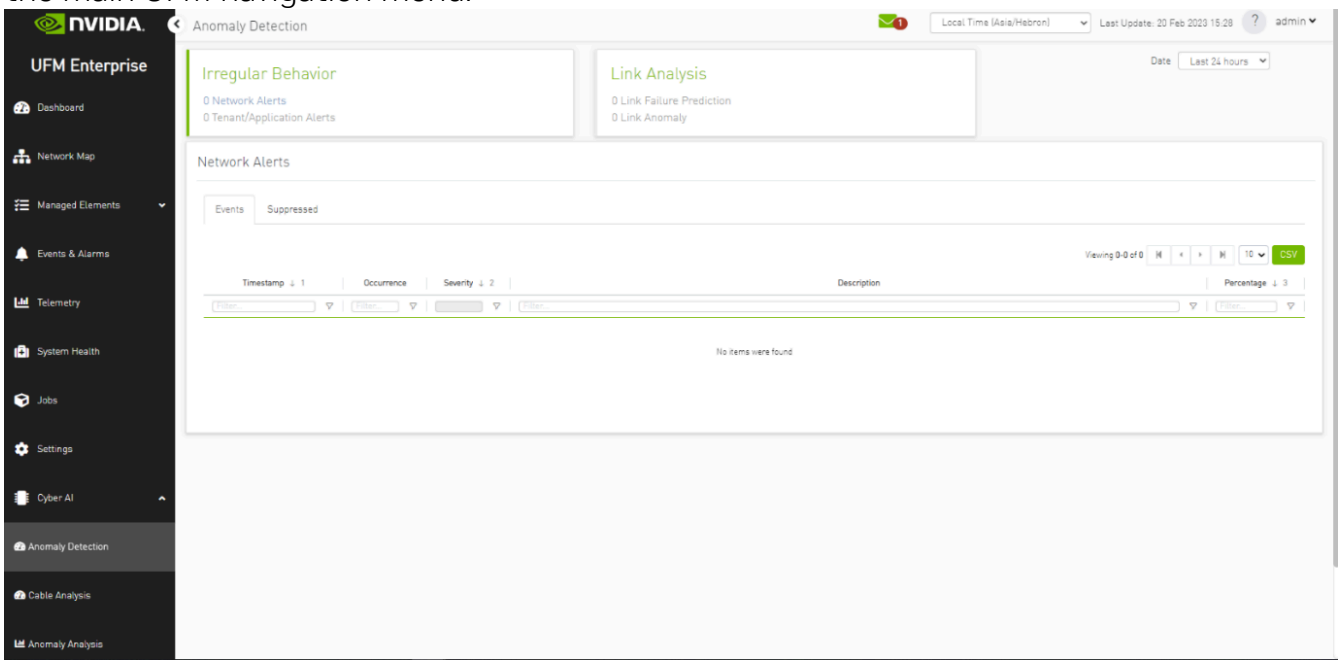
1. Login to the UFM server terminal.
2. Run:

```
docker load -I <path_to_image>
```

Once the plugin's image has been successfully loaded, you can locate the plugin in the Plugins management table within the UFM GUI. You can then run the plugin by right-clicking on the row associated with the plugin.



After running the plugin successfully. You should be able to see the Cyber-AI items under the main UFM navigation menu:



For more details, please refer to the UFM Cyber-AI User Manual

Autonomous Link Maintenance (ALM) Plugin

Overview

The primary objective of the Autonomous Link Maintenance (ALM) plugin is to enhance cluster availability and improve the rate of job completion. This objective is accomplished by utilizing machine learning (ML) models to predict potential link failures. The plugin then isolates the expected failing links, implements maintenance procedures on them, and subsequently restores the fixed links to their original state by removing the isolation.

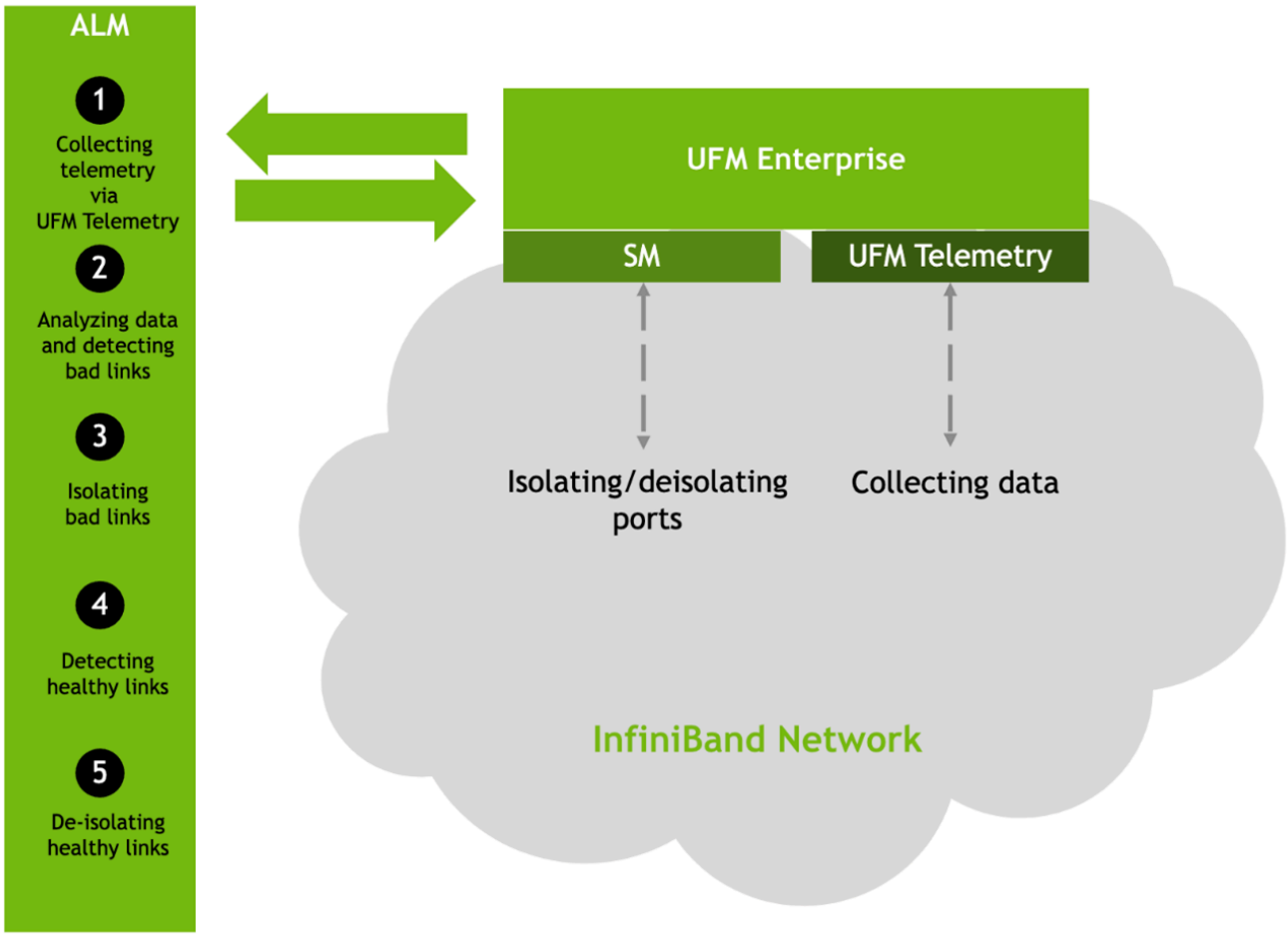
The ALM plugin performs the following tasks:

1. Collects telemetry data from UFM and employs ML jobs to predict which ports need to be isolated/de-isolated
2. Identifies potential link failures and isolates them to avert any interruption to traffic flow
3. Maintains a record of maintenance procedures that can be executed to restore an isolated link
4. After performing the required maintenance, the system verifies if the links can be de-isolated and restored to operational status (brought back online)

The ALM plugin operates in the following two distinct modes:

1. Shadow mode
    - Collects telemetry data, runs ML prediction jobs, and saves the predictions to files.
2. Active mode
    - Collects telemetry data, runs ML prediction jobs, and saves the predictions to files.
    - Automatically isolates and de-isolates based on predictions.
    - It is essential to note that a subset of the links must be specified in the allow list to enable this functionality.

Schematic Flow



Deployment

The Autonomous Link Maintenance (ALM) plugin can be deployed using the following methods:

1. On the UFM Appliance
2. On the UFM Software

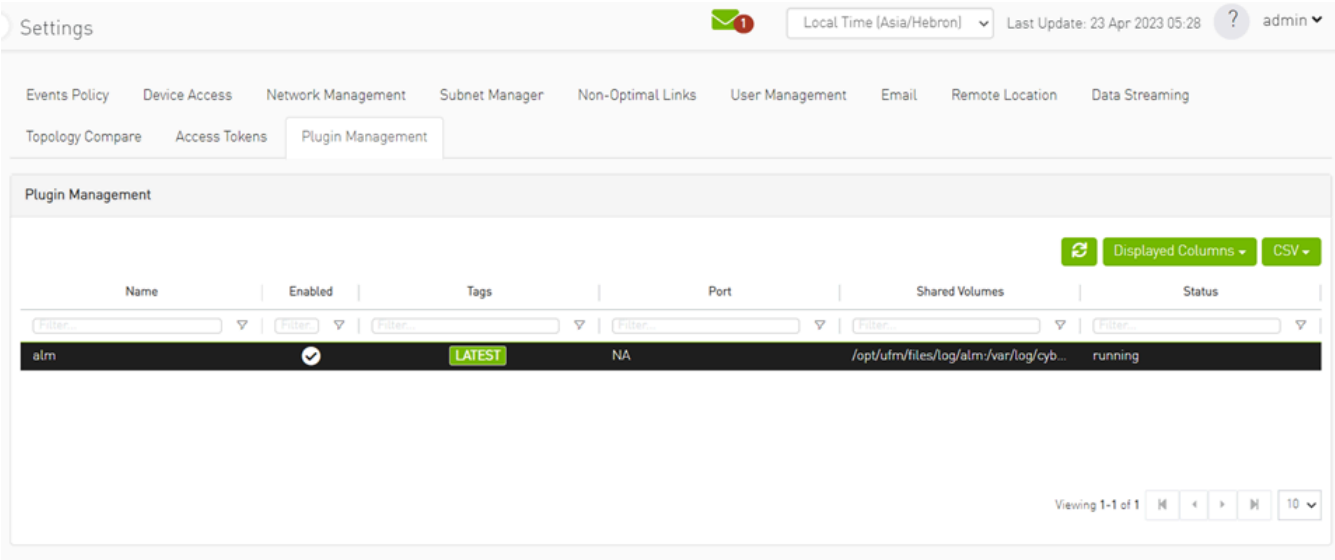
To deploy the plugin, follow these steps:

1. Download the ufm-plugin-alm-image from the NVIDIA License Portal (NLP).
2. Load the downloaded image onto the UFM server. This can be done either by using the UFM GUI by navigating to the Settings -> Plugins Management tab or by loading the image via the following instructions:
3. Log in to the UFM server terminal.

4. Run:

```
docker load -I <path_to_image>
```

5. After successfully loading the plugin image, the plugin should become visible within the plugins management table within the UFM GUI. To initiate the plugin's execution, simply right-click on the respective in the table.



> **Note**
>
> **The supported InfiniBand hardware technologies are HDR, Beta on NDR.**

Data Collection

The ALM plugin collects data from the UFM Enterprise appliance in the following two methods:

1. Low-frequency collection: This process occurs every 0 minutes and gathers data for the following counter: hist0, hist1, hist2, hist3, hist4, phy_effective_errors, phy_symbol_errors

2. High-frequency collection : This process occurs every 10 seconds and gathers data for the following counters:
   phy_state,logical_state,link_speed_active,link_width_active,fec_mode_active,
   raw_ber,eff_ber,symbol_ber,phy_raw_errors_lane0,phy_raw_errors_lane1,phy_raw_err
   phy_raw_errors_lane3,phy_effective_errors,phy_symbol_errors,time_since_last_clear,
   hist0,hist1,hist2,hist3,hist4,switch_temperature,CableInfo.temperature,link_down_eve
   plr_rcv_codes,plr_rcv_code_err,plr_rcv_uncorrectable_code,plr_xmit_codes,plr_xmit_r
   plr_xmit_retry_events,plr_sync_events,hi_retransmission_rate,fast_link_up_status,
   time_to_link_up,status_opcode,status_message,down_blame,local_reason_opcode,
   remote_reason_opcode,e2e_reason_opcode,num_of_ber_alarams,PortRcvRemotePhy
   PortRcvErrorsExtended,PortXmitDiscardsExtended,PortRcvSwitchRelayErrorsExtend
   VL15DroppedExtended,PortXmitWaitExtended,PortXmitDataExtended,PortRcvDataE
   PortRcvPktsExtended,PortUniCastXmitPktsExtended,PortUniCastRcvPktsExtended,P

3. The collected counters can be configurable and customized to suit your requirements. The counters can be found at /opt/ufm/conf/plugins/alm/counters.cfg

```
root@r-ufm116:~# cat  /opt/ufm/conf/plugins/alm/counters.cfg
[HighFreq]
phy_state = last_update_value
logical_state = last_update_value
link_speed_active = last_update_value
link_width_active = last_update_value
fec_mode_active = last_update_value
raw_ber = last_update_value
eff_ber = last_update_value
symbol_ber = last_update_value
phy_raw_errors_lane0 = delta
phy_raw_errors_lane1 = delta
phy_raw_errors_lane2 = delta
phy_raw_errors_lane3 = delta
phy_effective_errors = delta
phy_symbol_errors = delta
time_since_last_clear = last_update_value
hist0 = delta
hist1 = delta
hist2 = delta
hist3 = delta
hist4 = delta
switch_temperature = last_update_value
CableInfo.Temperature = last_update_value
link_down_events = delta
plr_rcv_codes = delta
plr_rcv_code_err = delta
plr_rcv_uncorrectable_code = delta
plr_xmit_codes = delta
plr_xmit_retry_codes = delta
plr_xmit_retry_events = delta
plr_sync_events = delta
hi_retransmission_rate = delta
fast_link_up_status = last_update_value
time_to_link_up = last_update_value
status_opcode = last_update_value
status_message = last_update_value
down_blame = last_update_value
local_reason_opcode = last_update_value
remote_reason_opcode = last_update_value
e2e_reason_opcode = last_update_value
num_of_ber_alarams = delta
PortRcvRemotePhysicalErrorsExtended = delta
PortRcvErrorsExtended = delta
PortXmitDiscardsExtended = delta
PortRcvSwitchRelayErrorsExtended = delta
```

ALM Configuration

The ALM configuration is used for controlling isolation/de-isolation. The configuration can be found under `/opt/ufm/cyber-ai/conf/cyberai.cfg.`

Name	Section name	Description
`mode`	CyberAi	The mode can be active or shadow The active mode means the alm will apply isolation/deisolation rule omn all ports exceptin the port in the expect list And the shadow mode mean the alm will apply isolation/deisolation rules on the ports on the except list The mode can be either "active" or "shadow." In active mode, the ALM will enforce isolation/deisolation rules on all ports except those listed in the "expect" list. In shadow mode, the ALM will enforce isolation/deisolation rules on the ports listed in the "except" list.
`except_list`	CyberAi	Includes the ports that receive the opposite treatment compared to the mode. Format: portguid_number, portguid_portnumber2
`max_per_hour`	Isolation	The maximum number of ports that can be isolated in a hour
`max_per_week`	Isolation	Maximum number of ports that can be isolated in a week
`max_per_month`	Isolation	Maximum number of the ports that can be isolated in a month

Name	Section name	Description
`Deisolation_time`	DeIsolation	The waiting time before deisolate the isolated port
`max_per_hour`	DeIsolation	The maximum number of deisolated port per hour
`absolute_threshold_of_isolated_ports`	Isolation	The maximum number of ports than can be isolated in one sample

ALM Jobs

The table presented below displays the names and descriptions of ALM jobs. These jobs are designed to predict the ports that require isolation/de-isolation. Upon enabling the ALM plugin, these ALM jobs run periodically.

ALM Job Name	Description	Frequency
Port_hist	By using **the** low frequency bit error histogram counters, the ALM job identifies the ports that will be monitored at high frequency in the next time interval. The job generates an output file that is later read by the high frequency telemetry monitoring job. It prioritizes links that are more susceptible to failure.	600 seconds
Low_freq_predict	Predicts the likelihood of a port failure by analyzing input data from low frequency telemetry, while only utilizing physical layer counters. The prediction works for isolated ports as well. The resulting output from this task serves as a critical input for determining whether to isolate or de-isolate ports.	10 seconds

DTS Plugin

Overview

The DTS Monitor can be run either as a standalone tool or as a plugin within UFM. It collects all the endpoint information for DPUs and consolidates it into a single interface.

Deployment

DPU Requirements

- OS: ubuntu 20/22
- BlueField: BlueField-2 or BlueField-3
- DTS: version > 1.12
- DPE service up and running
- yaml configured with "DTS_CONFIG_DIR=ufm"
-
    - Add to the following line in `file doca_telemetry_standalone.yaml`
-
    -
        - Command:

```
/bin/bash","-c","/usr/bin/telemetry-init.sh &&
/usr/bin/enable-fluent-forward.sh
```

        - Command:

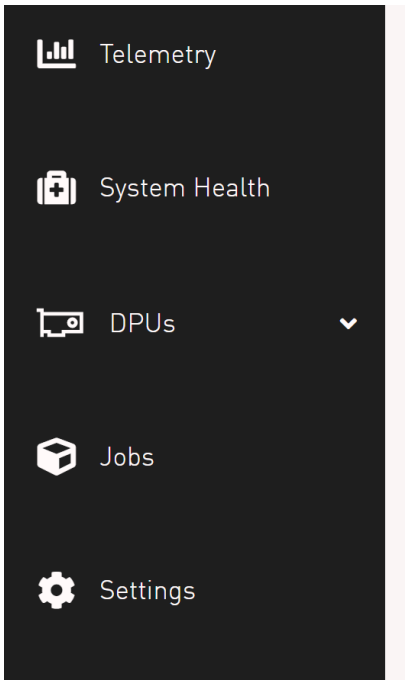
```
/bin/bash","-c"," DTS_CONFIG_DIR=ufm
/usr/bin/telemetry-init.sh && /usr/bin/enable-
fluent-forward.sh
```

Installation

you need to load the image on the UFM server; either using the UFM GUI -> Settings -> Plugins Management tab or by loading the image via the following command:

1. Login to the UFM server terminal.
2. Run: *docker load -I <path_to_image>*

After completing the plugin addition and refreshing the UFM GUI, a new menu item, titled DPUs, will be added to the left navigation bar.



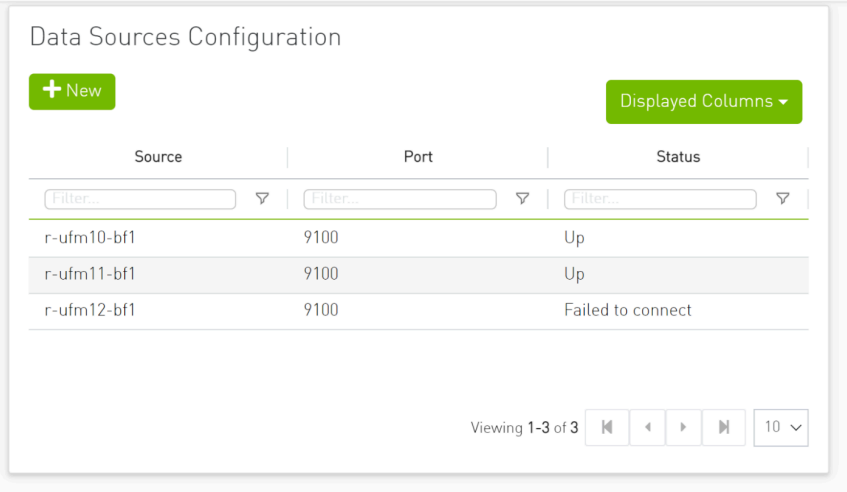
GUI Screens

Info

Health

Telemetry

Data Sources



GRPC-Streamer Plugin

Authentication

The following authentication types are supported:

- Basic (/ufmRest)
- Token (/ufmRestV3)

Create a Session to UFM from GRPC

Description: Creates a session to receive REST API results from the UFM's GRPC server. After a stream or one call, the session is deleted so the server would not save the authorizations.

- Call: CreateSession in the grpc
- Request Content Type – message SessionAuth

- Request Data:

```
message SessionAuth{
  string job_id=1;
  string username = 2;
  string password = 3;
  optional string token = 4;
}
```

- Job_id - The unique identifier for the client you want to have
- Username - The authentication username
- Password – The authentication password
- Token – The authentication token
- Response:

```
message SessionRespond{
  string respond=1;
}
```

- Respond types:
  - Success – Ok.
  - ConnectionError – UFM connection error (bad parameters or UFM is down).
  - Other exceptions – details sent in the respond.
- Console command:

```
client session --server_ip=server_ip --id=client_id --
auth=username,password --token=token
```

Create New Subscription

- Description: Only after the server has established a session for this grpc client,add all the requested REST APIs with intervals and delta requests.
- Call: AddSubscriber
- Request Content Type – Message SubscriberParams
- Request Data:

```
message SubscriberParams{
  message APIParams {
    string ufm_api_name = 1;
    int32 interval = 2;
    optional bool only_delta = 3;
  }
  string job_id = 1;
  repeated APIParams apiParams = 2;
}
```

- Job_id – A unique subscriber identifier
- apiParams – The list of apiParams from the above message above:
  - ufm_api_name – The name from the known to server request api list
  - interval – The interval between messages conducted in a stream run. Presented in seconds.
  - only_delta – Receives the difference between the previous messages in a stream run.

- Response content type:

```
message SessionRespond{
  string respond=1;
}
```

- Respond Types:
  - Created a user with session and added new IP– Ok.
  - Cannot add subscriber that do no have an established session – need to create a session before creating subscriber.
  - The server already have the ID – need to create new session and new subscriber with a new unique ID.
- Console command:

```
client create --server_ip=localhost --id=client_id --
apis=events;40;True,links,alarms;10
```

The API's list is separated by commas, and each modifier for the REST API is separated by a semi comma.

If the server is not given a modifier, default ones are used (where only_delta is False and interval is based on the API).

Edit Known Subscription

- Description: Changes a known IP. Whether the server has the IP or not.
- Call: AddSubscriber
- Request Content Type – Message SubscriberParams
- Request Data:

```
message SubscriberParams{
  message APIParams {
    string ufm_api_name = 1;
    int32 interval = 2;
    optional bool only_delta = 3;
  }
  string job_id = 1; //unique identifier for this job
  repeated APIParams apiParams = 2;
}
```

- Job_id – The subscriber unique identifier
- apiParams – A list of apiParams from the above message.
  - ufm_api_name – name from the known to server request api list
  - interval – The interval between messages conducted in a stream run. Presented in seconds.
  - only_delta – Receives the difference between the previous messages in a stream run.
- Response content type:

```
message SessionRespond{
  string respond=1;
}
```

- Respond Types:
  - Created user with new IP– Ok.
  - Cannot add subscriber without an established session – need to create a session before creating subscriber.

- Cannot add subscriber illegal apis – cannot create subscriber with empty API list, call again with correct API list.

Get List of Known Subscribers

- Description: Gets the list of subscribers, including the requested list of APIs.
- Call: ListSubscribers
- Request Content Type: google.protobuf.Empty
- Response:

```
message ListSubscriberParams{
  repeated SubscriberParams subscribers = 1;
}
```

- Console command: server subscribes –-server_ip=server_ip

Delete a Known Subscriber

- Description: Deletes an existing subscriber and removes the session.
- Call: DeleteSubscriber
- Request Content Type: Message gRPCStreamerID
- Request Data:

```
message gRPCStreamerID{
string job_id = 1;
}
```

- Response:protobuf.Empty

Run a Known Subscriber Once

- Description: Runs the Rest API list for a known subscriber once and returns the result in message runOnceRespond, and then delete the subscriber's session.
- Call: RunOnceJob
- Request Content Type: Message gRPCStreamerID
- Request Data:

```
message gRPCStreamerID{
string job_id = 1;
}
```

- Response content type:

```
message runOnceRespond{
  string job_id=1;
  repeated gRPCStreamerParams results = 2;
}
```

- Job_id- The first message unique identifier.
- Results – list of gRPCStreamerParams contains results from each REST API
- Responses:
    - Job id - Cannot run a client without an established session. Empty results – an existing session for this client is not found, and the client is not known to the server.
    - Job id - Cannot run the client without creating a subscriber. Empty results – a session was created for the client but the subscription is not created.
    - Job_id - Cannot connect to the UFM. empty result – the GRPC server cannot connect to the UFM machine and receive empty results, because it cannot create a subscriber with an empty API list. This means that the UFM machine is experiencing a problem.

    - Job_id - The first unique message identifier of the messages. Not empty results – Ok

- Console command:

```
client once_id --server_ip=server_ip --id=client_id
```

Run Streamed Data of a Known Subscriber

- Description: Run a stream of results from the Rest API list for a known Subscriber and return the result as interator, where each item is message gRPCStreamerParams. at the end, delete the session.

- Call: RunStreamJob

- Request Content Type: Message gRPCStreamerID

- Request Data:

```
message gRPCStreamerID{
string job_id = 1;
}
```

- Response content type: iterator of messages gRPCStreamerParams

```
message gRPCStreamerParams{
  string message_id = 1; // unique identifier for messages
  string ufm_api_name = 2; // what rest api receive the data from
  google.protobuf.Timestamp timestamp = 3; //what time we created the message, can be converted to Datetime
  string data = 4; // data of rest api call
}
```

- Response:

- One message only containing "Cannot run a client without a session" – A session has not been established
- No message – A session and/or a subscriber with this ID does not exist.
- Messages with interval between with the modifiers – Ok

- Console command:

```
client stream_id --server_ip=server_ip --id=client_id
```

Run a New Subscriber Once

- Description: After ensuring that a session for this specific job ID is established, the server runs the whole REST API list for the new subscriber once and returns the following result in message `runOnceRespond.` This action does not save the subscribe ID or the established session in the server.
- Call: RunOnce
- Request Content Type: Message SubscriberParams
- Request Data:

```
message SubscriberParams{
  message APIParams {
    string ufm_api_name = 1;
    int32 interval = 2;
    optional bool only_delta = 3;
  }
  string job_id = 1; //unique identifier for this job
  repeated APIParams apiParams = 2;
}
```

- Response content type:

```
message runOnceRespond{
  string job_id=1;
  repeated gRPCStreamerParams results = 2;
}
```

- Responses:
  - Job id = Cannot run a client without an established session. Empty results – no session for this client.
  - Job_id = 0 – The GRPC server cannot connect to the UFM machine and receive empty results, or it cannot create a subscriber with an empty API list.
  - Job_id = The messages' first unique identifier, and not an empty result – Ok.
- Console command:

```
client once --server_ip=server_ip --id=client_id --
auth=username,password --–token=token --
apis=events;40;True,links;20;False,alarms;10
```

- The console command creates a session for this specific client.
- A token or the basic authorization is needed, not both.

