# Table of Contents

## 1 Release Notes

- **1.1 Key Features** ................................................................. 13
- **1.2 Changes and New Features** ........................................ 13
  - **1.2.1 Unsupported Functionalities/Features** ....................... 15
- **1.3 Installation Notes** ........................................................... 16
  - **1.3.1 Supported Devices** ................................................... 16
  - **1.3.2 System Requirements** ............................................... 16
  - **1.3.3 Software Update from Prior Versions** ....................... 19
- **1.4 Bug Fixes in This Release** ............................................ 19
- **1.5 Known Issues in This Release** ...................................... 21
- **1.6 Changes and New Features History** ............................... 21
- **1.7 Bug Fixes History** ....................................................... 27
- **1.8 Known Issues History** .................................................. 35

## 2 UFM Overview

- **2.1 Scale-Out Your Fabric with Unified Fabric Manager** ........ 44
- **2.2 UFM Benefits** ............................................................... 45
- **2.3 Main Functionality Modules** ......................................... 46
- **2.4 UFM Communication Requirements** ................................. 47
  - **2.4.1 UFM Server Communication with Clients** ................... 47
  - **2.4.2 UFM Server Communication with InfiniBand Switches** .... 48
  - **2.4.3 UFM Server Communication with InfiniBand Hosts** ......... 49
  - **2.4.4 UFM Server High Availability (HA) Active—Standby Communication** .......................... 50
- **2.5 UFM Software Architecture** .......................................... 51
  - **2.5.1 Graphical User Interface** ............................................ 52
  - **2.5.2 Client Tier API** ....................................................... 52
  - **2.5.3 Client Tier SDK Tools** ............................................. 52
  - **2.5.4 UFM Server** ............................................................ 52
  - **2.5.5 Subnet Manager** ..................................................... 52
  - **2.5.6 NVIDIA Scalable Hierarchical Aggregation and Reduction Protocol (SHARP)™ Aggregation Manager** ..................................................... 52
  - **2.5.7 Performance Manager** .............................................. 52
  - **2.5.8 Device Manager** ..................................................... 53
2.5.9 UFM Switch Agent ............................................................................. 53
2.5.10 Communication Protocols ................................................................. 53
2.6 Getting Familiar with UFM's Data Model ............................................. 53
  2.6.1 UFM Model Basics ........................................................................... 53
  2.6.2 Physical Model .................................................................................. 53
2.7 UFM Web UI Overview ......................................................................... 54
  2.7.1 Access the WebUI ............................................................................ 54
  2.7.2 WebUI Layout ................................................................................... 55
  2.7.3 Set User Preferences ...................................................................... 57
3 UFM Installation and Initial Configuration ............................................. 61
  3.1 Prerequisites for UFM Server Software Installation .......................... 61
  3.2 UFM Installation Steps ......................................................................... 61
  3.3 UFM Installation Steps ......................................................................... 61
    3.3.1 Downloading UFM Software and License File ............................... 61
    3.3.2 Installing UFM Server Software .................................................... 64
  3.4 Activating Software License ................................................................. 78
    3.4.1 Licensing ....................................................................................... 78
  3.5 UFM Configuration .............................................................................. 80
    3.5.1 Initial Configuration ....................................................................... 80
    3.5.2 Additional Configuration (Optional) .............................................. 81
  3.6 Running UFM Server Software ............................................................. 100
    3.6.1 Running UFM Server Software in Management Mode ................ 101
    3.6.2 Running UFM Software in High Availability Mode ..................... 101
    3.6.3 HTTP/HTTPS Configuration ........................................................ 101
    3.6.4 UFM Internal Web Server Configuration ..................................... 102
    3.6.5 User Authentication ....................................................................... 102
    3.6.6 UFM Authentication Server ........................................................... 102
    3.6.7 Azure AD Authentication ............................................................... 103
    3.6.8 Licensing .................................................................................... 110
    3.6.9 Showing UFM Processes Status .................................................. 110
  3.7 Upgrading UFM Software ................................................................. 111
    3.7.1 Upgrading UFM on Bare Metal Server ....................................... 111
    3.7.2 Upgrading UFM on Docker Container ....................................... 113
  3.8 Uninstalling UFM .............................................................................. 115
# UFM Web UI