Run New Subscriber Streamed Data

- Description: After the server checks it has a session for this job ID, Run a stream of results from the Rest API list for a new Subscriber and return the result as interator, where each item is message gRPCStreamerParams. at the end, delete the session.
- Call: RunPeriodically
- Request Content Type: Message SubscriberParams
- Request Data:

```
message SubscriberParams{
  message APIParams {
    string ufm_api_name = 1;
    int32 interval = 2;
    optional bool only_delta = 3;
  }
  string job_id = 1; //unique identifier for this job
  repeated APIParams apiParams = 2;
}
```

- Response content type: iterator of messages gRPCStreamerParams
- Response:
  - Only one message with data equals to Cant run client without session – no session
  - Messages with intervals between with the modifiers – Ok
- Console command:

```
client stream --server_ip=server_ip --id=client_id --
auth=username,password --token=token --
apis=events;40;True,links;20;False,alarms;10
```

- console command also create session for that client.
- no need for both token and basic authorization, just one of them.

Run A Serialization on All the Running Streams

- Description: Run a serialization for each running stream. The serialization will return to each of the machines the results from the rest api list.
- Call: Serialization

- Request Content Type: google.protobuf.Empty
- Response: google.protobuf.Empty

Stop a Running Stream

- Description: Cancels running stream using the client id of the stream and stop it from outside, If found stop the stream.
- Call: StopStream
- Request Content Type: Message gRPCStreamerID
- Request Data:

```
message gRPCStreamerID{
string job_id = 1;
}
```

- Response: google.protobuf.Empty

Run a subscribe stream

- Description: Create a subscription to a client identifier, all new messages that go to that client, will be copied and also sent to this stream.
- Call: Serialization
- Request Content Type: message gRPCStreamerID
- Response: iterator of messages gRPCStreamerParams

```
message gRPCStreamerParams{
  string message_id = 1; // unique identifier for messages
  string ufm_api_name = 2; // what rest api receive the data from
  google.protobuf.Timestamp timestamp = 3; //what time we created the message, can be converted to Datetime
  string data = 4; // data of rest api call
}
```

- the identifier may or may not be in the grpc server.
- Cannot be stop streamed using StopStream.
- Console command:

```
client subscribe --server_ip=server_ip --id=client_id
```

Get the variables from a known subscriber

- Description: Get the variables of known subscriber if found, else return empty variables.
- Call: GetJobParams
- Request Content Type: message gRPCStreamerID
- Response:

```
message SubscriberParams{
  message APIParams {
    string ufm_api_name = 1; //currently the list of api from ufm that are supported are [Jobs, Events, Links, Alarms]
    int32 interval = 2;
    optional bool only_delta = 3;
  }
  string job_id = 1; //unique identifier for this job
  repeated APIParams apiParams = 2;
}
```

Get Help / Version

- Description: Get help and the version of the plugin, how to interact with the server. What stages need to be done to extract the rest apis (Session>run once/stream or Session>AddSubscriber>once_id/stream_id)
- Call: Help or Version
- Request Content Type: google.protobuf.Empty
- Response:

```
message SessionRespond{
  string respond=1;
}
```

Sysinfo Plugin

Overview

The Sysinfo plugin is a Docker container that is managed by UFM and comes with REST API support. Its purpose is to allow users to run commands and extract information from managed switches. This feature enables users to schedule runs at regular intervals and execute commands on switches directly from UFM.

The plugin takes care of managing sessions to the switches and can extend them if necessary. It also enables users to send both synchronous and asynchronous commands to all the managed switches. Additionally, it can intersect the given switches with the running UFM to ensure that only those switches that are on the UFM are activated.

Deployment

The following are the possible ways plugin plugin can be deployed:

1. On UFM Appliance
2. On UFM Software.
3. Authentication

Following authentication types are supported:

- basic (/ufmRest)
- client (/ufmRestV2**)**
- token (/ufmRestV3)

REST API

The following REST APIs are supported:

- GET /help
- GET /version
- POST /query
- POST /update
- POST /cancel
- POST /delete

Sysinfo Query Format

The Sysinfo plugin is responsible for extracting basic data needed to create a query. This is done using the following five fields:

1. Switches - An array of switch IP addresses. If this field is left empty, the plugin will gather all switches from the running UFM.
2. Callback - The URL location to which the answers should be sent.
3. Commands - An array of commands that need to be executed.
4. Schedule_run - An optional field used to set intervals for running the commands. The interval can be specified in seconds and can be set to run until a certain duration or end time. The start time can also be controlled.

There are additional flags for a configurable query:

- `ignore_ufm=True:` Does not check the UFM for switches or intersect it with given switches
- `username`: Overrides the switches' default username
- `password`: Overrides the switches' default password
- `is_async`: Rather than attempting to execute all commands simultaneously at the switch, the commands are executed one after the other in sequence.
- `one_by_one=False:` Instead of sending results from each switch as soon as information is obtained, all data is sent at once to the callback. This change eliminates multiple small sends and replaces them with a single large send.

For detailed information on how to interact with Sysinfo plugin, refer to the NVIDIA UFM Enterprise > Rest API > Sysinfo Plugin REST API.

SNMP Plugin

The SNMP plugin is a self-contained Docker container that includes REST API support and is managed by UFM. Its primary function is to receive SNMP traps from switches and forward them to UFM as external events. This feature enhances the user experience by

providing additional information about switches in the InfiniBand fabric via UFM events and alarms.

Deployment

There are two potential deployment options for the SNMP plugin:

- On UFM Appliance
- On UFM Software

For detailed instructions on how to deploy the SNMP plugin, refer to this page.

Authentication

The following authentication types are supported:

- basic (/ufmRest)
- client (/ufmRestV2**)**
- token (/ufmRestV3)

REST API

The following REST API are supported:

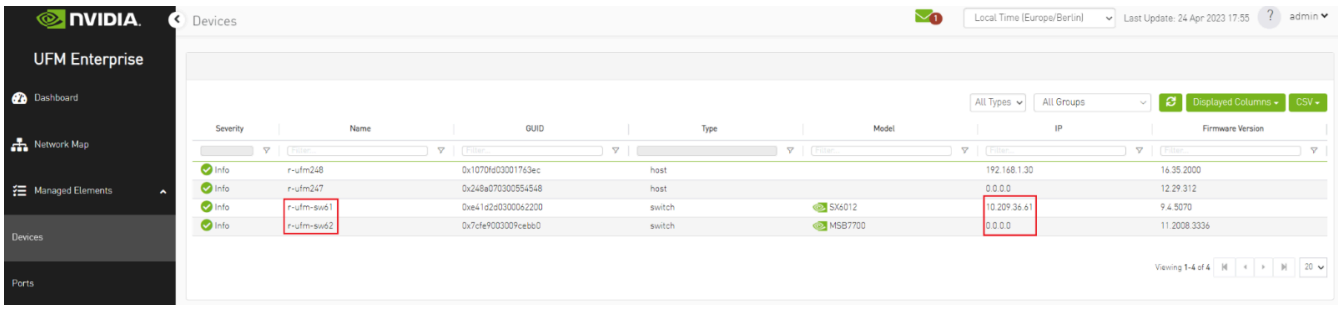
- GET /switch_list
- GET /trap_list
- POST /register
- POST /unregister
- POST /enable_trap
- POST /disable_trap
- GET /version

For more information, please refer to UFM Enterprise Documentation → UFM REST API → SNMP Plugin REST API.

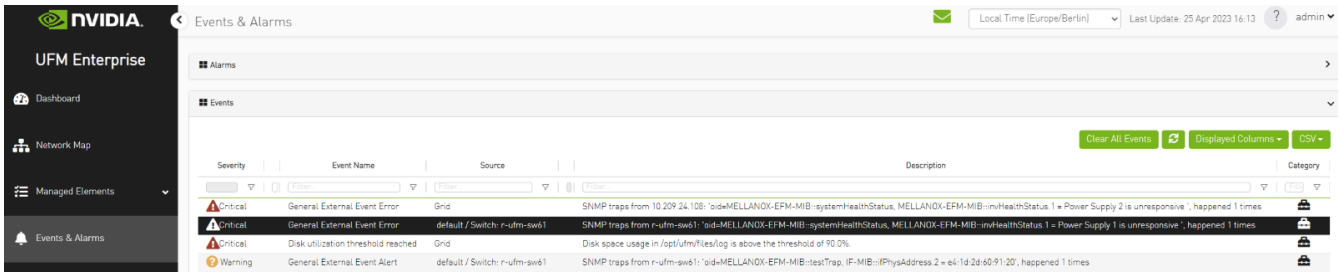
Usage

By default, upon initialization, the SNMP plugin captures traps from all switches within the fabric. However, this behavior can be modified through configuration settings utilizing the "snmp_mode" option, with available values of "auto" or "manual".

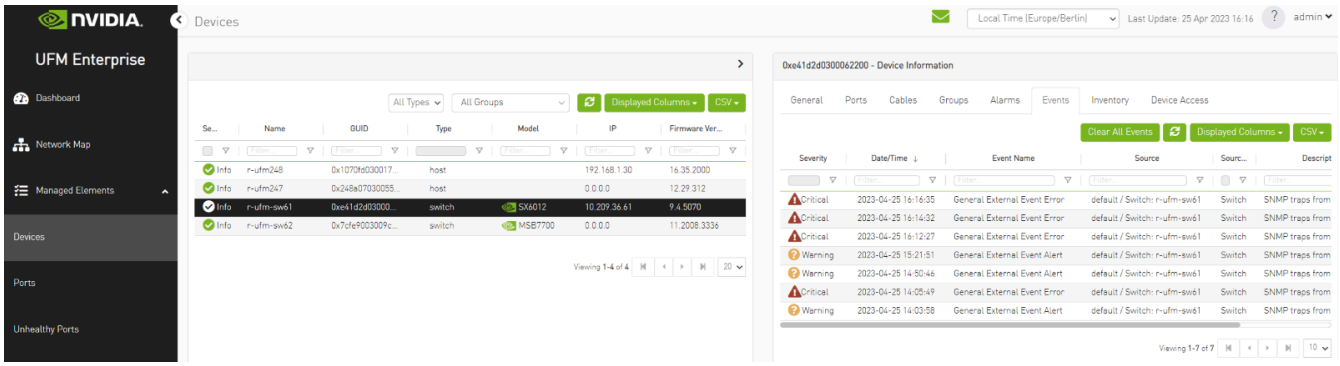
It is important to ensure that the switch is visible to UFM and has a valid IP address. As illustrated in the following example, switch traps will only be received from "r-ufm-sw61".



The following is an instance of a trap received by the SNMP plugin and displayed as a UFM event:



Additionally, there is an option to verify events/alarms for a particular switch:



The SNMP plugin performs a periodic check of the fabric every 180 seconds, allowing for prompt receipt of traps from new switches or updated IP addresses of existing switches in under 180 seconds. This interval may be adjusted via the

"ufm_switches_update_interval" option. To manually register or unregister a switch, please refer to the UFM Enterprise Documentation → UFM REST API → SNMP Plugin REST API.

The SNMP plugin employs the most up-to-date SNMP v3 protocol, which incorporates advanced security measures such as authentication and encryption. The "snmp_version" option enables the selection of SNMP versions “1” or “3”. It is essential to note that only switch-exposed traps will be transmitted to UFM as events.

OID	Name	Description	Status	Severity
MELLANOX-EFM-MIB::testTrap	send-test	A test trap ordered by the system administrator	Enabled	Warning
MELLANOX-EFM-MIB::asicChipDown	asic-chip-down	ASIC (Chip) Down	Enabled	Critical
MELLANOX-EFM-MIB::cpuUtilHigh	cpu-util-high	CPU utilization has risen too high	Enabled	Warning
MELLANOX-EFM-MIB::diskSpaceLow	disk-space-low	Filesystem free space has fallen too low	Enabled	Warning
MELLANOX-EFM-MIB::expectedShutdown	expected-shutdown	Expected system shutdown	Enabled	Info
MELLANOX-EFM-MIB::systemHealthStatus	health-module-status	Health module Status	Enabled	Critical
MELLANOX-EFM-MIB::insufficientFans	insufficient-fans	Insufficient amount of fans in system	Enabled	Warning
MELLANOX-EFM-MIB::insufficientFansRecover	insufficient-fans-recover	Insufficient amount of fans in system recovered	Enabled	Info
MELLANOX-EFM-MIB::insufficientPower	insufficient-power	Insufficient power supply	Enabled	Warning
RFC1213::linkdown	interface-down	An interface's link state has changed to down	Enabled	Minor
RFC1213::linkup	interface-up	An interface's link state has changed to up	Enabled	Info

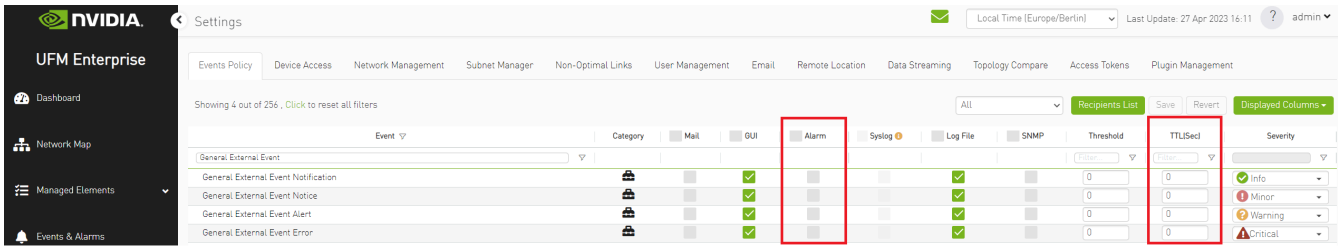
OID	Name	Description	Status	Severity
MELLANOX-EFM-MIB::unexpectedShutdown	unexpected-shutdown	Unexpected system shutdown	Enabled	Minor
SNMPv2-MIB::coldStart	cold-start	SNMP entity reinitialized	Enabled	Info

To learn more about how to enable or disable a specific trap, please refer to the UFM Enterprise Documentation → UFM REST API → SNMP Plugin REST API.

If some traps are not included in the default list, they may be added using the "snmp_additional_traps" option. The SNMP plugin will consider these traps as "enabled" and transmit them to UFM as events with an "Info" severity level.

To ensure the uninterrupted reception of traps from switches within a large fabric, changes must be made to the UFM configuration in the [/opt/ufm/conf/gv.cfg] file's [Events] section. Specifically, the "max_events" option should be raised from 100 to 1000, while "medium_rate_threshold" and "high_rate_threshold" should both be set to 500. To implement configuration adjustments, disable and then enable the plugin.

In case of an event storm, it is necessary to adjust the Event Policy settings such that General Events are non-alarmable and the TTL is set to zero, as illustrated in the following screenshot:



Other

Additional configurations are located in "/opt/ufm/conf/plugins/snmp/snmp.conf". To implement configuration adjustments, disable and then enable the plugin. For instructions on modifying the appliance, please refer to the UFM-SDN App CLI Guide.

Logs for the SNMP plugin are stored in "/opt/ufm/logs/snmptrap.log". For guidance on accessing logs on the appliance, please refer to the UFM-SDN App CLI Guide.

Packet Mirroring Collector (PMC) Plugin

Overview

The Packet Mirroring Collector/Controller plugin facilitates the configuration of pFRN and Congestion mirroring on switches and subsequently captures mirrored packets, enabling users to conduct real-time monitoring of network events.

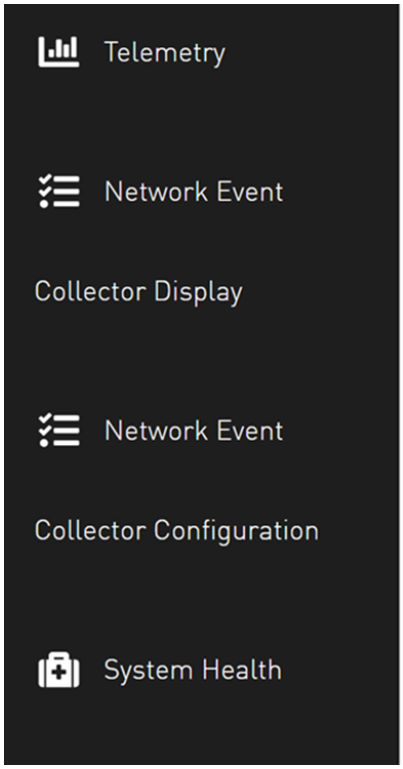
Deployment

Installation

Load the image on the UFM server; either using the UFM GUI -> Settings -> Plugins Management tab, or by loading the image via the following command:

1. Login to the UFM server terminal.
2. Run

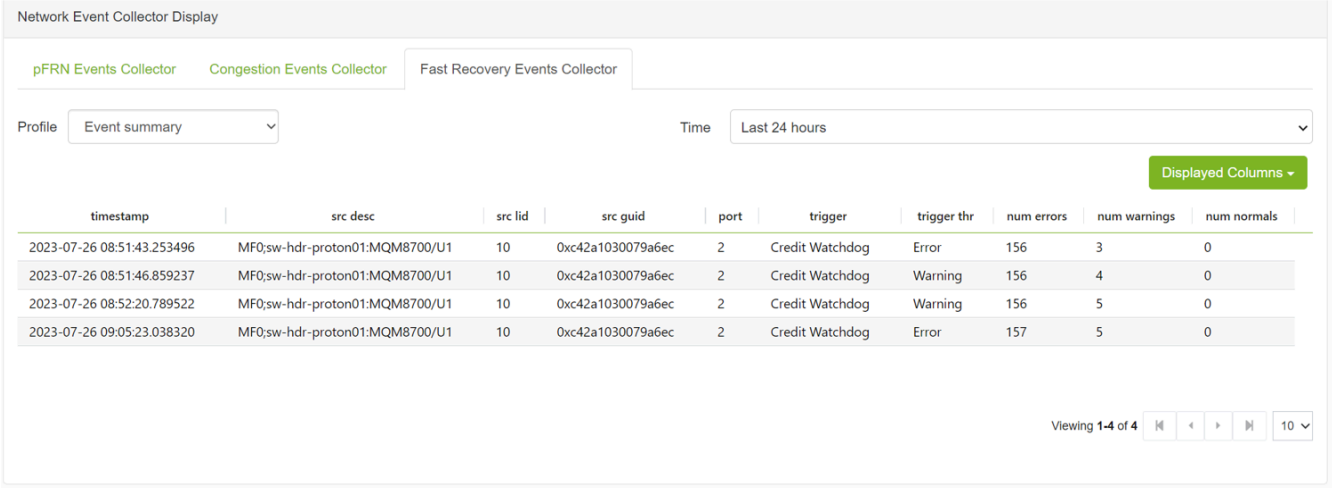
```
docker load -I <path_to_image>
```



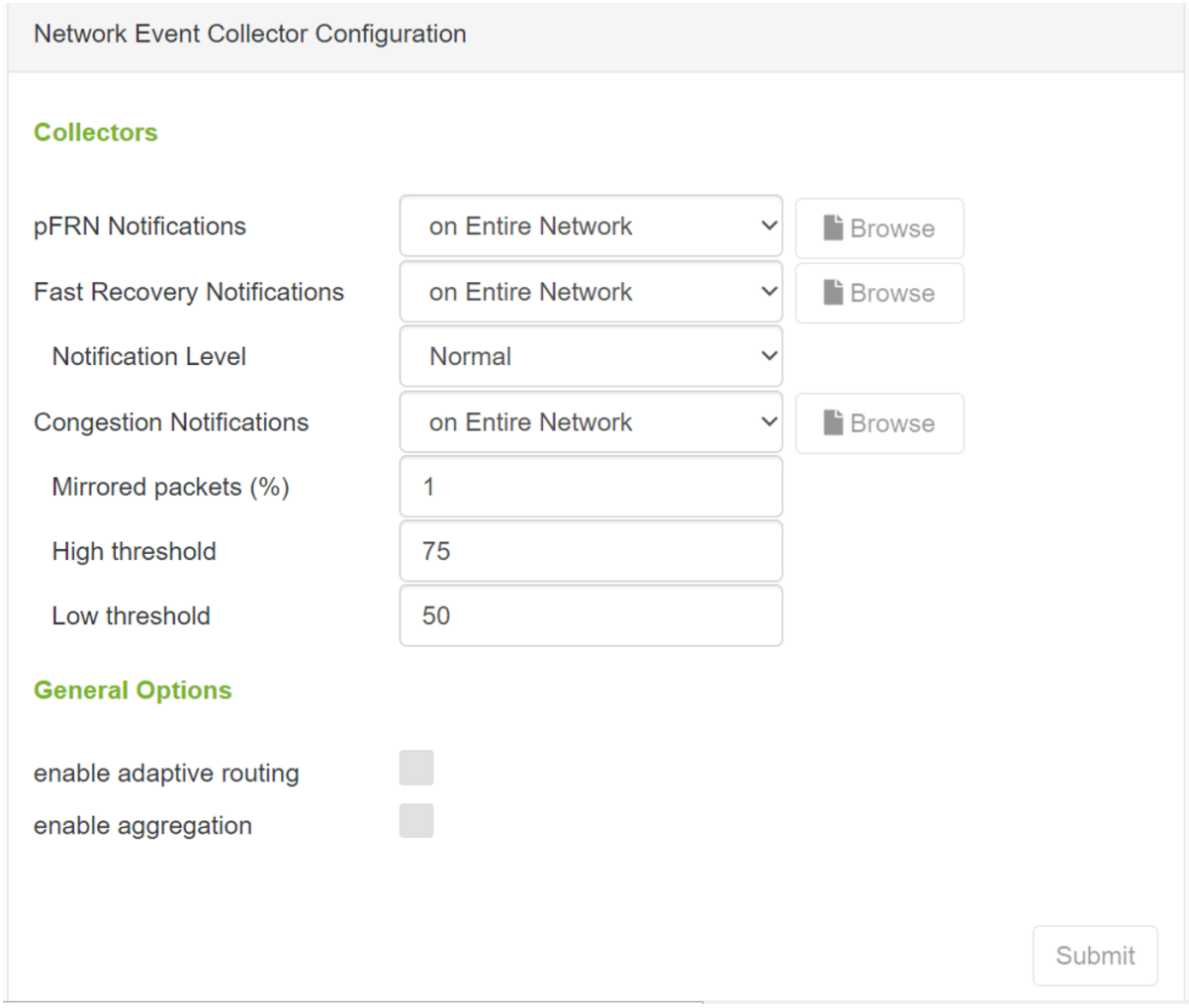
Upon completion of the plugin addition and subsequent refresh of the UFM GUI, the left navigation bar will display two new menu items. These two tabs can be observed in the following GUI screenshots

GUI Screens

Network Event Collector Display



Network Event Collector Configuration



PDR Deterministic Plugin

Overview

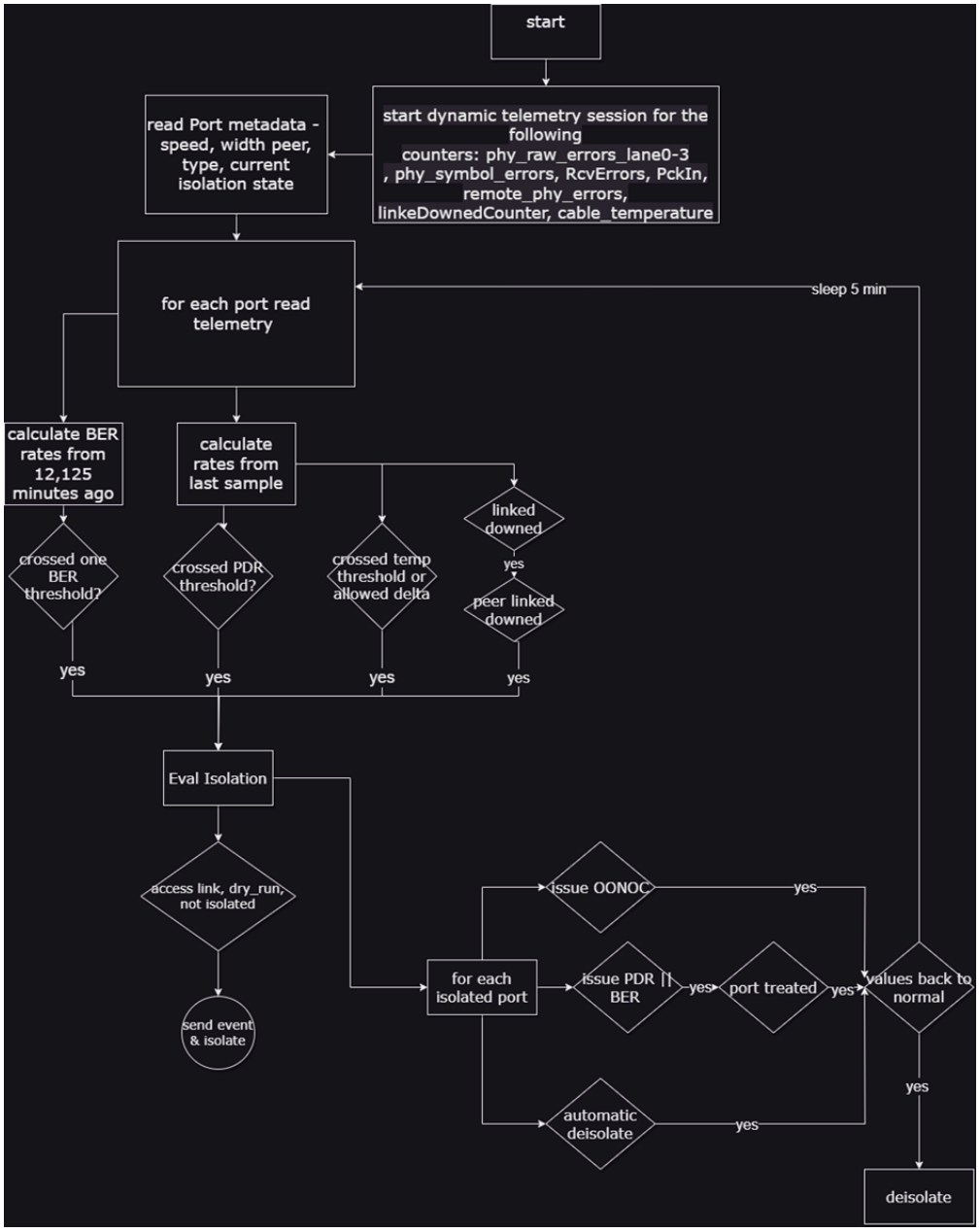
The PDR deterministic plugin, overseen by the UFM, is a docker container that isolates malfunctioning ports, and then reinstates the repaired links to their previous condition by lifting the isolation. The PDR plugin uses a specific algorithm to isolate ports, which is based on telemetry data from the UFM Telemetry. This data includes packet drop rate, BER counter values, link down counter, and port temperature. Any decisions made by the plugin will trigger an event in the UFM for tracking purposes.

The PDR plugin performs the following tasks:

1. Collects telemetry data using UFM Dynamic Telemetry
2. Identifies potential failures based on telemetry calculations and isolates them to avert any interruption to traffic flow
3. Maintains a record of maintenance procedures that can be executed to restore an isolated link
4. After performing the required maintenance, the system verifies if the ports can be de-isolated and restored to operational status (brought back online).

The plugin can simulate port isolation without actually executing it for the purpose of analyzing the algorithm's performance and decision-making process in order to make future adjustments. This behavior is achieved through the implementation of a "`dry_run`" flag that changes the plugin's behavior to solely record its port "isolation" decisions in the log, rather than invoking the port isolation API. All decisions will be recorded in the plugin's log.