## 9.1 Fabric Dashboard

- 9.1.1 Dashboard Views and Panel Management .................................................. 162
- 9.1.2 Dashboard Timeline Snapshots ................................................................. 163
- 9.1.3 Dashboard Panels ................................................................................. 164
- 9.1.4 Top N Servers/Switches by Rx or Tx Bandwidth ........................................ 164
- 9.1.5 Top N Congested Servers/Switches by Rx/Tx Bandwidth ............................. 168
- 9.1.6 Top N Utilized PKeys ............................................................................ 171
- 9.1.7 Top N Alarmed Servers/Switches ............................................................. 172
- 9.1.8 Inventory Summary ................................................................................ 176
- 9.1.9 Fabric Utilization .................................................................................... 177
- 9.1.10 Recent Activities .................................................................................... 178
- 9.1.11 Traffic Map .......................................................................................... 180
- 9.1.12 Events History ...................................................................................... 187

## 9.2 Network Map

- 9.2.1 Network Map Components .................................................................... 189
- 9.2.2 Selecting Map Elements ......................................................................... 190
- 9.2.3 Map Information and Settings ................................................................ 191
- 9.2.4 Map View Tab ........................................................................................ 194
- 9.2.5 Map Zoom In Tab .................................................................................... 194
- 9.2.6 Map Layouts ........................................................................................... 196
- 9.2.7 Information View Tab ............................................................................. 198
- 9.2.8 Link Analysis .......................................................................................... 199
- 9.2.9 Topology Compare .................................................................................. 204
- 9.2.10 Properties Tab ....................................................................................... 206
- 9.2.11 Network Map Elements Actions ............................................................ 207