Schematic Flow



Deployment

To deploy the plugin, follow these steps:

1. Download the ufm-plugin-pdr_deterministic-image from the **NVIDIA License Portal (NLP)**.

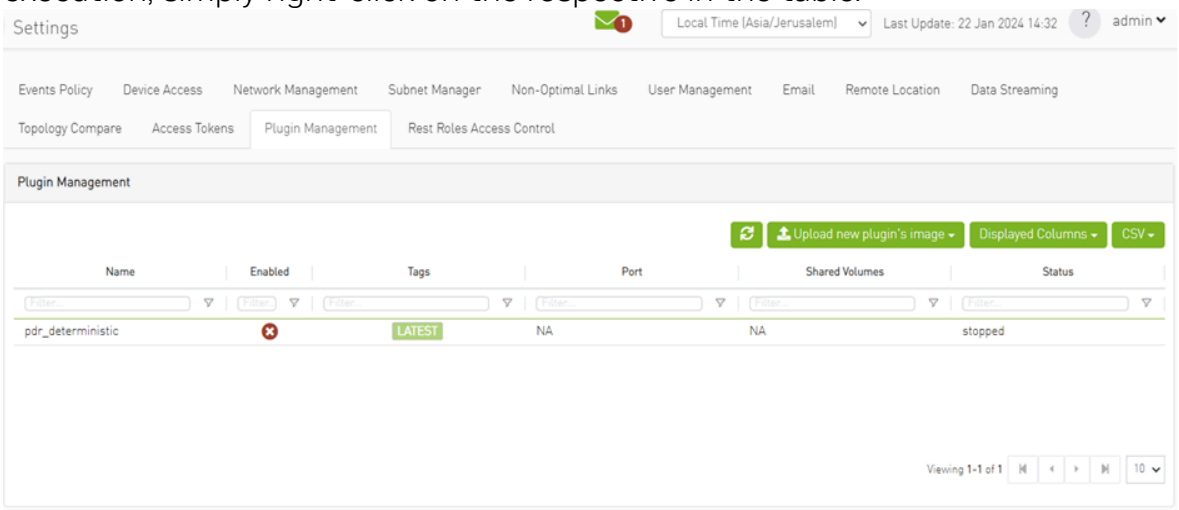
2. Load the downloaded image onto the UFM server. This can be done either by using the UFM GUI by navigating to the Settings -> Plugins Management tab or by loading the image via the following instructions:

    1. Log in to the UFM server terminal.

2. Run:

```
docker load -I <path_to_image>
```

3. After successfully loading the plugin image, the plugin should become visible in the plugin management table within the UFM GUI. To initiate the plugin's execution, simply right-click on the respective in the table.



Isolation Decisions

NDR Link Validation Procedure

Verify ports that are in INIT, ARMED or ACTIVE states only. Track the SymbolErrorsExt of every such link for at least 120m. If polling period is Pm, need to keep N=(125+Pm+1)/Pm samples. Also, two delta samples are computed: number of samples covering 12 minutes S12m = (12 + Pm + 1)/Pm and S125m = (125 + Pm + 1)/Pm. 12m_thd = LinkBW_Gbps*1e9*12*60*1e-14 (2.88 for NDR) and

125m_thd = LinkBW_Gbps*1e9*125*60*1e-15 (3 for NDR).

Check the following conditions for every port in the given set:

1. If the `Delta(LinkDownedCounterExt)` port is > 0 and the `Delta(LinkDownedCounterExt)` remote port is > 0, add it to the list of `bad_ports`. This condition should be ignored if the `--no_down_count` flag is provided.

2. If the `symbol_errors[now_idx] - symbol_errors[now_idx - S12m]` is > 12m_thd, add the link to the list of `bad_ports`, and continue with next link.

3. If the `symbol_errors[now_idx] - symbol_errors[now_idx - S125m]` is > 125m_thd, add the link to the list of `bad_ports`, continue with next linkPacket drop rate criteria

When packet drops due to the link health are detected, isolate the problematic link. To achieve this, a target packet_drop/packet_delivered ratio can be employed to include TX ports with a receiver exceeding this threshold in the list of bad_ports. However, the drawback of this method is that such links may fluctuate between bad/good state since their BER may be normal. Therefore, it is advisable to track their statistics over time and refrain from reintegrating them after their second or third de-isolation.

Return to Service

Continuously monitoring the collection of `bad_ports`, the plugin persistently assess their Bit Error Rate (BER) and determines their reintegration when they successfully pass the 126m test without errors.

Configuration

The following parameters are configurable via the plugin's configuration file. (`pdr_deterministic.conf`)

Name	Description	Default Value
`T_ISOLATE`	Interval for requesting telemetry counters, in seconds.	300
`MAX_NUM_ISOLATE`	Maximum ports to be isolated. max(MAX_NUM_ISOLATE, 0.5% * fabric_size)	10
`TMAX`	Maximum temperature threshold	70 (Celsius)
`D_TMAX`	Maximum allowed Temperature Delta	10

Name	Description	Default Value
`MAX_PDR`	Maximum allowed packet drop rate	1e-12
`CONFIGURED_BER_CHECK`	If set to true, the plugin will isolate based on BER calculations	True
`CONFIGURED_TEMP_CHECK`	If set to true, the plugin will isolate based on temperature measurements	True
`NO_DOWN_COUNT`	If set to true, the plugin will isolate based on LinkDownedCounterExt measurements	True
`ACCESS_ISOLATION`	If set to true, the plugin will isolate ports connected via access link	True
`DRY_RUN`	Isolation decisions will be only logged and will not take effect	False
`DEISOLATE_CONSIDER_TIME`	Consideration time for port de-isolation (in minutes)	5
`DO_DEISOLATION`	If set to false, the plugin will not perform de-isolation	True
`DYNAMIC_WAIT_TIME`	Seconds to wait for the dynamic telemetry session to respond	30

Calculating BER Counters

For calculating BER counters, the plugin extracts the maximum window it needs to wait for calculating the BER value, using the following formula:

$$seconds = \frac{max\_BER\_target^{-1}}{min\_port\_rate}$$

Example:

Rate		BER Target	Minimum Bits	Minimum Time in Seconds	In Minutes
HDR	2.00E+11	1.00E-12	1.00E+12	5	0.083333
HDR	2.00E+11	1.00E-13	1.00E+13	50	0.833333
HDR	2.00E+11	1.00E-14	1.00E+14	500	8.333333

Rate		BER Target	Minimum Bits	Minimum Time in Seconds	In Minutes
HDR	2.00E+11	1.00E-16	1.00E+16	50000	833.3333

BER counters are calculated with the following formula:

$$BER = \frac{error\ bits_i - error\ bits_{i-1}}{total\ bits_i - total\ bits_{i-1}} = \frac{error\ bits_i - error\ bits_{i-1}}{Link\ data\ rate * (time_i - time_{i-1})}$$

Deployment

1. Install UFM with the latest software version.
2. Run:

```
/etc/init.d/ufmd start
```

3. To get PDR plugin image, please contact the NVIDIA Support team. After that, load the plugin using this command:

   When working with UFM in HA mode, load the plugin on the standby node.

```
ufmapl [ mgmt-sa ] (config) # docker load ufm-plugin-pdr-
deterministic.tar
```

4. Run the following command. Add `-p pdr-deterministic` to enable the plugin:

```
/opt/ufm/scripts/manage_ufm_plugins.sh add -p pdr-
deterministic
```

5. Ensure that the plugin is up and running. Run:
   /opt/ufm/scripts/manage_ufm_plugins.sh show

GNMI-Telemetry Plugin

The GNMI Telemetry Plugin functions as a server that employs the gNMI protocol to stream data from UFM telemetry. Users can select what data to stream, specify the intervals, and choose whether to include only deltas (on-change mode).

The GNMI server is designed to support four functions: capability, get, subscribe, and set. However, it should be noted that the server does not currently support the "set" function, only "capability," "get," and "subscribe."

The streamed data is delivered in CSV format. Headers are initially provided in the first message, and subsequently, they are included in every other message. The data is presented in hex format to conserve space for data that remains unchanged. The values are presented as an array of strings, each representing a unique identifier (GUID) and port.

Depending on the selected mode, the values may have missing rows if there have been no changes in the GUID and port.

Furthermore, the plugin has the capability to stream UFM's metadata by providing an inventory of it. While the provided examples will use the gNMIc client for convenience, this functionality can work with any gNMI client.

Authentication

The server's authentication is determined by the gNMI protocol, and whether it is secured or unsecured is specified in the configuration. Two configurable items require authentication: the UFM Telemetry URL and the UFM inventory IP. Both of these items must be configured in the configuration file.

- Authentication is not necessary for the UFM telemetry URL. Therefore, only the telemetry URL is required.

- By default, the inventory is sourced from the UFM of the local host. However, it is possible to change the UFM inventory location to a different machine in the config file. To do so, token access to that machine is necessary.

Secure Server

The server can be secured by using certificates. To secure the server, modify the "`secure_mode_enabled`" flag to "`true`" in the configuration.

Upon initialization, the gNMI server retrieves the UFM certificates from the `/var/opt/ufm/webclient/` folder, utilizing both the server certificates and CA certificates. It is possible to change the certificate folder by changing the shared volume.

The server will requires certificates for client calls and grants access only if the client certificates match its own. The gNMI server periodically examines its certificates for updates and ensures that they remain up to date.

Capability Request

Description: The capability request provides information about the Yang files that the server supports, including their versions. This request can be fulfilled without the need for a connection to the telemetry or inventory.

Example:

```
gnmic -a localhost:9339 capability
```

Get Request

The Get request retrieves data at a specified path. If the telemetry is devoid of information, the server will respond with an empty response. Otherwise, it will respond with counters it can locate.

The path construction follows these steps:

1. Begin with "`nvidia/ib`"
2. Specify the `node_guid` that the user wants to select, with an asterisk (*) representing a selection of all nodes.
3. Choose the desired ports for the selected nodes.
4. Select "`amber`" and the desired counters group, and then specify the counter.

Example:

```
gnmic -a localhost:9339 --insecure get --path
nvidia/ib/guid[guid=0x5255456]/port[port_number=2]/amber/port_counter
```

The request from the above example is run from node_guid `0x5255456`, in port number 2, and the queried counter is hist0.

Example 2:

```
gnmic -a localhost:9339 --insecure get --path
nvidia/ib/guid[guid=*]/port[port_number=*]/amber/port_counters/his
```

The request from the above example is run from all the node_guids, in all ports, and the queried counter is hist0.

Example3:

```
gnmic -a localhost:9339 --insecure get --path
nvidia/ib/guid[guid=0x5255456]/port[port_number=2]/amber/*
```

The request from the above example is run from node_guid `0x5255456`, port 2, and all its counters.

Subscribe Stream Request

The subscribe request, similar to the get request, provides data from the specified path. When the telemetry is empty, the server responds with an empty result. However, if there is data available, the server responds with the counters it can locate. The stream delivers information at intervals corresponding to the requested interval. If a user fails to specify an interval, the server will transmit the information as soon as it becomes available. The path construction follows the same pattern as the get request.

Example:

```
gnmic -a localhost:9339 --insecure sub --path
nvidia/ib/guid[guid=0x5255456]/port[port_number=2]/amber/port_counter
-i 30s
```

TBD: This request from node_guid 0x5255456 port 2 the counter hist0 and set the interval to 30 seconds.

If the user wants to test the stream, the stream mode can be set to once, and after that one respond, the stream will be stopped.

Example:

```
gnmic -a localhost:9339 --insecure sub --path
nvidia/ib/guid[guid=0x5255456]/port[port_number=2]/amber/port_counter
-i 30s --mode once
```

TBD:This request is run from node_guid 0x5255456, port 2 the counter hist0 once, and then shut the stream off, much like a get request.

Subscribe On-Change Request

The subscribe on-change request, much like the standard subscribe request, provides data from the specified path. In the event that the telemetry lacks data, the server responds with an empty result. However, when data is available, the server responds with the counters it can locate. The stream delivers information according to the interval specified in the request, but only if there is new information to transmit. Otherwise, it will wait for the next interval to check the telemetry for updates. The path construction follows the same pattern as the get request.

Importantly, only the data that has been updated will be included in the response; all other parts will be empty but retain the specified format. Similarly, only the nodes that have been updated will be included in the response.

Example:

```
gnmic -a localhost:9339 --insecure  sub --path
nvidia/ib/guid[guid=0x5255456]/port[port_number=2]/amber/port_counter
 --stream-mode on-change --heartbeat-interval 1m
```

TBD: This request from node_guid 0x5255456 port 2 the counter hist0, every minute it will check for changes, if there are it will send the new value.

Example:

```
gnmic -a localhost:9339 --insecure  sub --path
nvidia/ib/guid[guid=*]/port[port_number=*]/amber/port_counters/*
--stream-mode on-change --heartbeat-interval 1m
```

This request involves all nodes and ports, aiming to retrieve all counters from the telemetry. It periodically checks for changes every minute, and when changes are detected, it promptly sends the updated values.

Messages Data Format

Telemetry messages consist of two key components: Headers and Values, both representing the telemetry data in CSV format. When utilizing a subscribe request, the headers transition to a string hash format after the second message, primarily to conserve message size. In the case of on-change subscribe messages, there is an additional adjustment where only nodes that have undergone changes are included, along with their corresponding modified values. All other counters for that node will remain empty.

Each value within the "Values" section starts with a timestamp, followed by the node_guid and port number, and then the value of the counter, maintaining the same order as the headers. If a specific counter is not present for the node, it will remain empty in the message.

Example:

```
gnmic -a localhost:9339 --insecure sub --path
nvidia/ib/guid[guid=*]/port[port_number=*]/amber/port_counters/hist
--path
nvidia/ib/guid[guid=*]/port[port_number=*]/amber/port_counters/hist
-i 30s
[{ "source": "localhost:9339",
   "subscription-name": "default-1690282472",
   "timestamp": 1690282475124352063,
   "time": "2023-07-25T13:54:35.124352063+03:00",
   "updates": [{ "Path": "hist0", "values": { "hist0": {
               "Headers": "timestamp,guid,port,hist0,hist1",
               "Values": ["240771222771818,0x8168793592c6a790,1,,2",
                  "240771222771818,0x47a67159c915493f,1,1,2",
…
                  "240771222771818,0x667203ac69f3f2bf,1,2,",
                  "240771222771818,0x113cd807bfed3853,1,0,"
               ]}}}]}
```

TBD: The second message and on the headers will be set to hash values.

Inventory Requests

Inventory messages are conveyed in separate updates, presenting the inventory details of the UFM associated with the provided IP. These messages display comprehensive information, including the total count of various components within the UFM, such as switches, routers, servers, and more, along with details about active ports and the total number of ports, including disabled ones. In cases where the plugin is unable to establish contact with the UFM, it will revert to using default values defined in the configuration file. It is worth noting that the path for inventory requests differs from the conventional path structure, as they do not rely on specific nodes or ports. Consequently, inventory requests are initiated after "`nvidia/ib`."

Example:

```
 gnmic -a localhost:9339 --insecure get –path
nvidia/ib/inventory/*
```

Response:

```
[{
      "source": "localhost:9339",
      "timestamp": 1698824237536878067,
      "time": "2023-11-01T09:37:17.536878067+02:00",
      "updates": [{
            "Path": "nvidia/ib/inventory",
            "values": {
               "nvidia/ib/inventory": {
                  "ActivePorts": 4, "Cables": 2, "Gateways": 0, "HCAs": 2, "Routers":
0,"Servers": 2,     "Switches": 1,"TotalPorts": 38,"timestamp":
1698824211535069000}
      }}]}]
```

Events Requests

Events messages are provided in separate updates, offering insights into the events occurring within the UFM associated with the specified IP. Given that the event metadata remains consistent, even when numerous events are part of a request, the message format adopts a CSV-like structure. The Headers section contains essential metadata regarding UFM events, while the Values section contains the raw event data. Users can subscribe to these events with the on-change feature enabled, receiving only the events triggered within the subscription interval. Notably, the path structure for event requests differs from the typical node or port-based structure and is requested after "`nvidia/ib`."

Example:

```
gnmic -a localhost:9339 --insecure get –path nvidia/ib/events/*
```

Response:


```
[ {
        "source" :  "localhost:9339" ,
        "timestamp" :  1698824809647515575 ,
        "time" :  "2023-11-01T09:46:49.647515575+02:00" ,
        "updates" :  [    {
                "Path" :  "nvidia/ib/events" ,
                "values" :  {
                    "nvidia/ib/events" :  {
                        "Headers" :  [
"id" , "object_name" , "write_to_syslog" , "description" , "type" , "event_type" , "severity" , "timestamp" , "counter"
                        "Values" :  [
                            "7718,Grid,false,Disk space usage in /opt/ufm/files/log is above the threshold of
90.0%.,Grid,525,Critical,2023-11-01 07:25:54,N/A,Maintenance,Grid,Disk utilization threshold reached" ,
                            "7717,Grid,false,Disk space usage in /opt/ufm/files/log is above the threshold of
90.0%.,Grid,525,Critical,2023-11-01 07:24:54,N/A,Maintenance,Grid,Disk utilization threshold reached" ,
                            "7716,Grid,false,Disk space usage in /opt/ufm/files/log is above the threshold of
90.0%.,Grid,525,Critical,2023-11-01 07:23:54,N/A,Maintenance,Grid,Disk utilization threshold reached" ,
….
                            "7491,ec0d9a0300d42e54,false,Mcast group is deleted: ff12601bffff0000,
00000002,Computer,67,Info,2023-10-31 06:39:21,N/A,Fabric Notification,default / Computer: r-
ufm59,MCast Group Deleted" ]                }
                }    }      ]    } ]
```

Split-Brain Recovery in HA Installation

The split-brain problem is a DRBD synchronization issue (HA status shows `DUnknown` in the DRBD disk state), which occurs when both HA nodes are rebooted. For example, in cases of electricity shut-down. To recover, please follow the below steps:

- **Step 1:** Manually choose a node where data modifications will be discarded.

  It is called the split-brain victim. Choose wisely; all modifications will be lost! When in doubt, run a backup of the victim's data before you continue.

  When running a Pacemaker cluster, you can enable maintenance mode. If the split-brain victim is in the Primary role, bring down all applications using this resource. Now switch the victim to the Secondary role:

  ```
  victim# drbdadm secondary ha_data
  ```

- **Step 2:** Disconnect the resource if it's in connection state **WFConnection**:

  ```
  victim# drbdadm disconnect ha_data
  ```

- **Step 3:** Force discard of all modifications on the split-brain victim:

  ```
  victim# drbdadm -- --discard-my-data connect ha_data
  ```

  For DRBD 8.4.x:

```
victim# drbdadm connect --discard-my-data ha_data
```

- **Step 4:** Resync starts automatically if the survivor is in a WFConnection network state. If the split-brain survivor is still in a Standalone connection state, reconnect it:

  ```
  survivor# drbdadm connect ha_data
  ```

  Now the resynchronization from the survivor (SyncSource) to the victim (SyncTarget) starts immediately. There is no full sync initiated, but all modifications on the victim will be overwritten by the survivor's data, and modifications on the survivor will be applied to the victim.

Appendixes

Appendix – Diagnostic Utilities

Note

For UFM-SDN Appliance, all the below diagnostics commands have ib prefix.

For example, for UFM-SDN Appliance, the command `ibstat` is `ib ibstat.`

InfiniBand Diagnostics Commands

Command	Description
ibstat	Shows the host adapters status.
ibstatus	Similar to ibstat but implemented as a script.
ibnetdiscover	Scans the topology.
ibaddr	Shows the LID range and default GID of the target (default is the local port).
ibroute	Displays unicast and multicast forwarding tables of the switches.
ibtracert	Displays unicast or multicast route from source to destination.
ibping	Uses vendor MADs to validate connectivity between InfiniBand nodes. On exit, (IP) ping-like output is shown.
ibsysstat	Obtains basic information for the specific node which may be remote. This information includes: hostname, CPUs, memory utilization.
sminfo	Queries the SMInfo attribute on a node.
smpdump	A general purpose SMP utility which gets SM attributes from a specified SMA. The result is dumped in hex by default.
smpquery	Enables a basic subset of standard SMP queries including the following: node info, node description, switch info, port info. Fields are displayed in human readable format.
perfquery	Dumps (and optionally clears) the performance counters of the destination port (including error counters).
ibswitches	Scans the net or uses existing net topology file and lists all switches.

Command	Description
ibhosts	Scans the net or uses existing net topology file and lists all hosts.
ibnodes	Scans the net or uses existing net topology file and lists all nodes.
ibportstate	Gets the logical and physical port states of an InfiniBand port or disables or enables the port (only on a switch). Note: This tool can change port settings. Should be used with caution.
saquery	Issues SA queries.
ibdiagnet	ibdiagnet scans the fabric using directed route packets and extracts all the available information regarding its connectivity and devices.
ibnetsplit	Automatically groups hosts and creates scripts that can be run to split the network into sub-networks each containing one group of hosts.
Ibqueryerrors	Queries IB spec-defined errors from all fabric ports. Note: This tool can change reset port counters Should be used with caution.
smparquery	Queries adaptive-routing related settings from a particular switch. Note: This tool can change reset port counters Should be used with caution.

Diagnostic Tools

Model of operation: All utilities use direct MAD access to operate. Operations that require QP 0 mads only, may use direct routed mads, and therefore may work even in subnets that are not configured. Almost all utilities can operate without accessing the SM, unless GUID to lid translation is required.

Dependencies

Multiple port/Multiple CA support:

When no InfiniBand device or port is specified (as shown in the following example for "Local umad parameters"), the tools select the interface port to use by the following criteria:

1. The first InfiniBand ACTIVE port.
2. If not found, the first InfiniBand port that is UP (physical link up).

If a port and/or CA name is specified, the **tool** attempts to fulfill the user's request and will fail if it is not possible.

For example:

```
ibaddr          # use the 'best port'
ibaddr -C mthca1        # pick the best port from mthca1 only.
ibaddr -P 2             # use the second (active/up) port from
the first available IB device.
ibaddr -C mthca0 -P 2   # use the specified port only.
```

Common Options & Flags

Most diagnostics take the following flags. The exact list of supported flags per utility can be found in the usage message and can be shown using util_name -h syntax.

```
# Debugging flags
-d   raise the IB debugging level. May be used several times (-
ddd or -d -d -d).
-e   show umad send receive errors (timeouts and others)
-h   show the usage message
-v   increase the application verbosity level.
     May be used several times (-vv or -v -v -v)
-V   show the internal version info.
```

```
# Addressing flags
-D          use directed path address arguments.
            The path is a comma separated list of out ports.
            Examples:
            "0"   # self port
            "0,1,2,1,4"   # out via port 1, then 2, ...
-G          use GUID address arguments.
            In most cases, it is the Port GUID.
            Examples:
            "0x08f1040023"
-s <smlid>  use 'smlid' as the target lid for SA queries.
```

```
# Local umad parameters:
-C <ca_name>     use the specified ca_name.
-P <ca_port>     use the specified ca_port.
-t <timeout_ms>  override the default timeout for the
                 solicited mads.
```

CLI notation: all utilities use the POSIX style notation, meaning that all options (flags) must precede all arguments (parameters).

Utilities Descriptions

ibstatus

A script that displays basic information obtained from the local InfiniBand driver. Output includes LID, SMLID, port state, link width active, and port physical state.

Syntax

```
ibstatus [-h] [devname[:port]]
```

Examples:

```
ibstatus                      # display status of all IB ports
ibstatus mthca1               # status of mthca1 ports
ibstatus mthca1:1 mthca0:2    # show status of specified ports
```

See also: ibstat

ibstat

Similar to the ibstatus utility but implemented as a binary and not as a script. Includes options to list CAs and/or ports.

Syntax

```
ibstat [-d(ebug) -l(ist_of_cas) -p(ort_list) -s(hort)] <ca_name>
[portnum]
```

Examples:

```
ibstat            # display status of all IB ports
ibstat mthca1     # status of mthca1 ports
ibstat mthca1 2   # show status of specified ports
ibstat -p mthca0  # list the port guids of mthca0
ibstat –l         # list all CA names
```

See also: ibstatus

ibroute

Uses SMPs to display the forwarding tables (unicast (LinearForwardingTable or LFT) or multicast (MulticastForwardingTable or MFT)) for the specified switch LID and the optional lid (mlid) range. The default range is all valid entries in the range 1...FDBTop.

Syntax

```
ibroute [options] <switch_addr> [<startlid> [<endlid>]]
```

Nonstandard flags:

```
-a              show all lids in range, even invalid entries.
-n              do not try to resolve destinations.
-M              show multicast forwarding tables. In this case
the range
                parameters are specifying mlid range.
node-name-map   node name map file
```

Examples:

```
ibroute 2            # dump all valid entries of switch lid 2
ibroute 2 15         # dump entries in the range 15...FDBTop.
ibroute -a 2 10 20   # dump all entries in the range 10..20
ibroute -n 2         # simple format
ibroute -M 2         # show multicast tables
```

See also: ibtracert

ibtracert

Uses SMPs to trace the path from a source GID/LID to a destination GID/LID. Each hop along the path is displayed until the destination is reached or a hop does not respond. By using the -m option, multicast path tracing can be performed between source and destination nodes.

Syntax

```
ibtracert [options] <src-addr> <dest-addr>
```

Nonstandard flags:

```
-n              simple format; don't show additional information.
-m <mlid>       show the multicast trace of the specified mlid.
-f <force>      force
node-name-map   node name map file
```

Examples:

```
ibtracert 2 23          # show trace between lid 2 and 23
ibtracert -m 0xc000 3 5 # show multicast trace between lid 3
and 5 for mcast lid 0xc000.
```

smpquery

Enables a basic subset of standard SMP queries including the following node info, node description, switch info, port info. Fields are displayed in human readable format.

Syntax

```
smpquery [options] <op> <dest_addr> [op_params]
```

Currently supported operations and their parameters:

```
nodeinfo <addr>
nodedesc <addr>
portinfo <addr> [<portnum>]   # default port is zero
switchinfo <addr>
pkeys <addr> [<portnum>]
sl2vl <addr> [<portnum>]
vlarb <addr> [<portnum>]
GUIDInfo (GI) <addr>
MlnxExtPortInfo (MEPI) <addr> [<portnum>]
Combined (-c) : use Combined route address argument
node-name-map : node name map file
extended (-x) :  use extended speeds
```

Examples:

```
smpquery nodeinfo 2       # show nodeinfo for lid 2
smpquery portinfo 2 5     # show portinfo for lid 2 port 5
```

smpdump

A general purpose SMP utility that gets SM attributes from a specified SMA. The result is dumped in hex by default.

Syntax

```
smpdump [options] <dest_addr> <attr> [mod]
```

Nonstandard flags:

```
-s       show output as string
```

Examples:

```
smpdump -D 0,1,2 0x15 2       # port info, port 2
smpdump 3 0x15 2              # port info, lid 3 port 2
```

ibaddr

Can be used to show the LID and GID addresses of the specified port or the local port by default. This utility can be used as simple address resolver.

Syntax

```
ibaddr [options] [<dest_addr>]
```

Nonstandard flags:

```
gid_show  (-g) : show gid address only
lid_show   (-l) : show lid range only
Lid_show (-L) : show lid range (in decimal) only
```

Examples:

```
ibaddr                # show local address
ibaddr 2               # show address of the specified port lid
ibaddr -G 0x8f1040023  # show address of the specified port guid
```

sminfo

Issues and dumps the output of an sminfo query in human readable format. The target SM is the one listed in the local port info or the SM specified by the optional SM LID or by the SM direct routed path.

> **⚠ Warning**
>
> CAUTION: Using sminfo for any purpose other than a simple query might result in a malfunction of the target SM.

Syntax

```
sminfo [options] <sm_lid|sm_dr_path> [sminfo_modifier]
```

Nonstandard flags:

```
-s <state>          # use the specified state in sminfo mad
-p <priority>       # use the specified priority in sminfo mad
-a <activity>       # use the specified activity in sminfo mad
```

Examples:

```
sminfo          # show sminfo of SM listed in local portinfo
sminfo 2        # query SM on port lid 2
```

perfquery

Uses PerfMgt GMPs to obtain the PortCounters (basic performance and error counters) from the Performance Management Agent (PMA) at the node specified. Optionally show aggregated counters for all ports of node. Also, optionally, reset after read, or only reset counters.

```
perfquery [options]  [<lid|guid> [[port] [reset_mask]]]
```

Nonstandard flags:

```
-a                      Shows aggregated counters for all ports
of the destination lid.
-r                      Resets counters after read.
-R                      Resets only counters.
Extended (-x)           Shows extended port counters
Xmtsl (-X)              Shows Xmt SL port counters
Rcvsl  ,( -S)           Shows Rcv SL port counters
Xmtdisc  (-D)           Shows Xmt Discard Details
rcverr, (-E)    Shows Rcv Error Details
extended_speeds (-T)    Shows port extended speeds counters
oprcvcounters   Shows Rcv Counters per Op code
flowctlcounters Shows flow control counters
vloppackets     Shows packets received per Op code per VL
vlopdata        Shows data received per Op code per VL
vlxmitflowctlerrors     Shows flow control update errors per VL
vlxmitcounters  Shows ticks waiting to transmit counters per VL
swportvlcong    Shows sw port VL congestion
rcvcc   Shows Rcv congestion control counters
slrcvfecn       Shows SL Rcv FECN counters
slrcvbecn       Shows SL Rcv BECN counters
xmitcc  Shows Xmit congestion control counters
vlxmittimecc    Shows VL Xmit Time congestion control counters
smplctl (-c)    Shows samples control
loop_ports (-l)         Iterates through each port
```

Examples:

```
perfquery               # read local port's performance counters
perfquery 32 1         # read performance counters from lid 32,
port 1
perfquery -a 32        # read from lid 32 aggregated performance
counters
perfquery -r 32 1     # read performance counters from lid 32
port 1 and reset
perfquery -R 32 1     # reset performance counters of lid 32 port
1 only
perfquery -R -a 32    # reset performance counters of all lid 32
ports
perfquery -R 32 2 0xf000  # reset only non-error counters of lid
32 port 2
```

ibping

Uses vendor mads to validate connectivity between InfiniBand nodes. On exit, (IP) ping like output is show. ibping is run as client/server. The default is to run as client. Note also that a default ping server is implemented within the kernel.

Syntax

```
ibping [options] <dest lid|guid>
```

Nonstandard flags:

```
-c <count>      stop after count packets
-f              flood destination: send packets back to back w/o
delay
-o <oui>        use specified OUI number to multiplex vendor MADs
-S              start in server mode (do not return)
```

ibnetdiscover

Performs InfiniBand subnet discovery and outputs a human readable topology file. GUIDs, node types, and port numbers are displayed as well as port LIDs and node descriptions. All nodes (and links) are displayed (full topology). This utility can also be used to list the current connected nodes. The output is printed to the standard output unless a topology file is specified.

Syntax

```
ibnetdiscover [options] [<topology-filename>]
```

Nonstandard flags:

```
l       Lists connected nodes
H       Lists connected HCAs
S       Lists connected switches
g       Groups
full (-f)       Shows full information (ports' speed and width,
vlcap)
show (-s)       Shows more information
Router_list  (-R)       Lists connected routers
node-name-map   Nodes name map file
cache   filename to cache ibnetdiscover data to
load-cache      filename of ibnetdiscover cache to load
diff    filename of ibnetdiscover cache to diff
diffcheck       Specifies checks to execute for --diff
ports : (-p)    Obtains a ports report
max_hops (-m)   Reports max hops discovered by the library
outstanding_smps (-o)   Specifies the number of outstanding SMP's
which should be issued during the scan
```

ibhosts

Traces the InfiniBand subnet topology or uses an already saved topology file to extract the CA nodes.

Syntax

```
ibhosts [-h] [<topology-file>]
```

Dependencies: ibnetdiscover, ibnetdiscover format

ibswitches

Traces the InfiniBand subnet topology or uses an already saved topology file to extract the InfiniBand switches.

Syntax

```
ibswitches [-h] [<topology-file>]
```

Dependencies: ibnetdiscover, ibnetdiscover format

ibportstate

Enables the port state and port physical state of an InfiniBand port to be queried or a switch port to be disabled or enabled.

Syntax

```
ibportstate [-d(ebug) -e(rr_show) -v(erbose) -D(irect) -G(uid) -s
smlid -V(ersion) -C ca_name -P ca_port -t timeout_ms] <dest
dr_path|lid|guid> <portnum> [<op>]
```

Supported ops: enable, disable, query, on, off, reset, speed, espeed, fdr10, width, down, arm, active, vls, mtu, lid, smlid, lmc, mkey, mkeylease, mkeyprot

Examples:

```
ibportstate 3 1 disable                 # by lid
ibportstate -G 0x2C9000100D051 1 enable # by guid
ibportstate -D 0 1                      # by direct route
```

ibnodes

Uses the current InfiniBand subnet topology or an already saved topology file and extracts the InfiniBand nodes (CAs and switches).

Syntax

```
ibnodes [<topology-file>]
```

Dependencies: ibnetdiscover, ibnetdiscover format

ibqueryerrors

Queries or clears the PMA error counters in PortCounters by walking the InfiniBand subnet topology.

```
ibqueryerrors [options]
```

Syntax

```
Options:
  --suppress, -s <err1,err2,...>  suppress errors listed
  --suppress-common, -c   suppress some of the common counters
  --node-name-map <file>  node name map file
  --port-guid, -G <port_guid>  report the node containing the
port
                              specified by <port_guid>
  --, -S <port_guid>      Same as "-G" for backward compatibility
  --Direct, -D <dr_path>  report the node containing the port
specified
                          by <dr_path>
  --skip-sl               don't obtain SL to all destinations
  --report-port, -r       report port link information
  --threshold-file <val>  specify an alternate threshold file,
default: /etc/infiniband-diags/error_thresholds
  --GNDN, -R              (This option is obsolete and does
nothing)
  --data                  include data counters for ports with
errors
  --switch                print data for switches only
  --ca                    print data for CA's only
  --router                print data for routers only
  --details               include transmit discard details
  --counters              print data counters only
  --clear-errors, -k      Clear error counters after read
  --clear-counts, -K      Clear data counters after read
  --load-cache <file>     filename of ibnetdiscover cache to load
  --outstanding_smps, -o <val>  specify the number of outstanding
SMP's
                             which should be issued during the
scan
  --config, -z <config>   use config file, default:
/etc/infiniband-diags/ibdiag.conf
  --Ca, -C <ca>           Ca name to use
  --Port, -P <port>       Ca port number to use
```

```
--timeout, -t <ms>      timeout in ms
--m_key, -y <key>       M_Key to use in request
--errors, -e            show send and receive errors
--verbose, -v           increase verbosity level
--debug, -d             raise debug level
--help, -h              help message
--version, -V           show version
```

smparquery

Issues Adaptive routing-related queries to the fabric switch.

Syntax

```
```

```
Supported ops (and aliases, case insensitive):
    ARInfo (ARI) <addr>
    ARGroupTable (ARGT) <addr> [<plft>] [<group_table>]
[<blocknum>]
    ARLFTTable (ARLT) <addr> [<plft>] [<blocknum>]
    PLFTInfo (PLFTI) <addr>
    PLFTDef (PLFTD) <addr> [<blocknum>]
    PLFTMap (PLFTM) <addr> [<plft>] [<control_map>]
    PortSLToPLFTMap (PLFTP) <addr> [<blocknum>]
    RNSubGroupDirectionTable (DIRT) <addr> [<blocknum>]
    RNGenStringTable (GSTR) <addr> [<plft>] [<blocknum>]
    RNGenBySubGroupPriority (GSGP) <addr>
    RNRcvString (RSTR) <addr> [<blocknum>]
    RNXmitPortMask (RNXM) <addr> [<blocknum>]
    PortRNCounters (RNPC) <addr>


Options:
    Main
        -C|--Ca <ca>                    : Ca name to use
        -P|--Port <port>                : Ca port number to use
        -D|--Direct                     : use Direct address
argument
        -L|--Lid                        : use LID address argument
        -h|--help                       : help message
        -V|--version                    : show version
        -d|--debug                      : Print debug logs
```

saquery

Issues SA queries.

Syntax

```
saquery [-h -d -P -N -L -G -s -g][<name>]
```

Queries node records by default.

dPNL (-L)G (-G)S (-S)G (-g)L (-l)O (-O)m(-m)x (-x)c (-c)S (-S)l (-I)list (-D)src-to-dst (<src:dst>)sgid-to-dgid (<sgid-dgid>)node-name-map smkey <val>slid <lid>dlid <lid>mild <lid>sgid <gid>dgid <gid>gid <gid>mgid <gid>Reversible", 'r', 1, NULL"numb_path ", 'n', 1, NULL"pkey: P_Key (PathRecord, MCMemberRecord).qos_class (-Q)slmtu : (-M)rate (-R)pkt_lifetime qkey (-q) (PathRecord, MCMemberRecord).tclass (-T)flow_label : (-F) hop_limit : (-H) scopejoin_state (-J)proxy_join (-X)service_id	Enables debuggingGets PathRecord infoGets NodeRecord infoReturns just the Lid of the name specifiedReturns just the Guid of the name specifiedReturns the PortInfoRecords with isSM capability mask bit onGets multicast group infoReturns the unique Lid of the name specifiedReturns name for the Lid specifiedGets multicast member info (if multicast group specified, list member GIDs only for group specified for example 'saquery -m 0xC000')Gets LinkRecord info"Gets the SA's class port infoGets ServiceRecord infoGets InformInfoRecord (subscription) infothe node desc of the CA'sGets a PathRecord for <src:dst> where src and dst are either node names or LIDsGets a PathRecord for <sgid-dgid> where sgid and dgid are addresses in IPv6 formatSpecifies a node name map fileSA SM_Key value for the query. If non-numeric value (like 'x') is specified then saquery will prompt for a value. Default (when not specified here or in ibdiag.conf) is to use SM_Key == 0 (or \"untrusted\")Source LID (PathRecord)Destination LID (PathRecord)Multicast LID (MCMemberRecord)Source GID (IPv6 format) (PathRecord)Destination GID (IPv6 format) (PathRecord)Port GID (MCMemberRecord)Multicast GID (MCMemberRecord)Reversible path (PathRecord)Number of paths (PathRecord) QoS Class (PathRecord)Service level (PathRecord, MCMemberRecord)MTU and selector (PathRecord, MCMemberRecord)Rate and selector (PathRecord, MCMemberRecord)Packet lifetime and selector (PathRecord, MCMemberRecord)If non-numeric value (like 'x') is specified then saquery will prompt for a value.Traffic Class (PathRecord, MCMemberRecord)Flow Label (PathRecord, MCMemberRecord)Hop limit (PathRecord, MCMemberRecord)Scope (MCMemberRecord)Join state (MCMemberRecord)Proxy join (MCMemberRecord)ServiceID (PathRecord)

Dependencies: OpenSM libvendor, OpenSM libopensm, libibumad

ibsysstat

```
ibsysstat [options] <dest lid|guid> [<op>]
```

Nonstandard flags:

```
Current supported operations:
  ping - verify connectivity to server (default)
  host - obtain host information from server
  cpu  - obtain cpu information from server
  -o <oui>    use specified OUI number to multiplex vendor mads
  -S          start in server mode (do not return)
```

ibnetsplit

Automatically groups hosts and creates scripts that can be run in order to split the network into sub-networks containing one group of hosts.

Syntax

- Group:

  ```
  ibnetsplit [-v][-h][-g grp-file] -s <.lst|.net|.topo> <-r
  head-ports|-d max-dist>
  ```

- Split:

  ```
  ibnetsplit [-v][-h][-g grp-file] -s <.lst|.net|.topo>
  -o out-dir
  ```

- Combined:

```
ibnetsplit [-v][-h][-g grp-file] -s <.lst|.net|.topo> <-r
head-ports|-d max-dist> -o out-dir
```

Usage

- Grouping:

  The grouping is performed if the -r or -d options are provided.

- If the -r is provided with a file containing group head ports, the algorithm examines the hosts distance from the set of node ports provided in the head-ports file (these are expected to be the ports running standby SM's).

- If the -d is provided with a maximum distance of the hosts in each group, the algorithm partition the hosts by that distance.

> **Note**
>
> This method of analyzation may not be suitable for some topologies.

The results of the identified groups are printed into the file defined by the -g option (default ibnetsplit.groups) and can be manually edited. For groups where the head port is a switch, the group file uses the FIRST host port as the port to run the isolation script from.

- Splitting:

  - If the -o flag is included, this algorithm analyzes the MinHop table of the topology and identifies the set of links and switches that may potentially be used for routing each group ports. The cross-switch links between switches of the group to other switches are declared as split-links and the commands to turn them off using Directed Routes from the original Group Head ports are written into the out-dir provided by the -o flag.

  Both stages require a subnet definition file to be provided by the -s flag. The supported formats for subnet definition are:

- *.net - for ibnetdiscover
- *.lst - for opensm-subnet.lst or ibiagnet.lst
- *.topo - for a topology file

HEAD PORTS FILE

This file is provided by the user and defines the ports by which grouping of the other host ports is defined.

Format:

Each line should contain either the name or the GUID of a single port. For switches the port number shall be 0.

```
<node-name>/P<port-num>|<PGUID>
```

GROUPS FILE

This file is generated by the program if the head-ports file is provided to it. Alternatively it can be provided (or edited) by the user if different grouping is desired. The generated script for isolating or connecting the group should be run from the first node in each group.

Format:

Each line may be either:

```
GROUP: <group name>
<node-name>/P<port-num>|<PGUID>
```

ibdiagnet

ibdiagnet scans the fabric using directed route packets and extracts all the available information regarding its connectivity and devices.

It then produces the following files in the output directory (see below):

- "ibdiagnet2.log" - A log file with detailed information.
- "ibdiagnet2.db_csv" - A dump of the internal tool database.
- "ibdiagnet2.lst" - A list of all the nodes, ports and links in the fabric.
- "ibdiagnet2.pm" - A dump of all the nodes PM counters.
- "ibdiagnet2.mlnx_cntrs" - A dump of all the nodes Mellanox diagnostic counters.
- "ibdiagnet2.net_dump" - A dump of all the links and their features.
- "ibdiagnet2.pkey" - A list of all pkeys found in the fabric.
- "ibdiagnet2.aguid" - A list of all alias GUIDs found in the fabric.
- "ibdiagnet2.sm" - A dump of all the SM (state and priority) in the fabric.
- "ibdiagnet2.fdbs" - A dump of unicast forwarding tables of the fabric switches.
- "ibdiagnet2.mcfdbs" - A dump of multicast forwarding tables of the fabric switches.
- "ibdiagnet2.slvl" - A dump of SLVL tables of the fabric switches.
- "ibdiagnet2.nodes_info" - A dump of all the nodes vendor specific general information for nodes who supports it.
- "ibdiagnet2.plft" - A dump of Private LFT Mapping of the fabric switches.
- "ibdiagnet2.ar" - A dump of Adaptive Routing configuration of the fabric switches.
- "ibdiagnet2.vl2vl" - A dump of VL to VL configuration of the fabric switches.

Load plugins from:

/tmp/ibutils2/share/ibdiagnet2.1.1/plugins/

You can specify additional paths to be looked in with "IBDIAGNET_PLUGINS_PATH" env variable.

```
Plugin Name                                   Result     Comment
libibdiagnet_cable_diag_plugin-2.1.1          Succeeded  Plugin loaded
libibdiagnet_phy_diag_plugin-2.1.1            Succeeded  Plugin loaded
```

Syntax

```
[-i|--device <dev-name>] [-p|--port <port-num>]
[-g|--guid <GUID in hex>] [--skip <stage>]
[--skip_plugin <library name>] [--sc]
[--scr] [--pc] [-P|--counter <<PM>=<value>>]
[--pm_pause_time <seconds>] [--ber_test]
[--ber_thresh <value>] [--llr_active_cell <64|128>]
[--extended_speeds <dev-type>] [--pm_per_lane]
[--ls <2.5|5|10|14|25|FDR10|EDR20>]
[--lw <1x|4x|8x|12x>] [--screen_num_errs <num>]
[--smp_window <num>] [--gmp_window <num>]
[--max_hops <max-hops>] [--read_capability <file name>]
[--write_capability <file name>]
[--back_compat_db <version.sub_version>]
[-V|--version] [-h|--help] [-H|--deep_help]
[--virtual] [--mads_timeout <mads-timeout>]
[--mads_retries <mads-retries>] [-m|--map <map-file>]
[--vlr <file>] [-r|--routing] [--r_opt <[vs,][mcast,]>]
[--sa_dump <file>] [-u|--fat_tree]
[--scope <file.guid>] [--exclude_scope <file.guid>]
[-w|--write_topo_file <file name>]
[-t|--topo_file <file>] [--out_ibnl_dir <directory>]
[-o|--output_path <directory>]
Cable Diagnostic (Plugin)
[--get_cable_info] [--cable_info_disconnected]
Phy Diagnostic (Plugin)
[--get_phy_info] [--reset_phy_info]
```

Options

```
-i|--device <dev-name>          : Specifies the name of the device
of the port
                                 used to connect to the IB fabric
(in case
                                 of multiple devices on he local
system).
-p|--port <port-num>            : Specifies the local device's port
number
                                 used to connect to the IB fabric.
-g|--guid <GUID in hex>         : Specifies the local port GUID
value of the
                                 port used to connect to the IB
fabric. If
                                 GUID given is 0 than ibdiagnet
displays
                                 a list of possible port GUIDs and
waits
                                 for user input.
--skip <stage>                  : Skip the executions of the given
stage.
                                 Applicable skip stages
(vs_cap_smp
                                 vs_cap_gmp | links | pm |
                                 speed_width_check | all).
--skip_plugin <library name>    : Skip the load of the given
library name.
                                 Applicable skip plugins:
                                 (libibdiagnet_cable_diag_plugin-
2.1.1 |
                                  libibdiagnet_phy_diag_plugin-
2.1.1).
--sc                             : Provides a report of Mellanox
counters
--scr                            : Reset all the Mellanox
counters (if –sc
```

```
                                    option selected).
--pc                               : Reset all the fabric PM
counters.
-P|--counter <<PM>=<value>>        : If any of the provided PM is
greater then
                                    its provided value than print
it.
--pm_pause_time <seconds>          : Specifies the seconds to wait
between
                                    first counters sample and
second counters
                                    sample. If seconds given is 0
than no
                                    second counters sample will be
done.
                                    (default=1).
--ber_test                         :Provides a BER test for each
port.
                                    Calculate BER for each port
and check no
                                    BER value has exceeds the BER
threshold.
                                    (default threshold="10^-12").
--ber_thresh <value>               :Specifies the threshold value
for the
                                    BER test. The reciprocal
number of the
                                    BER should be provided.
Example: for
                                    10^-12 than value need to be
                                    1000000000000 or 0xe8d4a51000
                                    (10^12).If threshold given is
0 than all
                                    BER values for all ports will
be
                                    reported.
```

```
--llr_active_cell <64|128>          : Specifies the LLR active
cell size
                                      for BER test, when LLR is
active in the
                                      fabric.
--extended_speeds <dev-type>        : Collect and test port
extended speeds
                                      counters. dev-type: (sw |
all).
--pm_per_lane                       : List all counters per lane
(when
                                      available).
--ls <0|2.5|5|10|14|25|50|100|FDR10>    : Specifies the expected
link speed.
--lw <1x|4x|8x|12x>                 : Specifies the expected
link width.
--screen_num_errs <num>             : Specifies the threshold
for printing
                                      errors to screen.
(default=5).
--smp_window <num>                  : Max smp MADs on wire.
(default=8).
--gmp_window <num>                  : Max gmp MADs on wire.
(default=128).
--max_hops <max-hops>               : Specifies the maximum hops
for the
                                      discovery process.
(default=64).
--read_capability <file name>       : Specifies capability masks
                                      configuration file, giving
capability
                                      mask configuration for the
fabric.
                                      ibdiagnet will use this
mapping for
```

```
                                        Vendor Specific MADs
sending.
--write_capability <file name>       : Write out an example file
for
                                        capability masks
configuration,
                                        and also the default
capability
                                        masks for some devices.
--back_compat_db <version.sub_version> : Show ports section in
                                         "ibdiagnet2.db_csv"
according to
                                         given version. Default
version 2.0.
-V|--version                         : Prints the version of the
tool.
-h|--help                            : Prints help information
(without
                                         plugins help if exists).
-H|--deep_help                       : Prints deep help
information
                                         (including plugins help).
--virtual                            : Discover VPorts during
discovery
                                         stage.
--mads_timeout <mads-timeout>        : Specifies the timeout (in
                                          milliseconds) for sent
and received
                                          mads. (default=500).
--mads_retries <mads-retries>        : Specifies the number of
retreis for
                                         every timeout mad.
(default=2).
-m|--map <map-file>                  : Specifies mapping file,
that maps
                                         node guid to name
```

```
                                        (format: 0x[0-9a-fA-F]+
"name").
                                        Maping file can also be
specified by
                                        Environment variable

"IBUTILS_NODE_NAME_MAP_FILE_PATH".
--src_lid <src-lid>                   : source lid
--dest_lid <dest-lid>                 : destination lid
--dr_path <dr-path>                   : direct route path
-o|--output_path <directory>          : Specifies the directory
where the
                                        Output files will be
placed.

(default="/var/tmp/ibdiagpath/").
Cable Diagnostic (Plugin)
--get_cable_info                      : Indicates to query all
QSFP cables
                                        for cable information.
Cable
                                        information will be
stored
                                        in "ibdiagnet2.cables".
--cable_info_disconnected             : Get cable info on
disconnected
                                        ports.
Phy Diagnostic (Plugin)
--get_phy_info                        : Indicates to query all
ports for phy
                                        information.
--reset_phy_info                      : Indicates to clear all
ports phy
                                        information.
```

ibdiagpath

ibdiagpath scans the fabric using directed route packets and extracts all the available information regarding its connectivity and devices. It then produces the following files in the output directory (see below):

- "ibdiagnet2.log" - A log file with detailed information.
- "ibdiagnet2.db_csv" - A dump of the internal tool database.
- "ibdiagnet2.lst" - A list of all the nodes, ports and links in the fabric.
- "ibdiagnet2.pm" - A dump of all the nodes PM counters.
- "ibdiagnet2.mlnx_cntrs" - A dump of all the nodes Mellanox diagnostic counters.
- "ibdiagnet2.net_dump" - A dump of all the links and their features.

Cable Diagnostic (Plugin):

This plugin performs cable diagnostic. It can collect cable info (vendor, PN, OUI etc..) on each valid QSFP cable, if specified.

It produces the following files in the output directory (see below):

- "ibdiagnet2.cables" - In case specified to collect cable info, this file will contain all collected cable info.

Phy Diagnostic (Plugin)

This plugin performs phy diagnostic.

Load Plugins from**:**

```
/tmp/ibutils2/share/ibdiagnet2.1.1/plugins/
```

You can specify additional paths to be looked in with "IBDIAGNET_PLUGINS_PATH" env variableLoad plugins from:

```
Plugin Name                                   Result     Comment
libibdiagnet_cable_diag_plugin-2.1.1          Succeeded  Plugin
loaded
libibdiagnet_phy_diag_plugin-2.1.1            Succeeded  Plugin
loaded
```

Syntax

```
[-i|--device <dev-name>] [-p|--port <port-num>]
[-g|--guid <GUID in hex>] [--skip <stage>]
[--skip_plugin <library name>] [--sc]
[--scr] [--pc] [-P|--counter <<PM>=<value>>]
[--pm_pause_time <seconds>] [--ber_test]
[--ber_thresh <value>] [--llr_active_cell <64|128>]
[--extended_speeds <dev-type>] [--pm_per_lane]
[--ls <2.5|5|10|14|25|FDR10|EDR20>]
[--lw <1x|4x|8x|12x>] [--screen_num_errs <num>]
[--smp_window <num>] [--gmp_window <num>]
[--max_hops <max-hops>] [--read_capability <file name>]
[--write_capability <file name>]
[--back_compat_db <version.sub_version>]
[-V|--version] [-h|--help] [-H|--deep_help]
[--virtual] [--mads_timeout <mads-timeout>]
[--mads_retries <mads-retries>] [-m|--map <map-file>]
[--src_lid <src-lid>] [--dest_lid <dest-lid>]
[--dr_path <dr-path>] [-o|--output_path <directory>]
Cable Diagnostic (Plugin)
[--get_cable_info] [--cable_info_disconnected]
Phy Diagnostic (Plugin)
[--get_phy_info] [--reset_phy_info]
```

Options

-i|--device <dev-name>-p|--port <port-num>-g|--guid <GUID in hex>--skip <stage>--skip_plugin <library name>--sc--scr--pc-P|--counter <<PM>=<value>>--pm_pause_time <seconds>--ber_test--ber_thresh <value>--llr_active_cell <64|128>--extended_speeds <dev-type>--pm_per_lane:List all counters per lane (when available).--ls <2.5|5|10|14|25|FDR10|EDR20>--lw <1x|4x|8x|12x>--screen_num_errs <num>--smp_window <num>--gmp_window <num>--max_hops <max-hops>--read_capability <file name>--write_capability <file name>--back_compat_db <version.sub_version>-V|--version-h|--help-H|--deep_help--virtual--mads_timeout <mads-timeout>--mads_retries <mads-retries>-m|--map <map-file>--src_lid <src-lid>--dest_lid <dest-lid>--dr_path <dr-path>-o|--output_path <directory>Cable Diagnostic (Plugin)--get_cable_info--cable_info_disconnectedPhy Diagnostic (Plugin)--get_phy_info--reset_phy_info	:Specifies the name of the device of the port used to connect to the IB fabric (in case of multiple devices on the local system).:Specifies the local device's port number used to connect to the IB fabric.:Specifies the local port GUID value of the port used to connect to the IB fabric. If GUID given is 0 than ibdiagnet displays a list of possible port GUIDs and waits for user input.:Skip the executions of the given stage. Applicable skip stages: (vs_cap_smp | vs_cap_gmp | links | pm | speed_width_check | all).:Skip the load of the given library name. Applicable skip plugins:(libibdiagnet_cable_diag_plugin-2.1.1 | libibdiagnet_phy_diag_plugin-2.1.1).:Provides a report of Mellanox counters:Reset all the Mellanox counters (if -sc option selected). :Reset all the fabric PM counters. :If any of the provided PM is greater then its provided value than print it.:Specifies the seconds to wait between first counters sample and second counters sample. If seconds given is 0 than no second counters sample will be done. (default=1).:Provides a BER test for each port. Calculate BER for each port and check no BER value has exceeds the BER threshold.(default threshold="10^-12").:Specifies the threshold value for the BER test. The reciprocal number of the BER should be provided. Example: for 10^-12 than value need to be 1000000000000 or 0xe8d4a51000(10^12).If threshold given is 0 than all BER values for all ports will be reported.:Specifies the LLR active cell size for BER test, when LLR is active in the fabric.:Collect and test port extended speeds counters. dev-type: (sw | all). :Specifies the expected link speed.:Specifies the expected link width.:Specifies the threshold for printing errors to screen. (default=5).:Max smp MADs on wire. (default=8).:Max gmp MADs on wire. (default=128).:Specifies the maximum hops for the discovery process.(default=64).:Specifies capability masks configuration file, giving capability mask configuration for the fabric. ibdiagnet will use this mapping for Vendor Specific MADs sending.:Write out an example file for capability masks configuration, and also the default capability masks for some devices.:Show ports section in "ibdiagnet2.db_csv" according to given version. Default version 2.0.:Prints the version of the tool.:Prints help information (without plugins help if exists).:Prints deep help information (including plugins help).:Discover VPorts during discovery stage.:Specifies the timeout (in milliseconds) for sent and received mads.(default=500).:Specifies the number of retries for every timeout mad.(default=2).:Specifies mapping file, that maps node guid to name (format: 0x[0-9a-fA-F]+ "name"). Mapping file can also be specified by environment variable "IBUTILS_NODE_NAME_MAP_FILE_PATH".:source liddestination lid:direct route path:Specifies the directory where the output files will be placed. (default="/var/tmp/ibdiagpath/").