## 9.3 Managed Elements

- 9.3.1 Devices Window ..................................................................................... 210
- 9.3.2 Ports Window .......................................................................................... 211
- 9.3.3 Virtual Ports Window ............................................................................. 234
- 9.3.4 Unhealthy Ports Window ......................................................................... 238
- 9.3.5 Cables Window ....................................................................................... 239
- 9.3.6 Groups Window ....................................................................................... 242
- 9.3.7 Inventory Window .................................................................................. 242
10.3 Functionality............................................................................................................. 327
11 UFM Plugins .............................................................................................................. 341
11.1 rest-rdma Plugin ..................................................................................................... 341
11.1.1 Deployment Server ............................................................................................. 341
11.1.2 How to Run........................................................................................................ 342
11.1.3 Authentication Configuration .............................................................................. 343
11.2 NDT Plugin .............................................................................................................. 345
11.2.1 Overview ........................................................................................................... 345
11.2.2 Deployment ........................................................................................................ 346
11.2.3 Authentication ................................................................................................... 346
11.2.4 REST API ......................................................................................................... 346
11.2.5 NDT Format - Topodiff ..................................................................................... 347
11.2.6 NDT Format - Subnet Merger .......................................................................... 349
11.3 UFM Telemetry Fluentd Streaming (TFS) Plugin .................................................... 357
11.3.1 Overview ........................................................................................................... 357
11.3.2 Deployment ........................................................................................................ 357
11.3.3 Authentication ................................................................................................... 357
11.3.4 Rest API ............................................................................................................ 357
11.4 UFM Events Fluent Streaming (EFS) Plugin .......................................................... 358
11.4.1 Overview ........................................................................................................... 358
11.4.2 Deployment ........................................................................................................ 358
11.4.3 Authentication ................................................................................................... 358
11.4.4 Rest API ............................................................................................................ 358
11.5 UFM Bright Cluster Integration Plugin .................................................................... 359
11.5.1 Overview ........................................................................................................... 359
11.5.2 Deployment ........................................................................................................ 359
11.5.3 Authentication ................................................................................................... 359
11.5.4 Bright Cluster Integration UI ............................................................................ 359
11.5.5 Rest API ............................................................................................................ 361
11.6 UFM Cyber-AI Plugin ............................................................................................. 361
11.6.1 Overview ........................................................................................................... 361
11.6.2 Deployment ........................................................................................................ 361
11.7 Autonomous Link Maintenance (ALM) Plugin ....................................................... 363
11.7.1 Overview ........................................................................................................... 363
<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>11.7.2</td>
<td>Schematic Flow</td>
<td>364</td>
</tr>
<tr>
<td>11.7.3</td>
<td>Deployment</td>
<td>364</td>
</tr>
<tr>
<td>11.7.4</td>
<td>Data Collection</td>
<td>365</td>
</tr>
<tr>
<td>11.7.5</td>
<td>ALM Configuration</td>
<td>366</td>
</tr>
<tr>
<td>11.7.6</td>
<td>ALM UI</td>
<td>367</td>
</tr>
<tr>
<td>11.7.7</td>
<td>ALM Jobs</td>
<td>372</td>
</tr>
<tr>
<td>11.8</td>
<td>DTS Plugin</td>
<td>372</td>
</tr>
<tr>
<td>11.8.1</td>
<td>Overview</td>
<td>372</td>
</tr>
<tr>
<td>11.8.2</td>
<td>Deployment</td>
<td>372</td>
</tr>
<tr>
<td>11.8.3</td>
<td>DTS UI</td>
<td>374</td>
</tr>
<tr>
<td>11.9</td>
<td>GRPC-Streamer Plugin</td>
<td>375</td>
</tr>
<tr>
<td>11.9.1</td>
<td>Authentication</td>
<td>375</td>
</tr>
<tr>
<td>11.9.2</td>
<td>Create a Session to UFM from GRPC</td>
<td>376</td>
</tr>
<tr>
<td>11.9.3</td>
<td>Create New Subscription</td>
<td>376</td>
</tr>
<tr>
<td>11.9.4</td>
<td>Edit Known Subscription</td>
<td>377</td>
</tr>
<tr>
<td>11.9.5</td>
<td>Get List of Known Subscribers</td>
<td>378</td>
</tr>
<tr>
<td>11.9.6</td>
<td>Delete a Known Subscriber</td>
<td>378</td>
</tr>
<tr>
<td>11.9.7</td>
<td>Run a Known Subscriber Once</td>
<td>378</td>
</tr>
<tr>
<td>11.9.8</td>
<td>Run Streamed Data of a Known Subscriber</td>
<td>379</td>
</tr>
<tr>
<td>11.9.9</td>
<td>Run a New Subscriber Once</td>
<td>379</td>
</tr>
<tr>
<td>11.9.10</td>
<td>Run New Subscriber Streamed Data</td>
<td>380</td>
</tr>
<tr>
<td>11.9.11</td>
<td>Run A Serialization on All the Running Streams</td>
<td>381</td>
</tr>
<tr>
<td>11.9.12</td>
<td>Stop a Running Stream</td>
<td>381</td>
</tr>
<tr>
<td>11.9.13</td>
<td>Run a subscribe stream</td>
<td>381</td>
</tr>
<tr>
<td>11.9.14</td>
<td>Get the variables from a known subscriber</td>
<td>382</td>
</tr>
<tr>
<td>11.10</td>
<td>Sysinfo Plugin</td>
<td>382</td>
</tr>
<tr>
<td>11.10.1</td>
<td>Overview</td>
<td>382</td>
</tr>
<tr>
<td>11.10.2</td>
<td>Deployment</td>
<td>382</td>
</tr>
<tr>
<td>11.10.3</td>
<td>REST API</td>
<td>383</td>
</tr>
<tr>
<td>11.10.4</td>
<td>Sysinfo Query Format</td>
<td>383</td>
</tr>
<tr>
<td>11.11</td>
<td>SNMP Plugin</td>
<td>383</td>
</tr>
<tr>
<td>11.11.1</td>
<td>Deployment</td>
<td>384</td>
</tr>
<tr>
<td>11.11.2</td>
<td>Authentication</td>
<td>384</td>
</tr>
<tr>
<td>11.11.3</td>
<td>REST API</td>
<td>384</td>
</tr>
<tr>
<td>Section</td>
<td>Title</td>
<td>Page</td>
</tr>
<tr>
<td>---------</td>
<td>----------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>13.7.7</td>
<td>NVIDIA SHARP AM Logs</td>
<td>481</td>
</tr>
<tr>
<td>13.7.8</td>
<td>NVIDIA SHARP AM Version</td>
<td>481</td>
</tr>
<tr>
<td>13.8</td>
<td>Appendix - UFM SLURM Integration</td>
<td>481</td>
</tr>
<tr>
<td>13.8.1</td>
<td>Prerequisites</td>
<td>481</td>
</tr>
<tr>
<td>13.8.2</td>
<td>Automatic Installation</td>
<td>481</td>
</tr>
<tr>
<td>13.8.3</td>
<td>Manual Installation</td>
<td>482</td>
</tr>
<tr>
<td>13.8.4</td>
<td>UFM SLURM Config File</td>
<td>482</td>
</tr>
<tr>
<td>13.8.5</td>
<td>Configuring UFM for NVIDIA SHARP Allocation</td>
<td>483</td>
</tr>
<tr>
<td>13.8.6</td>
<td>Prolog and Epilog</td>
<td>483</td>
</tr>
<tr>
<td>13.8.7</td>
<td>Integration Files</td>
<td>483</td>
</tr>
<tr>
<td>13.8.8</td>
<td>Running UFM-SLURM Integration</td>
<td>484</td>
</tr>
<tr>
<td>13.9</td>
<td>Appendix - Switch Grouping</td>
<td>485</td>
</tr>
<tr>
<td>13.9.1</td>
<td>UI Presentation</td>
<td>485</td>
</tr>
<tr>
<td>13.10</td>
<td>Appendix - Device Management Feature Support</td>
<td>489</td>
</tr>
<tr>
<td>14</td>
<td>Document Revision History</td>
<td>493</td>
</tr>
<tr>
<td>15</td>
<td>EULA, Legal Notices and 3rd Party Licenses</td>
<td>499</td>
</tr>
<tr>
<td>15.1</td>
<td>Legal Notice</td>
<td>499</td>
</tr>
<tr>
<td>15.2</td>
<td>Third-Party Licenses</td>
<td>499</td>
</tr>
<tr>
<td>15.3</td>
<td>License Agreement</td>
<td>499</td>
</tr>
<tr>
<td>15.3.1</td>
<td>Exhibit A</td>
<td>503</td>
</tr>
</tbody>
</table>
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.