	:Indicates to query all QSFP cables for cable information. Cable information will be stored in "ibdiagnet2.cables".:Get cable info on disconnected ports. :Indicates to query all ports for phy information.:Indicates to clear all ports phy information.

Appendix – SM Partitions.conf File Format

This appendix presents the content and format of the SM *partitions.conf* file.

```
OpenSM Partition configuration
==============================

The default partition will be created by OpenSM unconditionally
even
when partition configuration file does not exist or cannot be
accessed.

The default partition has P_Key value 0x7fff. OpenSM's port will
always
have full membership in default partition. All other end ports
will have
full membership if the partition configuration file is not found
or cannot
be accessed, or limited membership if the file exists and can be
accessed
but there is no rule for the Default partition.

Effectively, this amounts to the same as if one of the following
rules
below appear in the partition configuration file:
In the case of no rule for the Default partition:
Default=0x7fff : ALL=limited, SELF=full ;
In the case of no partition configuration file or file cannot be
accessed:
Default=0x7fff : ALL=full ;


File Format
===========

Comments:

Line content followed after \'#\' character is comment and
ignored by
```

```
parser.

General file format:

<Partition Definition>:[<newline>]<Partition Properties>;

     Partition Definition:
       [PartitionName][=PKey][,ipoib_bc_flags]
[,defmember=full|limited]

        PartitionName  - string, will be used with logging. When
omitted
                         empty string will be used.
        PKey           - P_Key value for this partition. Only low
15 bits will
                         be used. When omitted will be
autogenerated.
        ipoib_bc_flags - used to indicate/specify IPoIB
capability of this partition.

        defmember=full|limited - specifies default membership for
port guid
                        list. Default is limited.

     ipoib_bc_flags:
        ipoib_flag|[mgroup_flag]*

        ipoib_flag - indicates that this partition may be used
for IPoIB, as
                     a result the IPoIB broadcast group will be
created with
                     the flags given, if any.

     Partition Properties:
       [<Port list>|<MCast Group>]* | <Port list>
```

```
     Port list:
        <Port Specifier>[,<Port Specifier>]

     Port Specifier:
        <PortGUID>[=[full|limited]]

        PortGUID         - GUID of partition member EndPort.
Hexadecimal
                           numbers should start from 0x, decimal
numbers
                           are accepted too.
        full or limited  - indicates full or limited membership
for this
                           port.  When omitted (or unrecognized)
limited
                           membership is assumed.

     MCast Group:
        mgid=gid[,mgroup_flag]*<newline>

                    - gid specified is verified to be a Multicast
address
                      IP groups are verified to match the rate
and mtu of the
                      broadcast group.  The P_Key bits of the
mgid for IP
                      groups are verified to either match the
P_Key specified
                      in by "Partition Definition" or if they are
0x0000 the
                      P_Key will be copied into those bits.

     mgroup_flag:
        rate=<val>   - specifies rate for this MC group
                      (default is 3 (10GBps))
        mtu=<val>    - specifies MTU for this MC group
```

```
                        (default is 4 (2048))
        sl=<val>    - specifies SL for this MC group
                        (default is 0)
        scope=<val> - specifies scope for this MC group
                        (default is 2 (link local)).  Multiple
scope settings
                        are permitted for a partition.
                        NOTE: This overwrites the scope nibble of
the specified
                              mgid.  Furthermore specifying
multiple scope
                              settings will result in multiple MC
groups
                              being created.
        qkey=<val>      - specifies the Q_Key for this MC group
                          (default: 0x0b1b for IP groups, 0 for
other groups)
                          WARNING: changing this for the
broadcast group may
                                   break IPoIB on client nodes!!!
        tclass=<val>    - specifies tclass for this MC group
                          (default is 0)
        FlowLabel=<val> - specifies FlowLabel for this MC group
                          (default is 0)

     newline: '\n'


Note that values for rate, mtu, and scope, for both partitions
and multicast
groups, should be specified as defined in the IBTA specification
(for example,
mtu=4 for 2048).

There are several useful keywords for PortGUID definition:
```

```
 - 'ALL' means all end ports in this subnet.
 - 'ALL_CAS' means all Channel Adapter end ports in this subnet.
 - 'ALL_SWITCHES' means all Switch end ports in this subnet.
 - 'ALL_ROUTERS' means all Router end ports in this subnet.
 - 'SELF' means subnet manager's port.

Empty list means no ports in this partition.



Notes:
-----

White space is permitted between delimiters ('=', ',',':',';').

PartitionName does not need to be unique, PKey does need to be
unique.
If PKey is repeated then those partition configurations will be
merged
and first PartitionName will be used (see also next note).

It is possible to split partition configuration in more than one
definition, but then PKey should be explicitly specified
(otherwise
different PKey values will be generated for those definitions).



Examples:
--------

 Default=0x7fff : ALL, SELF=full ;
 Default=0x7fff : ALL, ALL_SWITCHES=full, SELF=full ;

 NewPartition , ipoib : 0x123456=full, 0x3456789034=limited,
0x2134af2306 ;
```

```
 YetAnotherOne = 0x300 : SELF=full ;
 YetAnotherOne = 0x300 : ALL=limited ;

 ShareIO = 0x80 , defmember=full : 0x123451, 0x123452;
 # 0x123453, 0x123454 will be limited
 ShareIO = 0x80 : 0x123453, 0x123454, 0x123455=full;
 # 0x123456, 0x123457 will be limited
 ShareIO = 0x80 : defmember=limited : 0x123456, 0x123457,
0x123458=full;
 ShareIO = 0x80 , defmember=full : 0x123459, 0x12345a;
 ShareIO = 0x80 , defmember=full : 0x12345b, 0x12345c=limited,
0x12345d;

 # multicast groups added to default
 Default=0x7fff,ipoib:
        mgid=ff12:401b::0707,sl=1 # random IPv4 group
        mgid=ff12:601b::16    # MLDv2-capable routers
        mgid=ff12:401b::16    # IGMP
        mgid=ff12:601b::2     # All routers
        mgid=ff12::1,sl=1,Q_Key=0xDEADBEEF,rate=3,mtu=2 # random
group
        ALL=full;


Note:
----

The following rule is equivalent to how OpenSM used to run prior
to the
partition manager:

Default=0x7fff,ipoib:ALL=full;
```

Appendix - Supported Port Counters and Events

Port counters and events are available in the following views:

- Events and Port Counters area, at the bottom of the UFM window
- Error window (Error tab) in the Manage Devices tab
- In the New Monitoring Session window, in the Monitor tab, when clicking Create New Session
- Event Log in the Log tab (click Show Event Log)

InfiniBand Port Counters

The following tables list and describe the port counters and events currently supported:

- InfiniBand Port Counters
- Calculated Port Counters

InfiniBand Port Counters	
Counter	Description
Xmit Data (in bytes)	Total number of data octets, divided by 4, transmitted on all VLs from the port, including all octets between (and not including) the start of packet delimiter and the VCRC, and may include packets containing errors. All link packets are excluded. Results are reported as a multiple of four octets.
Rcv Data (in bytes)	Total number of data octets, divided by 4, received on all VLs at the port. All octets between (and not including) the start of packet delimiter and the VCRC are excluded and may include packets containing errors. All link packets are excluded. When the received packet length exceeds the maximum allowed packet length specified in C7-45: the counter may include all data octets exceeding this limit. Results are reported as a multiple of four octets.
Xmit Packets	Total number of packets transmitted on all VLs from the port, including packets with errors and excluding link packets.

InfiniBand Port Counters	
Rcv Packets	Total number of packets, including packets containing errors and excluding link packets, received from all VLs on the port.
Rcv Errors	Total number of packets containing errors that were received on the port including: • Local physical errors (ICRC, VCRC, LPCRC, and all physical errors that cause entry into the BAD PACKET or BAD PACKET DISCARD states of the packet receiver state machine) • Malformed data packet errors (LVer, length, VL) • Malformed link packet errors (operand, length, VL) • ackets discarded due to buffer overrun (overflow)
Xmit Discards	Total number of outbound packets discarded by the port when the port is down or congested for the following reasons: • Output port is not in the active state • Packet length has exceeded NeighborMTU • Switch Lifetime Limit exceeded • Switch HOQ Lifetime Limit exceeded, including packets discarded while in VLStalled State.
Symbol Errors	Total number of minor link errors detected on one or more physical lanes.
Link Error Recovery	Total number of times the Port Training state machine has successfully completed the link error recovery process.
Link Error Downed	Total number of times the Port Training state machine has failed the link error recovery process and downed the link.
Local Integrity Error	The number of times that the count of local physical errors exceeded the threshold specified by LocalPhyErrors
Rcv Remote Physical Error	Total number of packets marked with the EBP delimiter received on the port.
Xmit Constraint Error	Total number of packets not transmitted from the switch physical port for the following reasons: • FilterRawOutbound is true and packet is raw • PartitionEnforcementOutbound is true and packet fails partition key check or IP version check

InfiniBand Port Counters	
Rcv Constraint Error	Total number of packets received on the switch physical port that are discarded for the following reasons: • FilterRawInbound is true and packet is raw • PartitionEnforcementInbound is true and packet fails partition key check or IP version check
Excess Buffer Overrun Error	The number of times that OverrunErrors consecutive flow control update periods occurred, each having at least one overrun error
Rcv Switch Relay Error	Total number of packets received on the port that were discarded when they could not be forwarded by the switch relay for the following reasons: • DLID mapping • VL mapping • Looping (output port = input port)
VL15 Dropped	Number of incoming VL15 packets dropped because of resource limitations (e.g., lack of buffers) in the port
XmitWait	The number of ticks during which the port selected by PortSelect had data to transmit but no data was sent during the entire tick because of insufficient credits or of lack of arbitration.

InfiniBand Calculated Port Counters	
Counter	Description
Normalized XmitData	Effective port bandwidth utilization in % XmitData incremental/ Link Capacity
Normalized Congested Bandwidth	Amount of bandwidth that was suppressed due to congestion (XmitWait incremental/ Time) * Link CapacitySeparate counters are used for Tier 4 ports and for the rest of the ports.

Supported Traps and Events

Device events are listed as VDM or CDM in the Source column of the Events table in the UFM GUI. For information about defining event policy, see Configuring Event Management.

Alarm ID	Alarm Name	To Log	Alarm	Default Severity	Default Threshold
64	GID Address In Service	1	0	Info	1
65	GID Address Out of Service	1	0	Warning	1
66	New MCast Group Created	1	0	Info	1
67	MCast Group Deleted	1	0	Info	1
110	Symbol Error	1	1	Warning	200
111	Link Error Recovery	1	1	Minor	1
112	Link Downed	1	1	Critical	1
113	Port Receive Errors	1	1	Minor	5
114	Port Receive Remote Physical Errors	0	0	Minor	5
115	Port Receive Switch Relay Errors	1	1	Minor	999
116	Port Xmit Discards	1	1	Minor	200
117	Port Xmit Constraint Errors	1	1	Minor	200
118	Port Receive Constraint Errors	1	1	Minor	200
119	Local Link Integrity Errors	1	1	Minor	5
120	Excessive Buffer Overrun Errors	1	1	Minor	100
121	VL15 Dropped	1	1	Minor	50
122	Congested Bandwidth (%) Threshold Reached	1	1	Minor	10
123	Port Bandwidth (%) Threshold Reached	1	1	Minor	95
130	Non-optimal link width	1	1	Minor	1

Alarm ID	Alarm Name	To Log	Alarm	Default Severity	Default Threshold
134	T4 Port Congested Bandwidth	1	1	Warning	10
141	Flow Control Update Watchdog Timer Expired	1	0	Warning	1
144	Capability Mask Modified	1	0	Info	1
145	System Image GUID changed	1	0	Info	1
156	Link Speed Enforcement Disabled	1	0	Critical	0
250	Running in Limited Mode	1	1	Critical	1
251	Switching to Limited Mode	1	1	Critical	1
252	License Expired	1	1	Warning	1
253	Duplicated licenses	1	0	Critical	1
254	License Limit Exceeded	1	0	Critical	1
255	License is About to Expire	1	0	Warning	1
256	Bad M_Key	1	0	Minor	1
257	Bad P_Key	1	0	Minor	1
258	Bad Q_Key	1	0	Minor	1
259	Bad P_Key Switch External Port	1	0	Critical	1
328	Link is Up	1	0	Info	1
329	Link is Down	1	0	Warning	1
331	Node is Down	1	0	Warning	1
332	Node is Up	1	0	Info	1
336	Port Action Succeeded	1	0	Info	1
337	Port Action Failed	1	0	Minor	1
338	Device Action Succeeded	1	0	Info	1
339	Device Action Failed	1	0	Minor	1
344	Partial Switch ASIC Failure	1	1	Critical	1
370	Gateway Ethernet Link State Changed	1	0	Warning	1

Alarm ID	Alarm Name	To Log	Alarm	Default Severity	Default Threshold
371	Gateway Reregister Event Received	1	0	Warning	1
372	Number of Gateways Changed	1	0	Warning	1
373	Gateway will be Rebooted	1	0	Warning	1
374	Gateway Reloading Finished	1	0	Info	1
380	Switch Upgrade Error	1	1	Critical	1
381	Switch Upgrade Failed	1	0	Info	1
328	Module status NOT PRESENT	1	1	Warning	1
383	Host Upgrade Failed	1	0	Info	1
384	Switch Module Powered Off	1	1	Info	1
385	Switch FW Upgrade Started	1	0	Info	1
386	Switch SW Upgrade Started	1	0	Info	1
387	Switch Upgrade Finished	1	0	Info	1
388	Host FW Upgrade Started	1	0	Info	1
389	Host SW Upgrade Started	1	0	Info	1
391	Switch Module Removed	1	0	Info	1
392	Module Temperature Threshold Reached	1	0	Info	40
393	Switch Module Added	1	0	Info	1
394	Module Status FAULT	1	1	Critical	1
395	Device Action Started	1	0	Info	1
396	Site Action Started	1	0	Info	1
397	Site Action Failed	1	0	Minor	1
398	Switch Chip Added	1	0	Info	1
399	Switch Chip Removed	1	0	Critical	1
403	Device Pending Reboot	1	1	Warning	0

Alarm ID	Alarm Name	To Log	Alarm	Default Severity	Default Threshold
404	System Information is missing	1	1	Warning	1
405	Switch Identity Validation Failed	1	1	Warning	1
406	Switch System Information is missing	1	1	Waring	1
407	COMEX Ambient Temperature Threshold Reached	1	1	Minor	60
408	Switch is Unresponsive	1	1	Critical	1
502	Device Upgrade Finished	1	0	Info	1
506	Device Upgrade Finished	1	0	Info	1
508	Core Dump Created	1	1	Info	1
510	SM Failover	0	1	Critical	1
511	SM State Change	0	1	Info	1
512	SM UP	0	1	Info	1
513	SM System Log Message	0	1	Minor	1
514	SM LID Change	0	1	Warning	1
515	Fabric Health Report Info	1	1	Info	1
516	Fabric Health Report Warning	1	1	Warning	1
517	Fabric Health Report Error	1	1	Critical	1
518	UFM-related process is down	1	1	Critical	1
519	Logs purge failure	1	1	Minor	1
520	Restart of UFM-related process succeeded	1	1	Info	1

Alarm ID	Alarm Name	To Log	Alarm	Default Severity	Default Threshold
521	UFM is being stopped	1	1	Critical	1
522	UFM is being restarted	1	1	Critical	1
523	UFM failover is being attempted	1	1	Info	1
524	UFM cannot connect to DB	1	1	Critical	1
525	Disk utilization threshold reached	1	1	Critical	1
526	Memory utilization threshold reached	1	1	Critical	1
527	CPU utilization threshold reached	1	1	Critical	1
528	Fabric interface is down	1	1	Critical	1
529	UFM standby server problem	1	1	Critical	1
530	SM is down	1	1	Critical	1
531	DRBD Bad Condition	1	1	Critical	1
532	Remote UFM-SM Sync	1	1	Info	1
533	Remote UFM-SM problem	1	1	Critical	1
535	MH Purge Failed	1	1	Warning	1
536	UFM Health Watchdog Info	1	1	Info	1
537	UFM Health Watchdog Critical	1	1	Critical	1
538	Time Diff Between HA Servers	1	1	Warning	1
539	DRBD TCP Connection Performance	1	1	Warning	1
540	Daily Report Completed successfully	1	0	Info	1
541	Daily Report Completed with Error	1	0	Minor	1
542	Daily Report Failed	1	0	Critical	1
543	Daily Report Mail Sent successfully	1	0	Info	1
544	Daily Report Mail Sent Failed	1	0	Minor	1
545	SM is not responding	1	1	Critical	1
560	User Connected				
561	User Disconnected				

Alarm ID	Alarm Name	To Log	Alarm	Default Severity	Default Threshold
602	UFM Server Failover	1	1	Critical	1
603	Events Suppression	1	0	Critical	0
604	Report Succeeded	1	1	Info	1
605	Report Failed	1	1	Critical	1
606	Correction Attempts Paused	1	0	Warning	1
701	Non-optimal Link Speed	1	1	Minor	1
702	Unhealthy IB Port	1	1	Warning	1
703	Fabric Collector Connected	1	0	Info	1
704	Fabric Collector Disconnected	1	1	Critical	1
750	High data retransmission count on port	1	1	Warning	500
901	Fabric Configuration Started	0	1	Info	1
902	Fabric Configuration Completed	0	1	Info	1
903	Fabric Configuration Failed	0	1	Critical	1
904	Device Configuration Failure	0	1	Critical	1
905	Device Configuration Timeout	0	1	Critical	1
906	Provisioning Validation Failure	0	1	Critical	1
907	Switch is Down	1	1	Critical	1
908	Switch is Up	1	1	Info	1
909	Director Switch is Down	1	1	Critical	1
910	Director Switch is Up	1	1	Info	1
911	Module Temperature Low Threshold Reached	1	1	Warning	60
912	Module Temperature High Threshold Reached	1	1	Critical	60

Alarm ID	Alarm Name	To Log	Alarm	Default Severity	Default Threshold
913	Module High Voltage	1	1	Warning	10
914	Module High Current	1	1	Warning	10
915	BER_ERROR	1	1	Critical	1e-8
916	BER_WARNING	1	1	Warning	1e-13
917	SYMBOL_BER_ERROR	1	1	Critical	10
918	High Symbol BER reported	1	1	Warning	10
919	Cable Temperature High	1	1	Critical	0
920	Cable Temperature Low	1	1	Critical	0
1300	SM_SAKEY_VIOLATION	1	1	Warning	
1301	SM_SGID_SPOOFED	1	1	Warning	
1302	SM_RATE_LIMIT_EXCEEDED	1	1	Warning	
1303	SM_MULTICAST_GROUPS_LIMIT_EXCEEDED	1	1	Warning	
1304	SM_SERVICES_LIMIT_EXCEEDED	1	1	Warning	
1305	SM_EVENT_SUBSCRIPTION_LIMIT_EXCEEDED	1	1	Warning	
1306	Unallowed SM was detected in the fabric	1	1	Warning	0
1307	SMInfo SET request was received from unallowed SM	1	1	Warning	0
1309	SM was detected with non-matching SMKey	1	1	Warning	0
1310	Duplicated node GUID was detected	1	1	Critical	1
1311	Duplicated port GUID was detected	1	1	Critical	1
1312	Switch was Rebooted	1	1	Info	1
1315	Topo Config File Error	1	1	Critical	1
1316	Topo Config Subnet Mismatch	1	1	Critical	1

Alarm ID	Alarm Name	To Log	Alarm	Default Severity	Default Threshold
1400	High Ambient Temperature	1	1	Warning	0
1401	High Fluid Temperature	1	1	Warning	0
1402	Low Fluid Level	1	1	Warning	0
1403	Low Supply Pressure	1	1	Warning	0
1404	High Supply Pressure	1	1	Warning	0
1405	Low Return Pressure	1	1	Warning	0
1406	High Return Pressure	1	1	Warning	0
1407	High Differential Pressure	1	1	Warning	0
1408	Low Differential Pressure	1	1	Warning	0
1409	System Fail Safe	1	1	Warning	0
1410	Fault Critical	1	1	Critical	0
1411	Fault Pump1	1	1	Critical	0
1412	Fault Pump2	1	1	Critical	0
1413	Fault Fluid Level Critical	1	1	Critical	0
1414	Fault Fluid Over Temperature	1	1	Critical	0
1415	Fault Primary DC	1	1	Critical	0
1416	Fault Redundant DC	1	1	Critical	0
1417	Fault Fluid Leak	1	1	Critical	0
1418	Fault Sensor Failure	1	1	Critical	0
1419	Cooling Device Monitoring Error	1	0	Critical	0
1420	Cooling Device Communication Error	1	1	Critical	0
1500	New cable detected	1	0	Info	1
1502	Cable detected in a new location	1	0	Warning	1
1503	Duplicate Cable Detected	1	0	Critical	1
1315	Topo Config File Error	1	1	Critical	1
1504	SHARP Allocation Succeeded	1	1	Info	1

Alarm ID	Alarm Name	To Log	Alarm	Default Severity	Default Threshold
1505	SHARP Allocation Failed	1	0	Warning	1
1506	SHARP Deallocation Succeeded	1	0	Info	1
1507	SHARP Deallocation Failed	1	0	Warning	1
1508	Device Collect System Dump Started	1	0	Info	1
1509	Device Collect System Dump Finished	1	0	Info	1
1510	Device Collect System Dump Error	1	0	Critical	1
1511	Virtual Port Added	1	0	Info	1
1512	Virtual Port Removed	1	0	Warning	1
1513	Burn Cables Transceivers Started	1	0	Info	1
1514	Burn Cables Transceivers Finished	1	0	Info	1
1515	Burn Cables Transceivers Failed	1	0	Warning	1
1516	Activate Cables Transceivers FW Finished	1	0	Info	1
1517	Activate Cables Transceivers FW Failed	1	0	Warning	1
1520	Aggregation Node Discovery Failed	1	0	Critical	1
1521	Job Started	1	0	Info	1
1522	Job Ended	1	0	Info	1
1523	Job Start Failed	1	0	Critical	1
1524	Job Error	1	0	Critical	1
1525	Trap QP Error	1	0	Critical	1
1526	Trap Invalid Request	1	0	Critical	1
1527	Trap Sharp Error	1	0	Critical	1

Alarm ID	Alarm Name	To Log	Alarm	Default Severity	Default Threshold
1528	Trap QP Alloc timeout	1	0	Critical	1
1529	Trap AMKey Violation	1	0	Critical	1
1530	Unsupported Trap	1	0	Critical	1
1531	Reservation Updated	1	0	Info	1
1532	Sharp is not Responding	1	0	Critical	1
1533	Agg Node Active	1	0	Info	1
1534	Agg Node Inactive	1	0	Warning	1
1535	Trap AMKey Violation Triggered by AM	1	0	Warning	1
1550	Guids Were Added to Pkey	1	0	Info	1
1551	Guids Were Removed from Pkey	1	0	Info	1
1600	VS/CC Classes Key Violation				
1602	PCI Speed Degradation Warning	1	1	Warning	1
1603	PCI Width Degradation Warning	1	1	Warning	1

Appendix – Used Ports

The following is the list of ports used by the UFM Server for internal and external communication:

Port	Purpose
80(tcp), 443(tcp)	Used by WS clients (Apache Web Server)
8000(udp)	Used for UFM server listening for REST API requests (redirected by Apache web server)
6306(udp)	Used for Multicast requests – communication with latest UFM Agents
8005(udp)	Used as UFM monitoring listening port
8089(tcp)	Used for internal communication between UFM server and MonitoirngHistoryEngine
8888(tcp)	Used by DRBD – communication between UFM Primary and Standby server
15800(tcp)	Used for communication with legacy UFM Agents on Mellanox Grid Director DDR switches
8081(tcp), 8082(tcp)	Used for internal communication with Subnet Manager

Appendix – Configuration Files Auditing

The main purpose of this feature is to allow users to track changes made to selected configuration files. When activating the feature, all the changes are reflected in specific log files which contain information about the changes and when they took place.

To activate this feature:

In *TrackConfig* section in gv.cfg, file value of *track_config* key should be set to **true** and value of *track_conf_files* key should contain a comma-separated list of defined conf files to be tracked.

By default – ALL conf-files are tracked. To activate the feature, after *track_config* key is set to true, the UFM server should be restarted.

Example:

```
[TrackConfig]
# track config files changes
track_config = true
# Could be selected options (comaseparated) UFM, SM, SHARP,
Telemetry. Or ALL for all the files.
track_conf_files = ALL
```

The below lists the configuration files that can be tracked:

Conf File Alias	Configuration Files
UFM	/opt/ufm/files/conf/gv.cfg
SM	/opt/ufm/files/conf/opensm/opensm.conf
SHARP	/opt/ufm/files/conf/sharp2/sharp_am.cfg
Telemetry	/opt/ufm/files/conf/telemetry/launch_ibdiagnet_config.ini
ALL	All the above configuration files.

Once the feature is activated and the UFM server is restarted, the UFM generates file which list the changes made in each of the tracked conf files. These files are located in */opt/ufm/files/auditing/* directory and the file naming convention is as follows: original conf file name with audit.log suffix.

Example: For gv.cfg, the name of the changes-tracking file is gv.cfg.audit.log. Changes are stored in auditing files in “linux diff”-like format.

Example:

```
cat /opt/ufm/files/auditing/gv.cfg.audit.log
=== Change occurred at 2022-07-24 07:31:48.679247 ===
---
+++
@@ -45,7 +45,7 @@
mon_mode_discovery_period = 60
check_interface_retry = 5
# The number of times to try if the InfiniBand fabric interface is
down. The duration of each retry is 1 second.
-ibport_check_retries = 90
+ibport_check_retries = 92
ws_address = UNDEFINED
ws_port = 8088
ws_protocol = https
```

Appendix - Managed Switches Configuration Info Persistency

UFM uses a periodic system information-pulling mechanism to query managed switches inventory data. The inventory information is saved in local JSON files for persistency and tracking of managed switches' status.

Upon UFM start up, UFM loads the saved JSON files to present them to the end user via REST API or UFM WEB UI.

After UFM startup is completed, UFM pulls all managed switches data and updates the JSON file and the UFM model periodically (the interval is configurable). In addition, the JSON files are part of UFM system dump.