This user guide provides documentation for network administrators responsible for deploying, configuring, monitoring, and troubleshooting the network in their data center.

For a list of the new features, bug fixes and known issues in this release, see Release Notes.

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.
1 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.

1.1 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

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

1.2 Changes and New Features

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

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

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>XDR Support</td>
<td>Added XDR support readiness (based on XDR setup simulations only).</td>
</tr>
<tr>
<td></td>
<td>Added support for UFM Network topology planarized network.</td>
</tr>
<tr>
<td>Switch NVOS Support</td>
<td>Added support for NVOS switches in UFM.</td>
</tr>
<tr>
<td>Feature</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Device Access</td>
<td>Added the ability to record two sets of switch login credentials on UFM. Refer to <a href="#">Device Access</a>.</td>
</tr>
<tr>
<td>UFM Authentication Server</td>
<td>The authentication server is enabled by default. Refer to <a href="#">Configurations of the UFM Authentication Server</a>.</td>
</tr>
</tbody>
</table>
| InfiniBand Cluster Procedures Automation | Upon UFM startup, the following procedures are initiated:  
  - Generate default Fabric health - Refer to [Fabric Health Tab](#).  
  - Validate cluster topology - Refer to [Topology Compare Tab](#).  
  - Obtain and keep the UFM's initial system dump - Refer to [UFM System Dump Tab](#).  
| Secondary Telemetry             | • Added the "rq_general_error" field to support retrieving the number of packet drops due to MPR mismatch.  
  • Added support for reporting cable length information for NDR optic cables.  
  • Added support for retrieving cable information for downed ports.  
  • Added switch/module power usage data in UFM telemetry.  
  For more information, refer to [Low-Frequency (Secondary) Telemetry Fields](#). |
| Events Simulation               | Added the ability to simulate any event from the Events policy tab. Refer to [Events Policy Simulation](#).                                                                                                 |
| Unhealthy Port Enhancement      | Added the ability to display valid unhealthy port information (eliminating non-zero port values) when added manually. Refer to [Unhealthy Ports Window](#).                                                   |
| UFM High Availability           | Added support for UFM HA to configure IPv4 and IPv6 concurrently to provide Virtual IP address. Refer to [UFM High-Availability User Guide](#).                                                             |
| REST APIs                       | Unhealthy Ports REST API: Added the ability to return device state (healthy/unhealthy). Refer to [NVIDIA UFM Enterprise REST API Guide](#) → Unhealthy Ports REST API  
  Add switch/module power usage data in UFM telemetry. Refer to [NVIDIA UFM Enterprise REST API Guide](#) → Systems REST API |
| Plugins                         | Enhanced Plugins management infrastructure.  
  - **GNMI-Telemetry Plugin**: Added support for streaming gNMI events and restricted authentication via client SAN pinning/filtering on the gNMI plugin server-side.  
  - **UFM Telemetry Manager (UTM) Plugin**: Added a new plugin to monitor and manage running UFM Telemetry instances.  
  - **UFM Consumer Plugin**: Added a new plugin that serves as a Multi-Subnet consumer within UFM, offering all the functionalities available for Multi-Subnet UFM.  
  - **PDR Deterministic Plugin**: Updated high BER analysis with the up-to-date high BER algorithm. |
<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
</table>
| **Autonomous Link Maintenance (ALM) Plugin** | Added the following capabilities:  
- Model configuration  
- Configure to reset both ports after isolation  
- Reflect model performance to the user |
| **UFM Telemetry Fluentd Streaming (TFS) Plugin** | Added an option to the TFS to stream the data using the CLX C streamer instead of the Python streamer. |

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

⚠️ 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](#).

### 1.2.1 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

⚠️ 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).
1.3 Installation Notes

1.3.1 Supported Devices

1.3.1.1 Supported NVIDIA Externally Managed Switches

<table>
<thead>
<tr>
<th>Type</th>
<th>Model</th>
<th>Latest Tested Firmware Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>NDR switches</td>
<td>MQM9790</td>
<td>31.2021.4036</td>
</tr>
<tr>
<td>HDR switches</td>
<td>MQM8790</td>
<td>27.2012.4036</td>
</tr>
<tr>
<td>EDR switches</td>
<td>SB7790, SB7890</td>
<td>15.2010.4402</td>
</tr>
</tbody>
</table>

1.3.1.2 Supported NVIDIA Internally Managed Switches

<table>
<thead>
<tr>
<th>Type</th>
<th>Model</th>
<th>Latest Tested OS Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>NDR switches</td>
<td>MQM9700</td>
<td>MLNX-OS 3.11.4002 NVOS 25.01.4000</td>
</tr>
<tr>
<td>HDR switches</td>
<td>MQ8700, MCS8500, TQ8100-HS2F, TQ8200-HS2F</td>
<td>MLNX-OS 3.11.4002</td>
</tr>
<tr>
<td>EDR switches</td>
<td>SB7700, SB7780, SB7800, CS7500, CS7510, CS7520</td>
<td>MLNX-OS 3.10.4400</td>
</tr>
</tbody>
</table>

1.3.2 System Requirements

1.3.2.1 Bare Metal Deployment Requirements

<table>
<thead>
<tr>
<th>Platform</th>
<th>Type and Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>OS</td>
<td>64-bit OS:</td>
</tr>
<tr>
<td>(Relevant for</td>
<td>• RedHat 8</td>
</tr>
<tr>
<td>Standalone</td>
<td>• RedHat 9</td>
</tr>
<tr>
<td>and High-</td>
<td>• Ubuntu 20.04</td>
</tr>
<tr>
<td>Availability</td>
<td>• Ubuntu 22.04</td>
</tr>
<tr>
<td>deployments)</td>
<td></td>
</tr>
</tbody>
</table>

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.
<table>
<thead>
<tr>
<th>Platform</th>
<th>Type and Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>x86_64</td>
</tr>
</tbody>
</table>
| HCAs     | • 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     | • MLNX_OFED 5.X  
          | • MLNX_OFED23.x  
          | • MLNX_OFED24.x  |

(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

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

Installation of UFM on minimal OS distribution is not supported.

UFM does not support systems in which NetworkManager service is enabled. Before installing UFM on RedHat OS, make sure to disable the service.