The following parameters allow configuration of the feature via gv.cfg fie:

```
[SrvMgmt]
# how often UFM should send json requests for sysinfo to switches
(in seconds)
systems_poll = 180
# To create UFM model in large setups might take a lot of time.
# This is an initial delay (in minutes) before starting to pull
sysinfo from switches.
systems_poll_init_timeout = 5
# to avoid sysinfo dump overloading and multiple writing to host
# switches sysinfo will be dumped to disc in json format every
set in this variable
# sysinfo request. If set to 0 - will not be dumped, if set to 1 -
will be dumped every sysinfo request
# this case (as example defined below) dump will be created every
fifth sysinfo request, so if system_poll is 180 sec (3 minutes)
sysinfo dump to the file will e performed every 15 minutes.
sysinfo_dump_interval = 5
# location of the sysinfo dump file (it is in /opt/ufm/files/logs
(it will be part of UFM dump)
sysinfo_dump_file_path = /opt/ufm/files/log/sysinfo.dump
```

Appendix – UFM Migration

Overview

UFM migration enables backup and restores UFM configuration files.

Backup UFM configuration

By default, the following folders (placed in `/opt/ufm/files`) are being backed up:

- conf
- dashboardViews

- licenses
- networkViews
- scripts
- sqlite
- templates/user-defined
- ufmhealth/scripts
- userdata
- users_preferences

> **Note**
>
> The user may also backup the UFM historical telemetry data ("-t" argument).

UFM (Bare Metal)

```
/opt/ufm/scripts/ufm_backup.sh --help
usage: ufm_backup.pyc [-h] [-f BACKUP_FILE] [-t]
```

Optional Arguments

-h	--help	show this help message and exit
-f	--backup-file BACKUP_FILE	full path of zip file to be generated
-t	--telemetry	backup UFM historical telemetry

UFM Docker Container

1. Backup UFM configuration. Run:

```
docker exec ufm /opt/ufm/scripts/ufm_backup.sh
```

2. Copy the backup file from UFM docker container to the host. Run:

```
docker cp ufm:/root/<backup file> <path on host>
```

UFM Appliance

1. Backup UFM configuration. Run:

```
ufm data backup [with-telemetry]
```

2. Upload the backup file to a remote host. Run:

```
ufm data upload <backup file> <upload URL>
```

> **Note**
>
> More details can be found in the log file `/tmp/ufm_backup.log.`

Restore UFM Configuration

> **Note**
>
> All folders which are a part of the UFM backup are restored (filter is done during the backup stage).

UFM Bare Metal

```
/opt/ufm/scripts/ufm_restore.sh --help
usage: ufm_restore.pyc [-h] -f BACKUP_FILE [-u] [-v]
```

Optional Arguments

-h	--help	show this help message and exit
-f BACKUP_FILE	--backup-file BACKUP_FILE	full path of zip file generated by backup script
-u	--upgrade	upgrades the restored UFM files
-v	--verbose	makes the operation more talkative

UFM Docker Container

1. Stop UFM. Run:

   ```
   docker exec ufm /etc/init.d/ufmd stop
   ```

2. Copy the backup file from the host into UFM docker container. Run:

```
docker cp <backup file> ufm:/tmp/<backup file>
```

3. Restore UFM configuration. Run:

```
docker exec ufm /opt/ufm/scripts/ufm_restore.sh -f
/tmp/<backup file> [--upgrade]
```

4. Start UFM. Run:

```
docker exec ufm /etc/init.d/ufmd start
```

UFM Appliance

1. Stop UFM. Run:

```
no ufm start
```

2. Copy the backup file from a remote host into UFM appliance. Run:

```
ufm data fetch <download URL>
```

3. Restore UFM configuration. Run:

```
ufm data restore <backup file>
```

4. Start UFM. Run:

```
ufm start
```

Note

When restoring the UFM configuration from host to a container, the following parameters in `/opt/ufm/files/conf/gv.cfg` may be reset the following:

- fabric_interface
- ufma_interfaces
- mgmt_interface

Note

UFM configuration upgrade during restore is not supported in UFM Appliance GEN2/GEN2.5

More details can be found in the log files `/tmp/ufm_restore.log` and `/tmp/ufm_restore_upgrade.log`

Appendix – IB Router

IB router provides the ability to send traffic between two or more IB subnets thereby potentially expanding the size of the network to over 40k end-ports, enabling separation

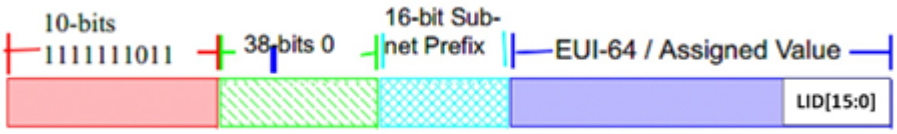
and fault resilience between islands and IB subnets, and enabling connection to different topologies used by different subnets.

The forwarding between the IB subnets is performed using GRH lookup. The IB router's basic functionality includes:

- Removal of current L2 LRH (local routing header)
- Routing table lookup – using GID from GRH
- Building new LRH according to the destination according to the routing table

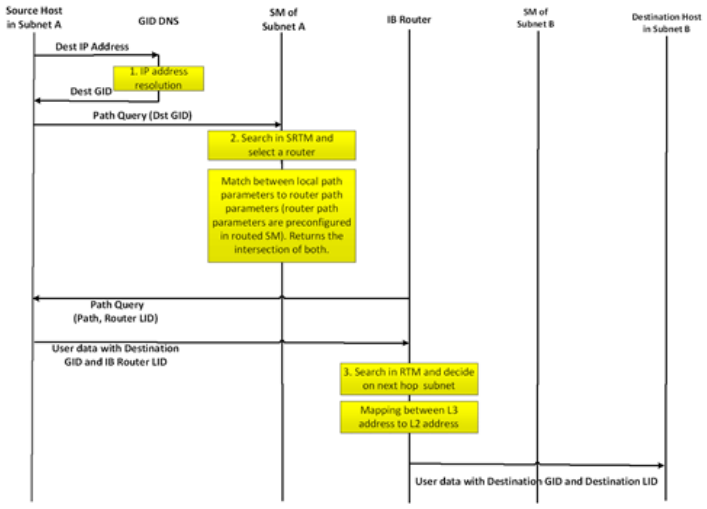
The DLID in the new LRH is built using simplified GID-to-LID mapping (where LID = 16 LSB bits of GID) thereby not requiring to send for ARP query/lookup.

Site-Local Unicast GID Format



For this to work, the SM allocates an alias GID for each host in the fabric where the alias GID = {subnet prefix[127:64], reserved[63:16], LID[15:0}. Hosts should use alias GIDs in order to transmit traffic to peers on remote subnets.

Host-to-Host IB Router Unicast Flow



IB Router Scripts

The following scripts are supplied as part of UFM installation package.

set_num_of_subnets.sh

- **Arguments**

```
/opt/ufm/scripts/ib_router/set_num_of_subnets.sh --hostname
<hostname> --username <username> --password <password> --num-
of-subnets <num-of-subnets>
```

- **Description** – Configures system profile to InfiniBand allowing multiple switch IDs
- **Syntax Description**

hostname	IB router hostname or IP address
username	IB router username
password	IB router user password
num-of-subnets	Specified number of subnets (AKA SWIDs) to be initialized by the system. Value range: 2-6

- **Example**

```
/opt/ufm/scripts/ib_router/set_num_of_subnets.sh --hostname
10.6.204.12 --username admin --password admin --num-of-
subnets 6
```

Note

As a result of running this script, reboot is performed and all configuration is removed

add_interfaces_to_subnet.sh

- **Arguments**

```
/opt/ufm/scripts/ib_router/add_interfaces_to_subnet.sh  --hostname <hostname> --username <username> --password <password> --interface <interface | interface-range> --subnet <subnet>
```

- **Description**

  Maps an interface to a subnet and enables it

- **SyntaxDescription**

hostname	IB router hostname or IP address
username	IB router username
password	IB router user password
interface | interface-range	Single IB interface or range of IB interfaces. Single IB interface: 1/<interface> Range of IB interfaces: 1/<interface>-1/<interface>
subnet	Name of IB subnet (AKA SWID): infiniband-default, infiniband-1…infiniband-5

- **Example**

```
/opt/ufm/scripts/ib_router/add_interfaces_to_subnet.sh  --hostname 10.6.204.12 --username admin --password admin --interface 1/1-1/6 --subnet infiniband-1
```

remove_interfaces_from_subnet.sh

- **Arguments**

```
/opt/ufm/scripts/ib_router/remove_interfaces_from_subnet.sh
--hostname <hostname> --username <username> --password
<password> --interface <interface | interface-range>
```

- **Description**

  Un-maps an interface from a subnet after it has been disabled

- **Syntax Description**

hostname	IB router hostname or IP address
username	IB router username
password	IB router user password
interface | interface-range	Single IB interface or range of IB interfaces. Single IB interface: 1/<interface> Range of IB interfaces: 1/<interface>-1/<interface>

- **Example**

```
/opt/ufm/scripts/ib_router/remove_interfaces_from_subnet.sh
--hostname 10.6.204.12 --username admin --password admin --
interface 1/6Example
```

add_subnet_to_router.sh

- **Arguments**

```
/opt/ufm/scripts/ib_router/add_subnet_to_router.sh --hostname
<hostname> --username <username> --password <password> --
subnet <subnet>
```

- **Description**

  Creates routing on IB subnet interface and enables routing on that interface

- **Syntax Description**

hostname	IB router hostname or IP address
username	IB router username
password	IB router user password
subnet	Name of IB subnet (AKA SWID): infiniband-default, infiniband-1... infiniband-5

- **Example**

```
/opt/ufm/scripts/ib_router/add_subnet_to_router.sh --hostname
10.6.204.12 --username admin --password admin --subnet
infiniband-3Example
```

Note

As a result of running this script, the set of commands that allow control of IB router functionality is being enabled

remove_subnet_from_router.sh

- **Arguments**

```
/opt/ufm/scripts/ib_router/remove_subnet_from_router.sh  --
hostname <hostname> --username <username> --password
<password> --subnet <subnet>
```

- **Description**

  Destroys routing on IB subnet interface after routing on that interface has been disabled

- **Syntax Description**

hostname	IB router hostname or IP address
username	IB router username
password	IB router user password
subnet	Name of IB subnet (AKA SWID): infiniband-default, infiniband-1… infiniband-5

- **Example**

```
/opt/ufm/scripts/ib_router/remove_subnet_from_router.sh  --
hostname 10.6.204.12 --username admin --password admin --
subnet infiniband-defaultExample
```

set_ufm_sm_router_support.sh

- **Arguments**

```
/opt/ufm/scripts/ib_router/set_ufm_sm_router_support.sh [-c
<subnet prefix>] [-r][-h]
```

- **Description**

  [-c <subnet prefix>]: Used for updating OpenSM configuration file with new subnet prefix and forces OpenSM to re-read configuration.

  [-r]: Used for resetting OpenSM configuration to default value and canceling IB routing.

- **Syntax Description**

-c	Configure new IB subnet prefix. Should be followed by new IB router subnet prefix value
-r	Reset to default
-h	Show help

- **Example**

```
/opt/ufm/scripts/ib_router/set_ufm_sm_router_support.sh -c
0xfec0000000001234Examples
```

```
/opt/ufm/scripts/ib_router/set_ufm_sm_router_support.sh -r
```

IB Router Configuration

Step 1: Configure multi-switch. Run:

```
/opt/ufm/scripts/set_num_of_subnets.sh --hostname 10.6.204.12 --
username admin --password admin --num-of-subnets 6
```

Step 2: Map interface to a subnet. Run:

```
/opt/ufm/scripts/add_ports_to_subnet.sh --hostname 10.6.204.12 --
username admin --password admin --interface 1/1 --subnet
infiniband-default
```

Step 3: Create routing on IB subnet interface. Run:

```
/opt/ufm/scripts/add_subnet_to_router.sh --hostname 10.6.204.12
--username admin --password admin --subnet infiniband-default
```

Appendix – NVIDIA SHARP Integration

NVIDIA Scalable Hierarchical Aggregation and Reduction Protocol (SHARP)™

NVIDIA SHARP is a technology that improves the performance of MPI operation by offloading collective operations from the CPU and dispatching to the switch network, and eliminating the need to send data multiple times between endpoints. This approach decreases the amount of data traversing the network as aggregation nodes are reached, and dramatically reduces the MPI operation time.

NVIDIA SHARP software is based on:

- Hardware capabilities in Switch-IB™ 2
- Hierarchical communication algorithms (HCOL) library into which NVIDIA SHARP capabilities are integrated

- NVIDIA SHARP daemons, running on the compute nodes
- NVIDIA SHARP Aggregation Manager, running on UFM

1. These components should be installed from HPCX or MLNX_OFED packages on compute nodes. Installation details can be found in SHARP Deployment Guide.

NVIDIA SHARP Aggregation Manager

Aggregation Manager (AM) is a system management component used for system level configuration and management of the switch-based reduction capabilities. It is used to set up the NVIDIA SHARP trees, and to manage the use of these entities.

AM is responsible for:

- NVIDIA SHARP resource discovery
- Creating topology aware NVIDIA SHARP trees
- Configuring NVIDIA SHARP switch capabilities
- Managing NVIDIA SHARP resources
- Assigning NVIDIA SHARP resource upon request
- Freeing NVIDIA SHARP resources upon job termination

AM is configured by a topology file created by Subnet Manager (SM): subnet.lst. The file includes information about switches and HCAs.

NVIDIA SHARP AM Prerequisites

In order for UFM to run NVIDIA SHARP AM, the following conditions should be met:

- Managed InfiniBand fabric must include at least one of the following Switch-IB 2 switches with minimal firmware version of 15.1300.0126:
    - CS7500
    - CS7510
    - CS7520
    - MSB7790

- MSB7800

- NVIDIA SHARP software capability should be enabled for all Switch-IB 2 switches in the fabric (a dedicated logical port #37, for NVIDIA SHARP packets transmission, should be enabled and should be visible via UFM).

- UFM OpenSM should be running to discover the fabric topology.

NVIDIA SHARP AM is tightly dependent on OpenSM as it uses the topology discovered by OpenSM.

- NVIDIA SHARP AM should be enabled in UFM configuration by running:

```
[Sharp]
sharp_enabled = true
```

NVIDIA SHARP AM Configuration

By default, when running NVIDIA SHARP AM by UFM, there is no need to run further configuration. To modify the configuration of NVIDIA SHAPR AM, you can edit the following NVIDIA SHARP AM configuration file: `/opt/ufm/files/conf/sharp/sharp_am.cfg`.

Running NVIDIA SHARP AM in UFM

To run NVIDIA SHARP AM within UFM, do the following:

1. Make sure that the root GUID configuration file (root_guid.conf) exists in conf/opensm. This file is required for activating NVIDIA SHARP AM.

2. Enable NVIDIA SHARP in conf/opensm/opensm.conf OpenSM configuration file by running "ib sm sharp enable" or by setting the sharp_enabled parameter to 2:

```
# SHArP support
# 0: Ignore SHArP - No SHArP support
# 1: Disable SHArP - Disable SHArP on all supporting switches
# 2: Enable SHArP - Enable SHArP on all supporting switches
sharp_enabled 2
```

3. Make sure that port #6126 (on which NVIDIA SHARP AM is communicating with NVIDIA SHARP daemons) is not being used by any other application. If the port is being used, you can change it by modifying **smx_sock_port** parameter in the NVIDIA SHARP AM configuration file: conf/sharp2/sharp_am.cfg or via the command "ib sharp port".

4. Enable NVIDIA SHARP AM in conf/gv.cfg UFM configuration file by running the command "ib sharp enable" or by setting the sharp_enabled parameter to true (it is false by default):

```
[Sharp]
sharp_enabled = true
```

5. (Optional) Enable NVIDIA SHARP allocation in conf/gv.cfg UFM configuration file by setting the sharp_allocation_enabled parameter to true (it is false by default):

```
[Sharp]
sharp_allocation_enabled = true
```

Note

If the field sharp_enabled, and sharp_allocation_enabled are both set as true in gv.cfg, UFM sends an allocation (reservation) request to NVIDIA SHARP Aggregation Manager (AM) to allocate a list of GUIDs to the specified PKey when a new “Set GUIDs for PKey” REST API is called. If an empty list of GUIDs is sent, a PKEY deallocation request is sent to the SHARP AM.

NVIDIA SHARP allocations (reservations) allow SHARP users to run jobs on top of these resource (port GUID) allocations for the specified PKey. For more information, please refer to the *UFM REST API Guide* under Actions REST API → PKey GUIDs → Set/Update PKey GUIDs.

Operating NVIDIA SHARP AM with UFM

If NVIDIA SHARP AM is enabled, running UFM will run NVIDIA SHARP AM, and stopping UFM will stop NVIDIA SHARP AM.

To

start UFM with NVIDIA SHARP AM (enabled):

```
/etc/init.d/ufmd start
```

The same command applies to HA, using /etc/init.d/ufmha.

Upon startup of UFM or SHARP Aggregation Manager, UFM will resend all existing persistent allocation to SHARP AM.

To stop UFM with NVIDIA SHARP AM (enabled):

```
/etc/init.d/ufmd stop
```

To stop only NVIDIA SHARP AM while leaving UFM running:

```
/etc/init.d/ufmd sharp_stop
```

To start only NVIDIA SHARP AM while UFM is already running:

```
/etc/init.d/ufmd sharp_start
```

Upon startup of UFM or SHARP Aggregation Manager, UFM will resend all existing persistent allocation to SHARP AM.

To restart only NVIDIA SHARP AM while UFM is running:

```
/etc/init.d/ufmd sharp_restart
```

Upon startup of UFM or SHARP Aggregation Manager, UFM will resend all existing persistent allocation to SHARP AM.

To display NVIDIA SHARP AM status while UFM is running:

```
/etc/init.d/ufmd sharp_status
```

Monitoring NVIDIA SHARP AM by UFMHealth

UFMHealth monitors SHARP AM and verifies that NVIDIA SHARP AM is always running. When UFMHealth detects that NVIDIA SHARP AM is down, it will try to re-start it, and will

trigger an event to the UFM to notify it that NVIDIA SHARP AM is down.

Managing NVIDIA SHARP AM by UFM High Availability (HA)

In case of a UFM HA failover or takeover, NVIDIA SHARP AM will be started on the new master node using the same configuration that was used prior to the failover/takeover.

NVIDIA SHARP AM Logs

NVIDIA SHARP AM log file (sharp_am.log) at /opt/ufm/files/log.

NVIDIA SHARP AM log files are rotated by UFM logrotate mechanism.

NVIDIA SHARP AM Version

NVIDIA SHARP AM version can be found at /opt/ufm/sharp/share/doc/SHARP_VERSION.

Appendix – AHX Monitoring

AHX Monitoring is a tool that is used to monitors AHX devices.

Overview

AHX monitoring enables monitoring HDR director switch cooling devices (i.e. AHX) and sends events to UFM.

The events are triggered on the switch associated with the cooling device if the monitoring utility encounters an issue.

The monitoring utility runs periodically and communicates with the AHX devices over the Modbus protocol (TCP port 502).

For deployment and configuration, please refer to the AHX Monitoring plugin in Mellanox Docker HUB.

Appendix - UFM SLURM Integration

Simple Linux Utility for Resource Management (SLURM) is a job scheduler for Linux and Unix-like kernels.

By integrating SLURM with UFM, you can:

- Assign partition keys (PKeys) to SLRUM nodes that are assigned for specific SLURM jobs.
- Create SHARP reservations based on SLURM nodes assigned for specific SLURM jobs.

Prerequisites

- UFM 6.9.0 (or newer) installed on a RedHat 7.x
- Python 2.7 on SLURM controller
- UFM-SLURM integration files (provided independently)

Automatic Installation

A script is provided to install the UFM-SLURM integration automatically.

1. Using the SLURM controller, extract the UFM-SLURM integration tar file:

```
tar -xf ufm_slurm_integration.tar.gz
```

2. Run the installation script using root privileges.

```
sudo ./install.sh
```

Manual Installation

To install the UFM-SLURM integration manually:

1. Extract the UFM-SLURM integration tar file:

```
tar -xf ufm_slurm_integration.tar.gz
```

2. Copy the UFM-SLURM integration files to the SLURM controller folder.

3. Change the permissions of the UFM-SLURM integration files to 755.

4. Modify the SLURM configuration file on the SLURM controller, `/etc/slurm/slurm.conf`, and add/modify the following two parameters:

```
PrologSlurmctld=/etc/slurm/ufm-prolog.sh
EpilogSlurmctld=/etc/slurm/ufm-epilog.sh
```

UFM SLURM Config File

The integration process uses a configuration file located at `/etc/slurm/ufm_slurm.conf`. This file is used to configure settings and attributes for UFM-SLURM integration.

Here are the contents:

Attribute Name	Description	Optionality
ufm_server	IP of UFM server to connect to	Mandatory
auth_type	Should be `token_auth`, or `basic_auth` If you select `basic_auth`, you need to set `ufm_server_user` and `ufm_server_pass` If you select `token_auth`, you need to set `token_auth`	Mandatory
ufm_server_user	Username of UFM server used to connect to UFM, if you set `auth_type=basic_auth`	Mandatory, depends on the auth_type

Attribute Name	Description	Optionality
ufm_server_pass	UFM server user password	Mandatory, depends on the auth_type
token	Generated token when you set uth_typea to `token_auth`	Mandatory, depends on the auth_type
pkey_allocation	By setting `pkey_allocation` to true, UFM SLURM Integration will use static Pkey assignment to create new Pkey, otherwise it will use the default management Pkey 0x7fff	Mandatory, default is True.
pkey	Hexadecimal string between "0x0001"-"0x7ffe" exclusive	Optional, default is "0x7fff" (This is the default management pkey)
ip_over_ib	PKey is a member in a multicast group that uses IP over InfiniBand	Hidden param, default is True
index0	If true, the API will store the PKey at index 0 of the PKey table of the GUID	Hidden param, default is False
sharp_allocation	By setting `sharp_allocation` to true, UFM SLURM Integration will create new SHARP allocation with all SLURM job IDs allocated to hosts	Mandatory, default is False
partially_alloc	By setting this to false, UFM will fail the SHARP allocation request if at least one node does not exist in the fabric	Optional, default is False
app_resources_limit	Application resources limitation	Hidden param, default is -1
log_file_name	Name of integration logging file	Optional

Configuring UFM for NVIDIA SHARP Allocation

To configure UFM for NVIDIA SHARP allocation/deallocation you must set `sharp_enabled` and `enable_sharp_allocation` to true in gv.cfg file.

Generate token_auth

If you set `auth_type=token_auth` in UFM SLURM's config file, you must generate a new token by logging into the UFM server and running the following `curl` command:

```
curl -H "X-Remote-User:admin" -XPOST
http://127.0.0.1:8000/app/tokens
```

Then you must copy the generated token and paste it into the config file beside the `token_auth` parameter.

Prolog and Epilog

After submitting jobs on SLURM, there are two scripts that are automatically executed:

- `ufm-prolog.sh` – the prolog script is executed when a job is submitted and before running the job itself. It creates the partition key (pkey) assignment and/or NVIDIA SHARP reservation and assigns the SLURM job hosts for them.
- `ufm-epilog.sh` – the epilog script is executed when a job is complete. It removes the partition key (PKey) assignment and/or NVIDIA SHARP reservation and free the associated SLURM job hosts.

Integration Files

The integration use scripts and configuration files to work, which should be copied to SLURM controller `/etc"/slurm`. Here is a list of these files:

File Name	Description
ufm-prolog.sh	Bash file which executes jobs related to UFM after the SLURM job is completed
ufm-epilog.sh	Bash file which executes jobs related to UFM before the SLURM job is executed
ufm_slurm.conf	UFM-SLURM integration configuration file

File Name	Description
ufm_slurm_prolog.py	Python script file which creates the partition key (pkey) assignment and/or SHARP reservation when the prolog bash script is running
ufm_slurm_epilog.py	Python script file which removes partition key (pkey) assignment and/or SHARP reservation based on the SLURM job hosts.
ufm_slurm_utils.py	Utility Python file containing functions and utilities used by the integration process

Running UFM-SLURM Integration

Using the SLURM controller, execute the following commands to run your batch job:

```
$ sbatch -N4 slurm_demo.sh
Submitted batch job 1
```

Note

N4 is the number of compute nodes used to run the jobs.
`slurm_demo.sh` is the job batch file to be run.

The output and result are stored on the working directory `slurm-{id}.out` where `{id}` is the ID of the submitted job.

In the above example, after executing `sbatch` command, you can see that the submitted job ID is 1. Therefore, the output file would be stored in slurm-1.out.

Execute the following command to see the output:

```
$cat slurm-1.out
```

On the UFM side, a partition key (PKey) is created in case the `pkey_allocation` parameter is set to true in the configuration file, and the user provided the PKey name including the SLURM job IDs allocated to the hosts. Otherwise it will use the default management PKey.

In addition, the UFM-SLURM will create SHARM AM reservation in case the `sharp_allocation` parameter is set to true in the `ufm_slurm.conf` file.

After the SLURM job is completed, the UFM removes the job-related partition key (PKey) assignment and SHARP reservation, if they were created.

From the moment a job is submitted by the SLURM server until its completion, a log file named `/tmp/ufm_slurm.log` logs all of the actions and errors that occurred during the execution.

This log file can be changed by modifying the `log_file_name` parameter in `/etc/slurm /ufm_slurm.conf`.

Appendix - Switch Grouping

To facilitate the logical grouping of 1U switches into a "director-like switch" group, the UFM implements a special dedicated group of interconnected 1U switches based on a YAML configuration file. This group, which is of type "superswitch", only includes 1U switches connected to each other, with some functioning as lines and others as spines.

To access the configuration file for superswitches, users can define the path in the [SubnetManager] section of the `gv.cfg` file, using the variable name "`super_switch_config_file_path`". For instance, the path can be specified as follows:
`super_switch_config_file_path=/opt/ufm/files/conf/super_switches_conf`
.

It is important to note that the file must be located in the `/opt/ufm/files file` tree, as it should be replicated between master and slave UFM servers in a high-availability configuration.