### 1.3.2.2 Docker Installation Requirements

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

The following operating systems were tested with Docker Container (as standalone container):

<table>
<thead>
<tr>
<th>Component</th>
<th>Type and Version</th>
</tr>
</thead>
</table>
| Supported OS | • RHEL8  
                   • RHEL9  
                   • Ubuntu18.04  
                   • Ubuntu20.04  
                   • Ubuntu22.04 |

For UFM Docker Container installation in HA mode, please refer to Bare Metal Deployment Requirements for the list of operating systems and kernels which support HA.
### 1.3.2.3 UFM Server Resource Requirements per Cluster Size

<table>
<thead>
<tr>
<th>Fabric Size</th>
<th>CPU Requirements*</th>
<th>Memory Requirements</th>
<th>Disk Space Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Minimum</td>
<td>Recommended</td>
</tr>
<tr>
<td>Up to 1000 nodes</td>
<td>4-core server</td>
<td>4 GB</td>
<td>20 GB</td>
</tr>
<tr>
<td>1000-5000 nodes</td>
<td>8-core server</td>
<td>16 GB</td>
<td>40 GB</td>
</tr>
<tr>
<td>5000-10000 nodes</td>
<td>16-core server</td>
<td>32 GB</td>
<td>80 GB</td>
</tr>
<tr>
<td>Above 10000 nodes</td>
<td>Contact NVIDIA Support</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### 1.3.2.4 UFM GUI Client Requirements

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

<table>
<thead>
<tr>
<th>Platform</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Browser</td>
<td>Edge, Internet Explorer, Firefox, Chrome, Opera, Safari</td>
</tr>
</tbody>
</table>
| Memory   | • Minimum: 8 GB  
|          | • Recommended: 16 GB |

### 1.3.2.5 MFT Package Version

<table>
<thead>
<tr>
<th>Platform</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>MFT</td>
<td>Integrated with MFT version mft-4.28.0-95</td>
</tr>
</tbody>
</table>

### 1.3.2.6 UFM SM Version

<table>
<thead>
<tr>
<th>Platform</th>
<th>Type and Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>SM</td>
<td>UFM package includes SM version 5.19.1</td>
</tr>
</tbody>
</table>

⚠️ 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.
1.3.2.7 UFM NVIDIA SHARP Software Version

<table>
<thead>
<tr>
<th>Platform</th>
<th>Type and Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>NVIDIA® Scalable Hierarchical Aggregation and Reduction Protocol (SHARP)™</td>
<td>UFM package includes NVIDIA SHARP software version 3.7.0</td>
</tr>
</tbody>
</table>

1.3.2.8 Used Ports by UFM Server

For a list of ports used by the UFM Server for internal and external communication, refer to Appendix - Used Ports.

1.3.3 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.

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 v6.17.0, it is possible to do so only from UFM Enterprise v6.16.0 or v6.15.0.

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.

1.4 Bug Fixes in This Release

<table>
<thead>
<tr>
<th>Ref #</th>
<th>Description</th>
<th>Keywords</th>
<th>Discovered in Release</th>
</tr>
</thead>
<tbody>
<tr>
<td>3912416</td>
<td>Description: Fixed issue with the authentication server being repeatedly restarted by the UFM health check after the default admin password is changed</td>
<td>Authentication, Server, Disable</td>
<td>v6.17.0</td>
</tr>
<tr>
<td>3863958</td>
<td>Description: Fixed issue where IB-IB go to INIT states due to failed UFM failover after enabling SHARP with PKeys</td>
<td>SHARP, PKey, IB-IB Link, Failover</td>
<td>v6.16.0</td>
</tr>
<tr>
<td>3850673</td>
<td>Description: Fixed issue where multiple ports go down simultaneously (link-downed counter increment)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ref #</td>
<td>Description</td>
<td>Keywords</td>
<td>Discovered in Release</td>
</tr>
<tr>
<td>----------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-----------------------------------------------</td>
<td>------------------------</td>
</tr>
<tr>
<td>3850217</td>
<td>Description: Fixed issues with ALM plugin (disabled handling topology changes and limited the number of trials of creating dynamic telemetry sessions by configurable variables)</td>
<td>Keywords: ALM Plugin</td>
<td>v6.15.1</td>
</tr>
<tr>
<td>3826544</td>
<td>Description: Fixed node info discovery issue</td>
<td>Keywords: Node, Info, Discovery</td>
<td>v6.15.0</td>
</tr>
<tr>
<td>3826069</td>
<td>Description: Fixed HCA port naming convention inconsistencies in UFM WebUI</td>
<td>Keywords: HCA port, Port name, WebUI</td>
<td>v6.16.0</td>
</tr>
<tr>
<td>3816196</td>
<td>Description: Fixed issue where UFM creates empty PKeys by UFM Rest API</td>
<td>Keywords: Empty, PKey, REST API</td>
<td>v6.15.2</td>
</tr>
<tr>
<td>3811475</td>
<td>Description: Fixed issue where UFM loggings REST API omits additional contents of the log when it spans over multiple lines</td>
<td>Keywords: UFM Loggings, REST API, Span over, Multiple Lines</td>
<td>v6.15.3</td>
</tr>
<tr>
<td>3803527</td>
<td>Description: Fixed issue with Create History REST API while collecting SM Logs Error</td>
<td>Keywords: Create History, SM Log Error</td>
<td>v6.15.3</td>
</tr>
<tr>
<td>3752196</td>
<td>Description: Fixed intermittent UFM REST API Failures</td>
<td>Keywords: UFM REST API, Failures</td>
<td>v6.16.0</td>
</tr>
<tr>
<td>3864876</td>
<td>Description: Fixed issue with UFM events not appearing in remote syslog</td>
<td>Keywords: UFM Events, Remote syslog</td>
<td>v6.15.1</td>
</tr>
<tr>
<td>3809574</td>
<td>Description: Fixed WebUI issues in the “Power” column</td>
<td>Keywords: WebUI, Power</td>
<td>v6.16.0</td>
</tr>
<tr>
<td>3766079</td>
<td>Description: Fixed issue with UFM not showing SSH user/pass tab</td>
<td>Keywords: SSH, User/Pass Tab</td>
<td>v6.15.0</td>
</tr>
<tr>
<td>3916656</td>
<td>Description: Fixed issue with releasing lock without acquiring when handling MC join requests from unknown source.</td>
<td>Keywords:</td>
<td></td>
</tr>
<tr>
<td>Ref #</td>
<td>Description</td>
<td></td>
<td></td>
</tr>
<tr>
<td>----------</td>
<td>-----------------------------------------------------------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3859362</td>
<td><strong>Description:</strong> UFM TFS endpoint dashboard report Switch port TX/RX rate reach Tbps</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Keywords:</strong> TFS, Switch Port, TX/RX</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Workaround:</strong> N/A</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Discovered in Release:</strong> v6.15.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3881365</td>
<td><strong>Description:</strong> Malfunctioning of the rest API when deleting port associated to a pkey</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Keywords:</strong> CloudX, API, Bare-Metal</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Workaround:</strong> N/A</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Discovered in Release:</strong> v6.15.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3862847</td>
<td><strong>Description:</strong> UFM reports wrong cable length for NDR optical cables connected to Quantum-2 NDR switch</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Keywords:</strong> NDR, Optical Cables, Quantum-2, Switch</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Workaround:</strong> N/A</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Discovered in Release:</strong> v6.17.0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

1.5 Known Issues in This Release

For a list of known issues from previous releases, please refer to [Known Issues History](#).