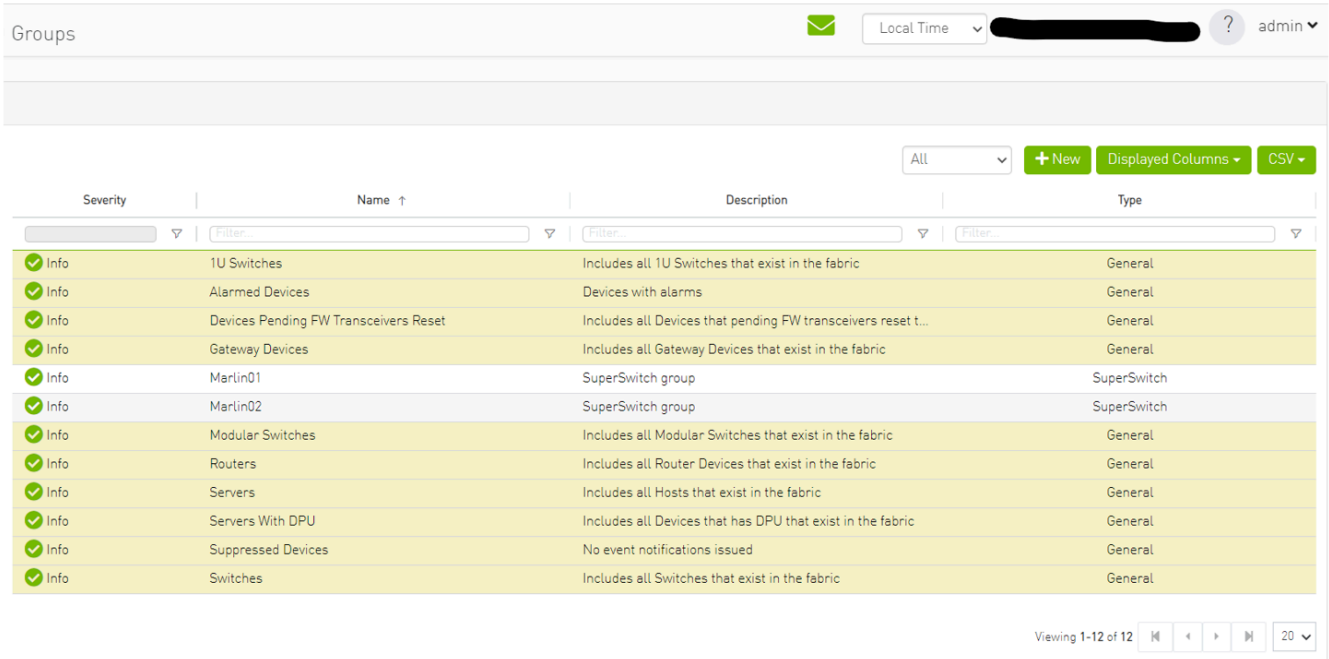
The structure of the superswitch definition should be as follows, based on the following example:

```
superswitch:
  - name: "Marlin01" # Director switch name
    description: "primary dc switch" # Free text with the customer
facing description
    location: "US, NC, DC01" # Director switch location (global
location, includes all racks/switches)
    racks: # Director switch Racks definitions
      #Rack definition
      - name: "rack A" # Director switch rack name
        location:
          dc-grid-row: "A" # formalized rack location in DC
          dc-grid-column: "1" # formalized
          comments: "left-most rack in the line" #Cutomer facing commnent on
the rack
        leafs: # List of Director switch leafs (for the rack
specified)
          - guid: "0x043f720300922a00" #required filed. Switch GUID.
            location-u: 1 # required field. Device location in
rack: "U#"
            description: "MF0;gorilla-01:MQM9700/U1" # optional field.
          - guid: "0x043f720300899cc0" #required filed. Switch GUID.
            location-u: XX # required field. Device location in
rack: "U#"
            description: "MF0;gorilla-01:MQM9700/U2" # optional field.
        spines: # List of Director switch spines (for the rack
specified)
          - guid: "0x043f720900922a00" #required filed. Switch GUID.
            location-u: 10 # required field. Device location in
rack: "U#"
            description: "MF0;gorilla-02:MQM9700/U1" # optional field.
          - guid: "0x043f720900899cc0" #required filed. Switch GUID.
            location-u: XX # required field. Device location in
rack: "U#"
            description: "MF0;gorilla-02:MQM9700/U2" # optional field.
  - name: "Marlin02" # Director switch name
```

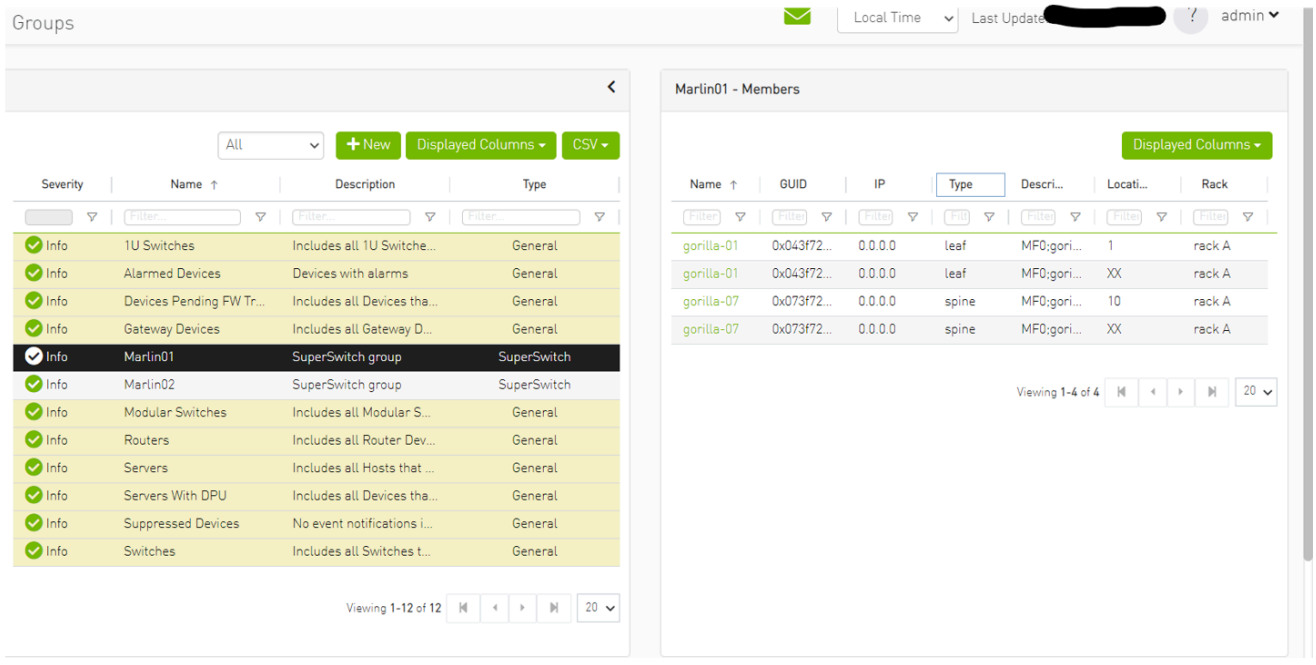
```
    description: "primary dc switch" # Free text with the customer
facing description
    location: "US, NC, DC01" # Director switch location (global
location, includes all racks/switches)
    racks: # Director switch Racks definitions
      #Rack definition
      - name: "rack B" # Director switch rack name
        location:
          dc-grid-row: "B" # formalized rack location in DC
          dc-grid-column: "1" # formalized
          comments: "left-most rack in the line" #Cutomer facing commnent on
the rack
        leafs: # List of Director switch leafs (for the rack
specified)
          - guid: "0x093f720300922a00" #required filed. Switch GUID.
            location-u: 1 # required field. Device location in
rack: "U#"
            description: "MF0;gorilla-03:MQM9700/U1" # optional field.
          - guid: "0x093f720300899cc0" #required filed. Switch GUID.
            location-u: XX # required field. Device location in
rack: "U#"
            description: "MF0;gorilla-03:MQM9700/U2" # optional field.
        spines: # List of Director switch spines (for the rack
specified)
          - guid: "0x093f720900922a00" #required filed. Switch GUID.
            location-u: 10 # required field. Device location in
rack: "U#"
            description: "MF0;gorilla-04:MQM9700/U1" # optional field.
          - guid: "0x093f720900899cc0" #required filed. Switch GUID.
            location-u: XX # required field. Device location in
rack: "U#"
            description: "MF0;gorilla-04:MQM9700/U2" # optional field
```

UI Presentation

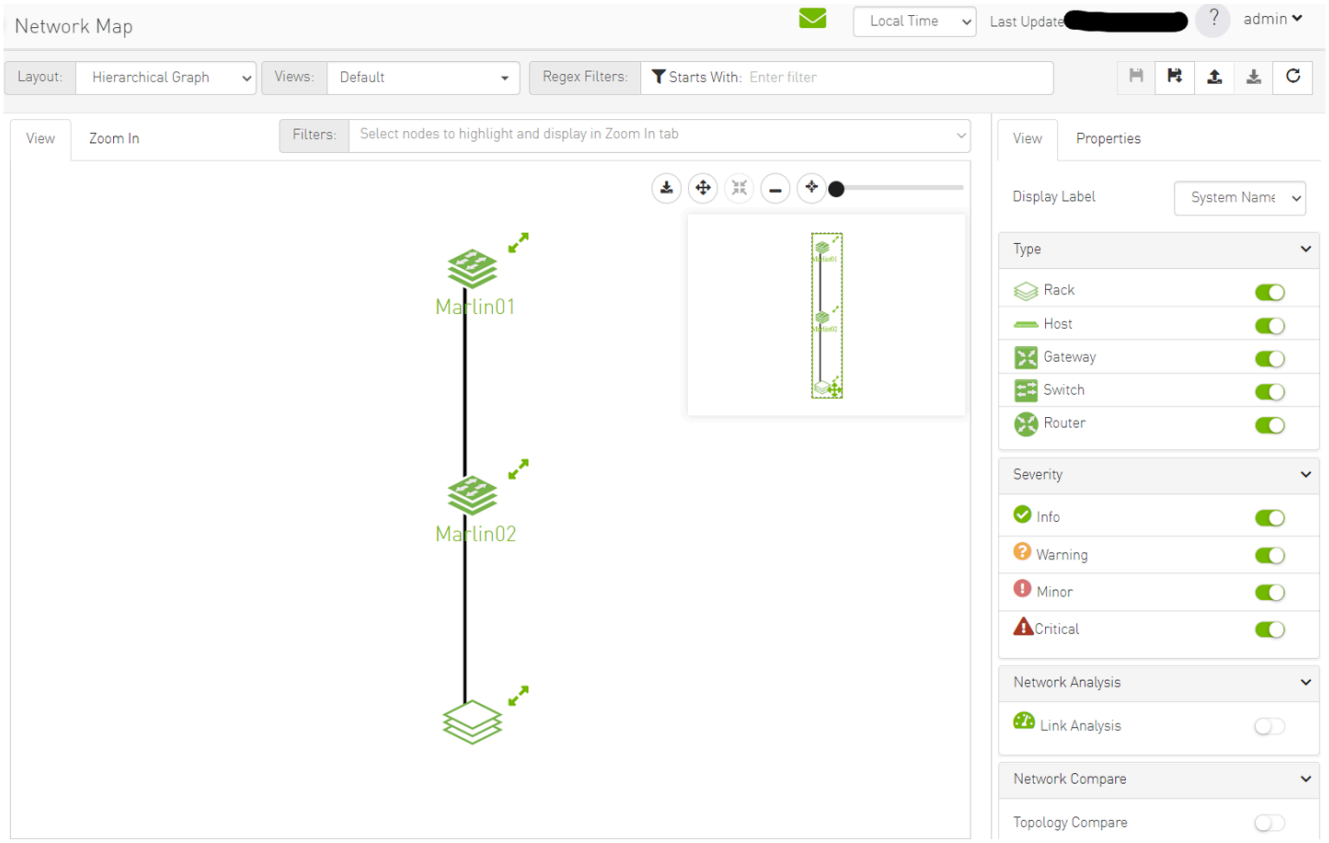
The logical grouping can be accessed under the "Groups" view, specifically listed as "SuperSwitch group" type.



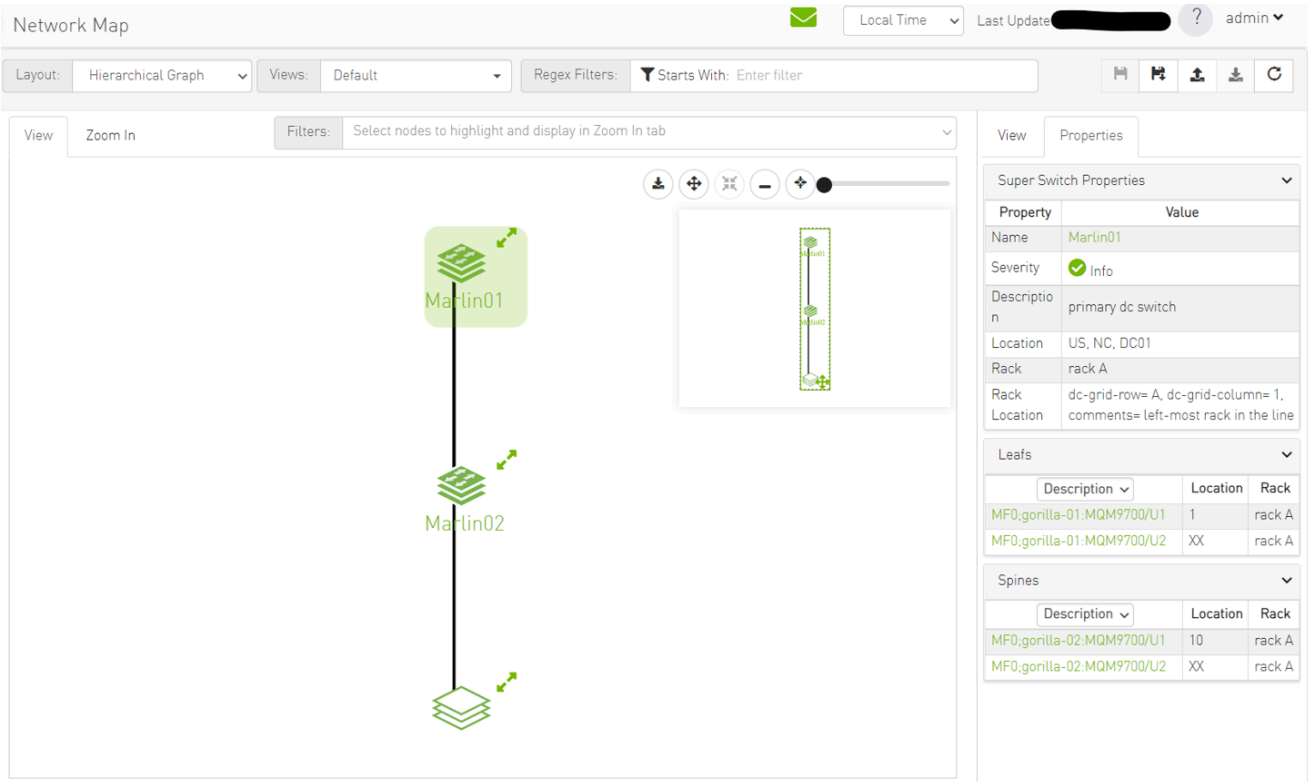
Upon selecting the group type SuperSwitch, additional columns containing information related to the SuperSwitch are added to the details view.



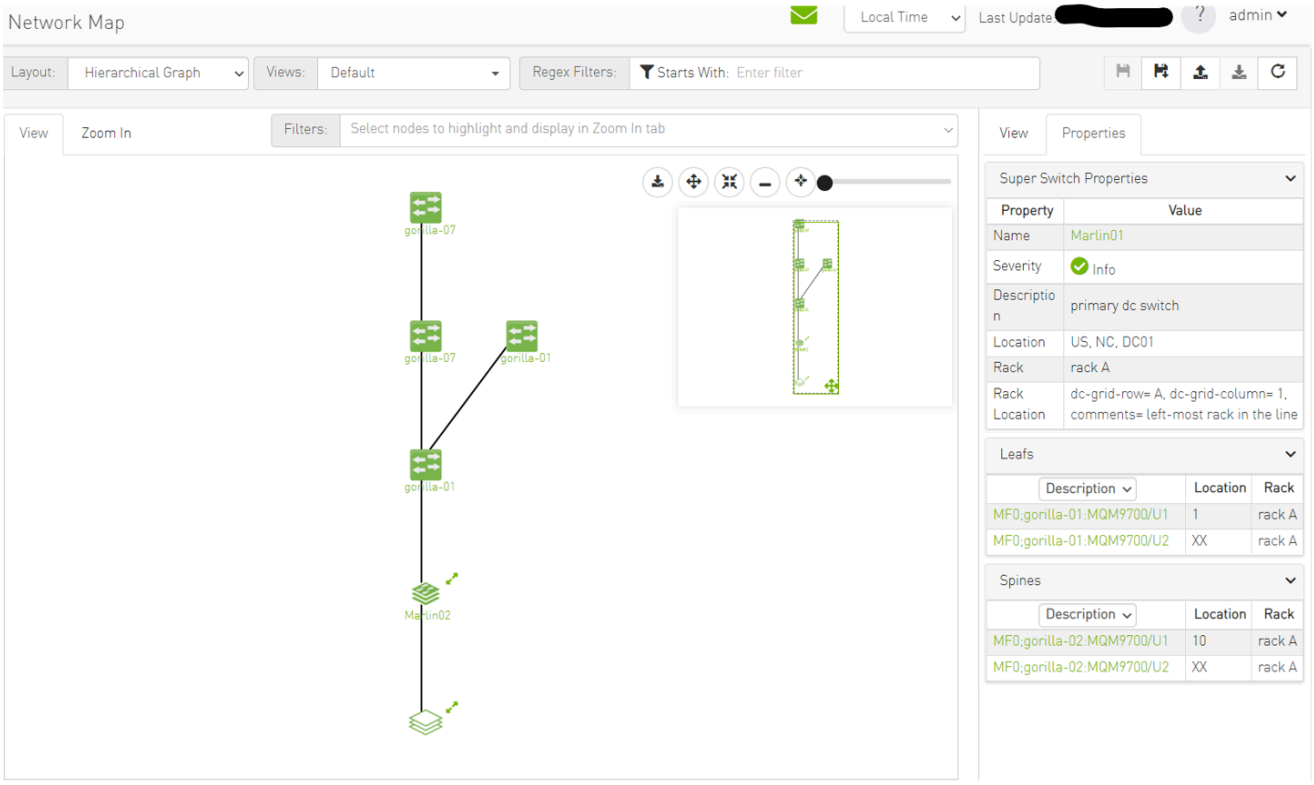
An icon for the SuperSwitch group in its collapsed view exists on the network map.

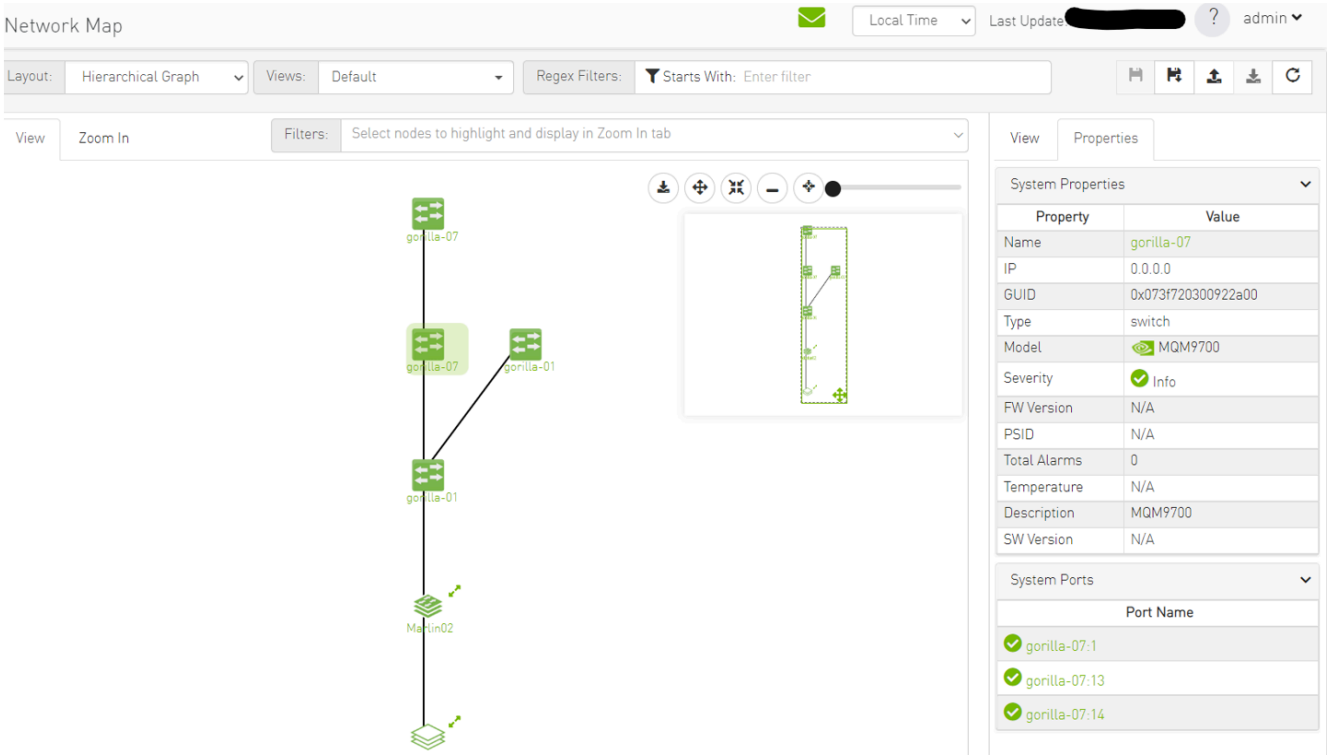


Upon selecting the SuperSwitch group, all of its properties can be viewed in the details view.

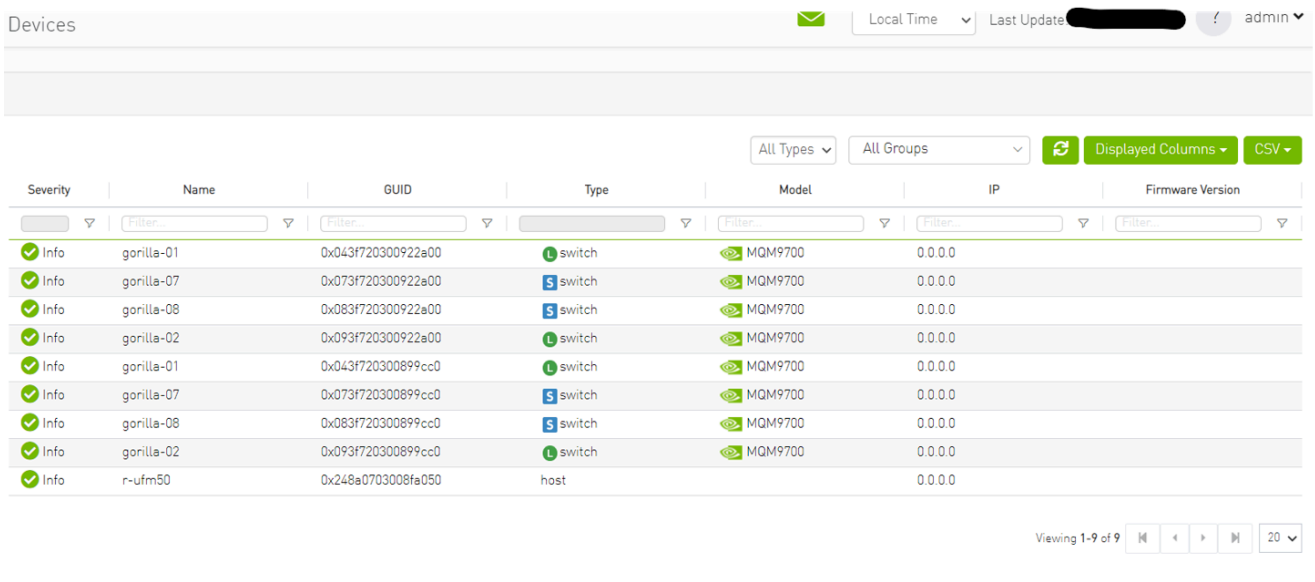


Expanding the SuperSwitch group icon displays all the switches included in the group as separate 1U switches, along with their respective properties.

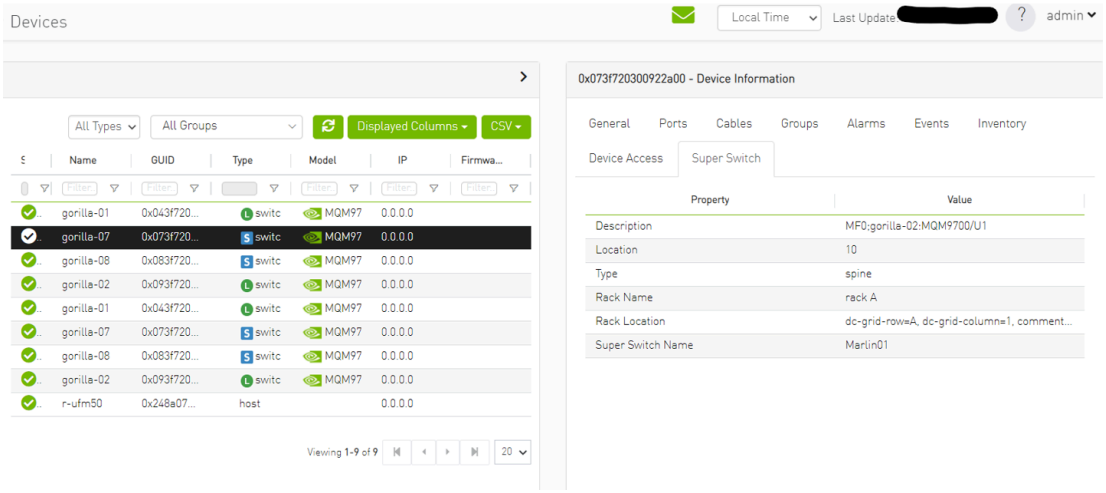




On the devices view, switches that are part of the SuperSwitch group are marked with an additional icon that indicates their role in the group. The "S" icon denotes spines, while the "L" icon denotes lines.



Selecting a switch that belongs to the SuperSwitch group in the properties view allows you to view all the switch properties related to the SuperSwitch group.



Note

Each SuperSwitch definition can include one or more racks where each embedded rack can include multiple leafs and spines switches.

Appendix – Device Management Feature Support

The following table describes the management features available on supported devices.

Feature	10 Gb Ethernet Gateway Module	Grid Director 4700/ 4200/ 4036/ 4036E v3.5	Managed IS5000 Switchesv	Managed SX6000 Switches	Externally Managed IS5000 / SX6000 Switches	Gateway BX5020	HP C-Class
Discovery							
IB L2 Discovery	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Feature	10 Gb Ethernet Gateway Module	Grid Director 4700/ 4200/ 4036/ 4036E v3.5	Managed IS5000 Switchesv	Managed SX6000 Switches	Externally Managed IS5000 / SX6000 Switches	Gateway BX5020	HP C-Class
Advanced Discovery (IP, hostname, Hosts: CPU, memory, FW version)	Yes	Yes	No	Yes	No	No	No
Ethernet access Management interface	Yes	Yes	Yes	Yes	No	No	No
Provisioning/ Configuration							
IB Partitioning (pkey)	Yes	Yes	Yes	Yes	Yes	Yes	Yes
QoS: SL (SM configuration)	N/A	Yes	Yes	Yes	Yes	Yes	Yes
QoS: Rate Limit (SM configuration)	N/A	Yes	Yes	Yes	Yes	Yes	Yes
Interface/VIF Configuration (IP, hostname, mtu, Bonding)	N/A	N/A	N/A	N/A	N/A	No	N/A
Device Monitoring							
Device Resources: CPU, Memory, Disk	No	Yes	No	No	No	No	No

Feature	10 Gb Ethernet Gateway Module	Grid Director 4700/ 4200/ 4036/ 4036E v3.5	Managed IS5000 Switchesv	Managed SX6000 Switches	Externally Managed IS5000 / SX6000 Switches	Gateway BX5020	HP C-Class
Get device alerts (Temperature, PS, Fan) **Note:** This feature is not supported on Switch-X switches.	Yes	Yes	No	Yes	Yes	No	No
L1 (Physical Port) – Monitoring	Yes	Yes	Yes	Yes	Yes	Yes	Yes
L2-3 (Interface/VIF) – Monitoring	No	No	No	No	No	No	No
Congestion Monitoring per port (enables congestion map)	No	Yes	Yes	Yes	Yes	Yes	Yes
Congestion Monitoring per flow (Advanced Package)	No	Yes	No	No	No	No	No
Device Management							
Add/remove to/from Rack	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Add/remove to/from Logical Server	N/A	N/A	N/A	N/A	N/A	N/A	N/A

Feature	10 Gb Ethernet Gateway Module	Grid Director 4700/ 4200/ 4036/ 4036E v3.5	Managed IS5000 Switchesv	Managed SX6000 Switches	Externally Managed IS5000 / SX6000 Switches	Gateway BX5020	HP C-Class
View/clear Alarms	Yes	Yes	Yes	Yes	Yes	Yes	Yes
SSH terminal to device	Yes	Yes	Yes	Yes	No	No	No
Power On	No	No	No	No	No	No	No
Reboot	No	No	No	Yes (SX3606 only)	No	No	No
Shutdown	No	No	No	No	No	No	No
Port Enable/Disable	No	Yes	Yes	Yes	Yes	Yes	Yes
Firmware Upgrade (HCA & switch)	No	Yes	No	Yes (Upon SW upgrade – SX6036 only)	No	No	No
Inband Firmware Upgrade (over InfiniBand connection)	No	No	No	No	Yes	No	No
Software Upgrade (OFED & switch)	No	Yes	No	Yes (SX3606 only)	No	No	No
Protocols							

Feature	10 Gb Ethernet Gateway Module	Grid Director 4700/ 4200/ 4036/ 4036E v3.5	Managed IS5000 Switchesv	Managed SX6000 Switches	Externally Managed IS5000 / SX6000 Switches	Gateway BX5020	HP C-Class
Communication UFM Server – Device	IB/SNMP	IB/UDP /SSH	IB	IB/HTTP/ SSH	IB	IB	IB

1. For a full list of supported IS5000 switches, see Supported IS5000 Switches.
2. QoS Rate Limit (SM configuration): On ConnectX HCAs-only, for hosts.
3. XmitWait counter monitoring requires ConnectX HCAs with firmware version 2.6 and above.
4. This feature requires that the IP address is configured.

Appendix – UFM Event Forwarder

UFM Event Forwarder enables streaming of UFM events via FluentBit forwarder plugin to any external destination.

To deploy the UFM Event Forwarder on a Linux machine:

1. Connect to the Linux host via SSH.
2. Ensure the docker is installed on the host. Run:

```
# docker –version
```

3. Make sure that the docker service is up and running. If it is not, start the docker service. Run:

```
# sudo service docker start
```

4. Pull the UFM Event Forwarder image. Run:

```
# sudo docker pull mellanox/ufm-events-forwarder
```

Alternatively, if you do not have internet connection, contact NVIDIA Support to receive the UFM Event Forwarder docker image and load it to the host. Run:

```
#  sudo cp <ufm-events-forwarder image path> /tmp/
#  sudo docker load -i /tmp/<image name>
```

5. If you are running in HA mode, repeat step 1-4 on the standby node.

> **Note**
>
> Steps 6-9 should only be configured on the master node.

6. Enable the event-forwarder in main UFM config file. Run:

```
# vim /opt/ufm/files/conf/gv.cfg
  [Plugins]
  events_forwarder_enabled=true
```

7. Configure UFM to send events via syslog to the FluentBit event forwarder in gv.cfg.

```
[Logging]
syslog_addr=127.0.0.1:5140
syslog = true
ufm_syslog = true
event_syslog = true
syslog_level = <severity>
```

Note

`<severity>` may be set to any of the following values: CRITICAL, ERROR, WARNING, INFO, or DEBUG.

8. Configure the destination IP and port for the FluentBit event forwarder (requires Python 3):

```
# python /opt/ufm/scripts/events-forwarder/configure-fluent-bit.pyc -i <IP> -p <port>
```

Alternatively, if you have Python 2:

```
# /opt/ufm/venv_ufm/bin/python /opt/ufm/scripts/events-forwarder/configure-fluent-bit.pyc -i <IP> -p <port>
```

9. Start UFM. Run:

```
# /etc/init.d/ufmd start
```

Alternatively, if you are running in HA:

```
# /etc/init.d/ufmha start
```

10. Verify that UFM Event Forwarder is running successfully. Run:

```
# /etc/init.d/ufmd start
ufmd start
Starting opensm:                                           [
OK  ]
Starting MySQL:                                            [
OK  ]
Restarting httpd:                                          [
OK  ]
Starting snmpd:                                            [
OK  ]
Starting UFM main module:                                  [
OK  ]
Starting Events-Forwarder:                                 [
OK  ]
Starting Daily Report:                                     [
OK  ]
Starting UnhealthyPorts:                                   [
OK  ]
Starting ibpm:                                             [
OK  ]
```

Note

Make sure the status of Events-Forwarder is OK.

Stopping UFM will also stop the Event Forwarder.

```
#  /etc/init.d/ufmd stop
ufmd stop
Stopping ibpm:                                             [  OK
]
Stopping Daily Report:                                     [  OK
]
Stopping UnhealthyPorts:                                   [  OK
]
Stopping Events-Forwarder:                                 [  OK
]
Stopping UFM main module:                                  [  OK
]
Stopping MySQL:                                            [  OK
]
Stopping OpenSM:                                           [  OK
]
```

After configuration, the Event Forwarder should always be running on the active node only. After a failover, for example, it will be stopped on the old master and will be started on the new active node.

If the destination IP and port are reconfigured (step 8), the Event Forwarder container should be restarted automatically with the newly applied configuration.

Document Revision History

Release	Date	Description
6.15.6	Jul 8, 2024	Updated: • Bug Fixes in This Release • Installation Notes
6.15.4	Mar 1, 2024	Updated: • Bug Fixes in This Release • Installation Notes
6.15.1	Dec 14, 2023	Updated: • Bug Fixes in This Release • Known Issues in This Release • Supported NVIDIA Internally Managed Switches - Removed MTX6100, MTX6240 and MTX6280 switches and the SX6036G (FDR) gateway • Installation Notes - Updated with the new MFT package version • System Requirements - Added MLNX_OFED23.x • Unsupported Functionalities/Features Added: Cable Validation Report in Subnet Merger
	Dec 19, 2023	• Updated Changes and New Features • Added a Known issue to Bug Fixes in This Release

Release	Date	Description
6.15.0	Nov 5, 2023	Updated: • Changes and New Features • Bug Fixes in This Release • Azure Authentication Login Page - Introduced new Azure authentication login page • Enabling Azure AD Authentication - Added further instructions • UFM Logs Tab - Added log occurrences display Added • Events History • Device Status Events • Link Status Events • GNMI-Telemetry Plugin • In Secondary Telemetry, added instructions on Exposing Switch Aggregation Nodes Telemetry and Stopping Telemetry Endpoint Using CLI Command • UFM Authentication Server • Enabling UFM Authentication Server • Appendix – Secondary Telemetry Fields
6.14.1	Aug 31, 2023	Updated: • Changes and New Features • Bug Fixes in This Release
	Oct 17, 2023	Updated: System Requirements

Release	Date	Description
6.14.0	Aug 10, 2023	Updated: • Changes and New Features • Bug Fixes in This Release • Known Issues in This Release • Plugin Management • Secondary Telemetry • PDR Deterministic Plugin - Updated step 3 in "Deployment". • rest-rdma Plugin • NDT Plugin • Autonomous Link Maintenance (ALM) Plugin • Appendix - Supported Port Counters and Events - Added alarm ID #134, 1602 and 1603 and status column for all alarm IDs. Added: • Disabling Rest Roles Access Control • Enabling Azure AD Authentication • Azure AD Authentication • Health Policy Management • Rest Roles Access Control • Appendix - UFM Factory Reset
6.13.1	May 18, 2023	Updated: • Changes and New Features • Bug Fixes in This Release

Release	Date	Description

6.13.0	May 5, 2023	Updated:
		• Changes and New Features
		• Bug Fixes in This Release
		• Known Issues in This Release
		• Email - Added time zone preference
		• NDT Plugin
		• UFM Telemetry FluentD Streaming (TFS) Plugin - Updated REST API
		• UFM System Dump Tab
		• Appendix - Supported Port Counters and Events
		
		Added:
		• Multi-Subnet UFM
		• Enable Network Fast Recovery
		• NDT Format Merger
		• Subnet Merger UI
		• Added the following Plugins:
		  ◦ UFM Bright Cluster Integration Plugin
		  ◦ UFM Cyber-AI Plugin
		  ◦ Autonomous Link Maintenance (ALM) Plugin
		  ◦ DTS Plugin
		  ◦ Sysinfo Plugin
		  ◦ SNMP Plugin
		  ◦ Packet Mirroring Collector (PMC) Plugin
		  ◦ PDR Deterministic Plugin

	May 9, 2023	Updated • Known Issues in This Release • Appendix – Enhanced Quality of Service - Updated notes and example

Release	Date	Description
6.12.1	Feb 19, 2023	Updated • Changes and New Features • Bug Fixes in This Release • Known Issues in This Release
	Mar 1, 2023	Updated Changes and New Features
	Mar 16, 2023	Updated Changes and New Features - Added MFT package integration details
	Mar 27, 2023	Updated UFM Server Communication with Externally Managed Switches
6.12.0	Feb 2, 2023	Updated: • Changes and New Features • Bug Fixes in This Release • Known Issues in This Release • Configuring Partial Switch ASIC Failure Events • Updated example in Multi-port SM • UFM System Dump Tab • Appendix – Used Ports • Appendix - UFM SLURM Integration Added: • Added a note under Ports Window • Added a note under Unhealthy Ports Window • Delegate Authentication to a Proxy Removed: • UFM Logical Elements tab from the Web UI
	Feb 6, 2023	Updated Troubleshooting

Release	Date	Description
6.11.1	Dec 1, 2022	Updated: • Changes and New Features to include the upgrade of NVIDIA SHARP SW version • Installation Notes • Known Issues in This Release • Troubleshooting
	Dec 19, 2022	Updated Changes and New Features

6.11.0	Nov 21, 2022	Updated:
		• Added a link to UFM SDK 3.0 under Related Documentation
		• Changes and New Features
		• Installation Notes
		• Bug Fixes in This Release
		• Known Issues in This Release
		• Installing UFM HA Package
		• Network Map with new screenshots and new instructions for Map Information and Settings
		• Devices Window with new screenshots
		• PSID and Firmware Version In-Band Discovery
		• Groups Window with new screenshots
		• Table Enhancements with new screenshots
		• UFM Telemetry FluentD Streaming (TFS) Plugin
		• Enabling UFM Telemetry
		
		Added:
		• CPU Affinity on UFM
		• Switch Management IP Address Discovery
		• UFM Events Fluent Streaming (EFS) Plugin
		• In Telemetry
		  ◦ Changing UFM Telemetry Default Configuration
		  ◦ Supporting Generic Counters Parsing and Display
		  ◦ Supporting Multiple Telemetry Instances Fetch
		  ◦ Secondary Telemetry

EULA, Legal Notices and 3rd Party Licenses

License Agreement

This license is a legal agreement (“Agreement”) between you and Mellanox Technologies, Ltd. ("NVIDIA") and governs the use of the NVIDIA UFM software and materials provided hereunder (“SOFTWARE”). If you are entering into this Agreement on behalf of a company or other legal entity, you represent that you have the legal authority to bind the entity to this Agreement, in which case “you” will mean the entity you represent.

You agree to use the SOFTWARE only for purposes that are permitted by (a) this license, and (b) any applicable law, regulation, or generally accepted practices or guidelines in the relevant jurisdictions.

1. License. Subject to the terms and conditions of this Agreement and payment of applicable subscription fee, NVIDIA MELLANOX grants you a personal, non-exclusive, non-sublicensable (except as provided in this Agreement), non-transferable, non-commercial license to install and use the Software for your internal business purposes for configuring, operating, and managing your InfiniBand network and not for further distribution.

2. Authorized Users. You may allow access and use of the Software to: (i) employees and contractors of your entity provided that the access and use of the Software is made from your secure network to perform work on your behalf and (ii) If you are an academic institution you may allow users enrolled or employed by the academic institution to access and use the Software from your secure network (“Authorized Users”). You hereby undertake to be responsible and liable for any non-compliance with the terms of this Agreement by your Authorized Users. You further agree to immediately resolve any non-compliance by your Authorized Users of which you become aware and endeavor take necessary steps to prevent any new occurrences.

3. Limitations Your license to use the SOFTWARE is restricted as follows:

3.1 The SOFTWARE is licensed for your use in systems with the registered NVIDIA Host Channel Adapter (HCA) Products or related adapter products.

3.2 Each copy of the SOFTWARE shall be limited to the number of HCAs indicated in the applicable purchase order.

3.3 You may use software back-up utilities to make one back-up copy of the Software Product. You may use the back-up copy solely for archival purposes

3.4 You may not use the SOFTWARE in conjunction with a number of managed nodes or managed devices which is beyond the allowable limit or copy the SOFTWARE on additional hardware. You shall not use any features which are not included in the scope of this Agreement as described in the accompanying documentation.

3.5 You may not reverse engineer, decompile or disassemble, or remove copyright or other proprietary notices from any portion of the SOFTWARE or copies of the SOFTWARE.

3.6 You may not disclose the results of benchmarking, competitive analysis, regression, or performance data relating to the SOFTWARE without the prior written permission from NVIDIA Mellanox.

3.7 Except as expressly provided in this license, you may not copy, sell, rent, sublicense, transfer, distribute, modify, or create derivative works of any portion of the SOFTWARE. For clarity, unless, you have an agreement with NVIDIA Mellanox for this purpose you may not distribute or sublicense the SOFTWARE as a stand-alone product.

3.8 You may not bypass, disable, or circumvent any technical limitation, encryption, security, digital rights management, or authentication mechanism in the SOFTWARE.

3.9 You may not use the Software in any manner that would cause it to become subject to an open source software license. As examples, licenses that require as a condition of use, modification, and/or distribution that the Software be: (i) disclosed or distributed in source code form; (ii) licensed for the purpose of making derivative works; or (iii) redistributable at no charge.

3.10 Unless you have an agreement with NVIDIA Mellanox for this purpose, you may not use the Software with any system or application where the use or failure of the system or application can reasonably be expected to threaten or result in personal injury, death, or catastrophic loss. Examples include use in avionics, navigation, military, medical, life support or other life critical applications. NVIDIA Mellanox does not design, test, or manufacture the Software for these critical uses and NVIDIA

Mellanox shall not be liable to you or any third party, in whole or in part, for any claims or damages arising from such uses.

3.11 You agree to defend, indemnify and hold harmless NVIDIA Mellanox and its affiliates, and their respective employees, contractors, agents, officers and directors, from and against any and all claims, damages, obligations, losses, liabilities, costs or debt, fines, restitutions and expenses (including but not limited to attorney's fees and costs incident to establishing the right of indemnification) arising out of or related to your use of the Software outside of the scope of this license, or not in compliance with its terms.

4. Updates. NVIDIA Mellanox may, at its option, make available patches, workarounds, or other updates to this Software. Unless the updates are provided with their separate governing terms, they are deemed part of the Software licensed to you as provided in this license. You agree that the form and content of the Software that NVIDIA Mellanox provides may change without prior notice to you. While NVIDIA Mellanox generally maintains compatibility between versions, NVIDIA Mellanox may in some cases make changes that introduce incompatibilities in future versions of the SOFTWARE.

5. Pre-Release Versions. Software versions identified as alpha, beta, preview, early access or otherwise as pre-release may not be fully functional, may contain errors or design flaws, and may have reduced or different security, privacy, availability, and reliability standards relative to commercial versions of NVIDIA Mellanox software and materials. You may use a pre-release Software version at your own risk, understanding that these versions are not intended for use in production or business-critical systems. NVIDIA Mellanox may choose not to make available a commercial version of any pre-release Software. NVIDIA Mellanox may also choose to abandon development and terminate the availability of a pre-release Software at any time without liability.

6. Third-Party Components. The Software may include third-party components with separate legal notices or terms as may be described in proprietary notices accompanying the Software or as provided in an Exhibit to this Agreement. If and to the extent there is a conflict between the terms in this license and the third-party license terms, the third-party terms control only to the extent necessary to resolve the conflict. For details regarding the third party components, please review Exhibit A.

7. OWNERSHIP

   7.1 NVIDIA Mellanox or its licensors reserves all rights, title, and interest in and to the Software not expressly granted to you under this license NVIDIA Mellanox and its suppliers hold all rights, title, and interest in and to the Software, including their respective intellectual property rights. The Software is copyrighted and protected by

the laws of the United States and other countries, and international treaty provisions.

7.2 Subject to the rights of NVIDIA Mellanox and its suppliers in the Software, you hold all rights, title, and interest in and to your applications and your derivative works of the sample source code delivered in the Software including their respective intellectual property rights.

8. You may, but are not obligated to, provide to NVIDIA Mellanox Feedback. “Feedback” means suggestions, fixes, modifications, feature requests or other feedback regarding the Software. Feedback, even if designated as confidential by you, shall not create any confidentiality obligation for NVIDIA Mellanox. NVIDIA Mellanox and its designees have a perpetual, non-exclusive, worldwide, irrevocable license to use, reproduce, publicly display, modify, create derivative works of, license, sublicense, and otherwise distribute and exploit Feedback as NVIDIA Mellanox sees fit without payment and without obligation or restriction of any kind on account of intellectual property rights or otherwise.

9. No Warranties. THE SOFTWARE IS PROVIDED AS-IS. TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW NVIDIA MELLANOX AND ITS AFFILIATES EXPRESSLY DISCLAIM ALL WARRANTIES OF ANY KIND OR NATURE, WHETHER EXPRESS, IMPLIED OR STATUTORY, INCLUDING, BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, OR FITNESS FOR A PARTICULAR PURPOSE. NVIDIA MELLANOX DOES NOT WARRANT THAT THE SOFTWARE WILL MEET YOUR REQUIREMENTS OR THAT THE OPERATION THEREOF WILL BE UNINTERRUPTED OR ERROR-FREE, OR THAT ALL ERRORS WILL BE CORRECTED.

10. Limitations of Liability. TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW NVIDIA MELLANOX AND ITS AFFILIATES SHALL NOT BE LIABLE FOR ANY SPECIAL, INCIDENTAL, PUNITIVE OR CONSEQUENTIAL DAMAGES, OR FOR ANY LOST PROFITS, PROJECT DELAYS, LOSS OF USE, LOSS OF DATA OR LOSS OF GOODWILL, OR THE COSTS OF PROCURING SUBSTITUTE PRODUCTS, ARISING OUT OF OR IN CONNECTION WITH THIS LICENSE OR THE USE OR PERFORMANCE OF THE SOFTWARE, WHETHER SUCH LIABILITY ARISES FROM ANY CLAIM BASED UPON BREACH OF CONTRACT, BREACH OF WARRANTY, TORT (INCLUDING NEGLIGENCE), PRODUCT LIABILITY OR ANY OTHER CAUSE OF ACTION OR THEORY OF LIABILITY, EVEN IF NVIDIA MELLANOX HAS PREVIOUSLY BEEN ADVISED OF, OR COULD REASONABLY HAVE FORESEEN, THE POSSIBILITY OF SUCH DAMAGES. IN NO EVENT WILL NVIDIA MELLANOX AND ITS AFFILIATES TOTAL CUMULATIVE LIABILITY UNDER OR ARISING OUT OF THIS LICENSE EXCEED US$10.00. THE NATURE OF THE LIABILITY OR THE NUMBER OF CLAIMS OR SUITS SHALL NOT ENLARGE OR EXTEND THIS LIMIT.

11. T Your rights under this license will terminate automatically without notice from NVIDIA Mellanox (a) upon expiration of your subscription, (b) if you fail to comply with

any term and condition of this license including non-payment of applicable fees, or (c) if you commence or participate in any legal proceeding against NVIDIA Mellanox with respect to the Software. NVIDIA Mellanox may terminate this license with advance written notice to you, if NVIDIA Mellanox decides to no longer provide the Software in a country or, in NVIDIA Mellanox's sole discretion, the continued use of it is no longer commercially viable. Upon any termination of this license, you agree to promptly discontinue use of the Software and destroy all copies in your possession or control. All provisions of this license will survive termination, except for the license granted to you.

12. Product Support. Product support for the Software Product is provided by NVIDIA Mellanox or its authorized agents under the applicable subscription license, in accordance with NVIDIA Mellanox's standard support and maintenance terms and conditions. For product support, please refer to NVIDIA Mellanox support number provided in the documentation.

13. Applicable Law. This license will be governed in all respects by the laws of the United States and of the State of Delaware, without regard to the conflicts of laws principles. The United Nations Convention on Contracts for the International Sale of Goods is specifically disclaimed. You agree to all terms of this license in the English language. The state or federal courts residing in Santa Clara County, California shall have exclusive jurisdiction over any dispute or claim arising out of this license. Notwithstanding this, you agree that NVIDIA Mellanox shall still be allowed to apply for injunctive remedies or urgent legal relief in any jurisdiction.

14. No Assignment. This license and your rights and obligations thereunder may not be assigned by you by any means or operation of law without NVIDIA Mellanox's permission. Any attempted assignment not approved by NVIDIA MELLANOX in writing shall be void and of no effect. NVIDIA Mellanox may assign, delegate, or transfer this license and its rights and obligations, and if to a non-affiliate you will be notified.

15. E The Software is subject to United States export laws and regulations. You agree to comply with all applicable U.S. and international export laws, including the Export Administration Regulations (EAR) administered by the U.S. Department of Commerce and economic sanctions administered by the U.S. Department of Treasury's Office of Foreign Assets Control (OFAC). These laws include restrictions on destinations, end-users and end-use. By accepting this license, you confirm that you are not currently residing in a country or region currently embargoed by the U.S. and that you are not otherwise prohibited from receiving the Software.

16. Government Use. The Software is, and shall be treated as being, "Commercial Items" as that term is defined at 48 CFR § 2.101, consisting of "commercial computer software" and "commercial computer software documentation", respectively, as such terms are used in, respectively, 48 CFR § 12.212 and 48 CFR §§ 227.7202 & 252.227-

7014(a)(1). Use, duplication or disclosure by the U.S. Government or a U.S. Government subcontractor is subject to the restrictions in this license pursuant to 48 CFR § 12.212 or 48 CFR § 227.7202. In no event shall the US Government user acquire rights in the Software beyond those specified in 48 C.F.R. 52.227-19(b)(1)-(2).

17. Please direct your legal notices or other correspondence to NVIDIA Corporation, 2788 San Tomas Expressway, Santa Clara, CA, 95051 United States of America, Attention: Legal Department and to: NBU-Legal_Notices@exchange.nvidia.com

18. Entire Agreement. This license is the final, complete, and exclusive agreement between the parties relating to the subject matter of this license and supersedes all prior or contemporaneous understandings and agreements relating to this subject matter, whether oral or written. If any court of competent jurisdiction determines that any provision of this license is illegal, invalid, or unenforceable, the remaining provisions will remain in full force and effect. Any amendment or waiver under this license shall be in writing and signed by representatives of both parties.

(v APR. 28, 2022)

Exhibit A

SOFTWARE includes the following open source/ freeware that are subject to specific license conditions listed in the table below, which may be updated from time to time by NVIDIA Mellanox or the Open Source provider. The below table is current as of December 2021. To obtain source code for software provided under licenses that require redistribution of source code, including the GNU General Public License or for update queries contact: http://www.mellanox.com/page/gnu_code_request .This offer is valid for a period of three (3) years from the date of the distribution of this product by NVIDIA Mellanox.

Embedded Software Components

Notice

This document is provided for information purposes only and shall not be regarded as a warranty of a certain functionality, condition, or quality of a product. NVIDIA Corporation (“NVIDIA”) makes no representations or warranties, expressed or implied, as to the accuracy or completeness of the information contained in this document and assumes no responsibility for any errors contained herein. NVIDIA shall have no liability for the consequences or use of such information or for any infringement of patents or other rights of third parties that may result from its use. This document is not a commitment to develop, release, or deliver any Material (defined below), code, or functionality.

NVIDIA reserves the right to make corrections, modifications, enhancements, improvements, and any other changes to this document, at any time without notice.

Customer should obtain the latest relevant information before placing orders and should verify that such information is current and complete.

NVIDIA products are sold subject to the NVIDIA standard terms and conditions of sale supplied at the time of order acknowledgement, unless otherwise agreed in an individual sales agreement signed by authorized representatives of NVIDIA and customer (“Terms of Sale”). NVIDIA hereby expressly objects to applying any customer general terms and conditions with regards to the purchase of the NVIDIA product referenced in this document. No contractual obligations are formed either directly or indirectly by this document.

NVIDIA products are not designed, authorized, or warranted to be suitable for use in medical, military, aircraft, space, or life support equipment, nor in applications where failure or malfunction of the NVIDIA product can reasonably be expected to result in personal injury, death, or property or environmental damage. NVIDIA accepts no liability for inclusion and/or use of NVIDIA products in such equipment or applications and therefore such inclusion and/or use is at customer’s own risk.

NVIDIA makes no representation or warranty that products based on this document will be suitable for any specified use. Testing of all parameters of each product is not necessarily performed by NVIDIA. It is customer’s sole responsibility to evaluate and determine the applicability of any information contained in this document, ensure the product is suitable and fit for the application planned by customer, and perform the necessary testing for the application in order to avoid a default of the application or the product. Weaknesses in customer’s product designs may affect the quality and reliability of the NVIDIA product and may result in additional or different conditions and/or requirements beyond those contained in this document. NVIDIA accepts no liability related to any default, damage, costs, or problem which may be based on or attributable to: (i) the use of the NVIDIA product in any manner that is contrary to this document or (ii) customer product designs.

No license, either expressed or implied, is granted under any NVIDIA patent right, copyright, or other NVIDIA intellectual property right under this document. Information published by NVIDIA regarding third-party products or services does not constitute a license from NVIDIA to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property rights of the third party, or a license from NVIDIA under the patents or other intellectual property rights of NVIDIA.

Reproduction of information in this document is permissible only if approved in advance by NVIDIA in writing, reproduced without alteration and in full compliance with all applicable export laws and regulations, and accompanied by all associated conditions, limitations, and notices.

THIS DOCUMENT AND ALL NVIDIA DESIGN SPECIFICATIONS, REFERENCE BOARDS, FILES, DRAWINGS, DIAGNOSTICS, LISTS, AND OTHER DOCUMENTS (TOGETHER AND SEPARATELY, “MATERIALS”) ARE BEING PROVIDED “AS IS.” NVIDIA MAKES NO WARRANTIES, EXPRESSED, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE MATERIALS, AND EXPRESSLY DISCLAIMS ALL IMPLIED WARRANTIES OF NONINFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE. TO THE EXTENT NOT PROHIBITED BY LAW, IN NO EVENT WILL NVIDIA BE LIABLE FOR ANY DAMAGES, INCLUDING WITHOUT LIMITATION ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, PUNITIVE, OR CONSEQUENTIAL DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF

ANY USE OF THIS DOCUMENT, EVEN IF NVIDIA HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Notwithstanding any damages that customer might incur for any reason whatsoever, NVIDIA's aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms of Sale for the product.

Trademarks

NVIDIA and the NVIDIA logo are trademarks and/or registered trademarks of NVIDIA Corporation in the U.S. and other countries. Other company and product names may be trademarks of the respective companies with which they are associated.