1.6 Changes and New Features History

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syslog Streaming</td>
<td>Added the option for setting UFM syslog streaming facility. For more information, refer to <a href="#">Configuring Syslog</a>.</td>
</tr>
<tr>
<td>Switch Cables REST API</td>
<td>Added the option to query specific switch cables (using Ports API).</td>
</tr>
<tr>
<td>Switch Power Information</td>
<td>Added support for switch and modules power usage data in UFM telemetry and REST API. For more information, refer to <a href="#">Devices Window</a> and <a href="#">Inventory Window</a>.</td>
</tr>
<tr>
<td>Feature</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>UFM Data Streaming</td>
<td>Added the ability to change the UFM Data streaming log facility. For more information, refer to Configuring Syslog and Configuring UFM Logging.</td>
</tr>
<tr>
<td>Kerberos Authentication</td>
<td>Added the ability for Kerberos authentication, a strong network authentication protocol for client-server applications. For more information, refer to Kerberos Authentication and Enabling Kerberos Authentication.</td>
</tr>
<tr>
<td>SM Settings</td>
<td>Changed the default maximal number of VLS to 2 (VL0–VL1). For more information, refer to Appendix - UFM Subnet Manager Default Properties.</td>
</tr>
<tr>
<td>Cable Management</td>
<td>Added support for showing transceiver information for downed links. For more information, refer to Cables Window and Network Map.</td>
</tr>
<tr>
<td>Secondary Telemetry</td>
<td>Added the <code>secondary_slvl_support</code> flag and information on the default counters. For more information, refer to Secondary Telemetry.</td>
</tr>
<tr>
<td>Congestion Control</td>
<td>Added support for SM congestion control settings. For more information, refer to Appendix - OpenSM Configuration Files for Congestion Control.</td>
</tr>
<tr>
<td>UFM HA</td>
<td>Enhanced reliability and added support for setting UFM HA on LVM (Logical Volume Manager). For more information, refer to UFM High-Availability Documentation.</td>
</tr>
<tr>
<td>Plugins</td>
<td>Packet Mirroring Collector (PMC) Plugin: Added support for event on PF indicating a QP closing with error on any other GVMI/VF. For more information, refer to Packet Mirroring Collector (PMC) Plugin.</td>
</tr>
<tr>
<td></td>
<td>PDR Deterministic Plugin: Updated instructions. For more information, refer to PDR Deterministic Plugin.</td>
</tr>
<tr>
<td></td>
<td>GNMI-Telemetry Plugin: Added gNMI telemetry streaming support (supporting secured mode streaming). For more information, refer to GNMI-Telemetry Plugin.</td>
</tr>
<tr>
<td></td>
<td>NDT Plugin (Subnet Merger): Added the option to validate the extended fabric using cable validation tool. For more information, refer to the NDT Plugin.</td>
</tr>
<tr>
<td></td>
<td><strong>Rev 6.15.2</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Rev 6.15.1</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Rev 6.15.0</strong></td>
</tr>
<tr>
<td>Defining Node Description</td>
<td>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 HCA s during host startup. For more information, refer to Defining Node Description Black-List.</td>
</tr>
<tr>
<td>Network Reports</td>
<td>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.</td>
</tr>
<tr>
<td>User Authentication</td>
<td>Introduced a new user authentication login page. For more information, refer to Azure Authentication Login Page and Enabling Azure AD Authentication.</td>
</tr>
<tr>
<td></td>
<td>Added support for a separate authentication server. For more information, refer to UFM Authentication Server and Enabling UFM Authentication Server.</td>
</tr>
<tr>
<td>Secondary Telemetry</td>
<td>Added the ability to expose SHARP telemetry in UFM Telemetry. For more information, refer to Exposing Switch Aggregation Nodes Telemetry.</td>
</tr>
<tr>
<td></td>
<td>Added the ability to stop SHARP telemetry endpoint using CLI commands. For more information, refer to Stopping Telemetry Endpoint Using CLI Command.</td>
</tr>
<tr>
<td>REST APIs</td>
<td>Enhanced the logging REST API by adding the ability to get event logs in JSON file format. For more information, refer to <a href="#">Get Events Logs in JSON Format</a>.</td>
</tr>
<tr>
<td>-----------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>Added the ability to expose managed switch power consumption in Web UI. For more information, refer to <a href="#">Get Managed Switches Power Consumption</a>.</td>
</tr>
<tr>
<td></td>
<td>Added ability to filter the event logs by source. For more information, refer to <a href="#">Create Log History</a>.</td>
</tr>
<tr>
<td></td>
<td>Added the ability to generate enterprise network reports. For more information, refer to <a href="#">Events History</a>, <a href="#">Device Status Events</a> and <a href="#">Link Status Events</a>.</td>
</tr>
<tr>
<td></td>
<td>Introduced REST APIs for various authentication types. For more information, refer to <a href="#">Examples of REST APIs Using Various Authentication Types</a>.</td>
</tr>
<tr>
<td></td>
<td>Added the ability to update UFM Configuration REST API. For more information, refer to <a href="#">UFM Configuration REST API</a>.</td>
</tr>
<tr>
<td></td>
<td>Added the option to expose cable information. For more information, refer to <a href="#">Get Ports with Cable Information</a>.</td>
</tr>
<tr>
<td></td>
<td>Improved dynamic telemetry by adding the ability to instantiate a new instance and delete a running instance. For more information, refer to <a href="#">UFM Dynamic Telemetry Instances REST API</a>.</td>
</tr>
<tr>
<td></td>
<td>Added the option to set “down” ports as unhealthy. For more information, refer to <a href="#">Unhealthy Ports REST API</a>.</td>
</tr>
<tr>
<td></td>
<td>Added forge InfiniBand anti-spoofing support. For more information, refer to <a href="#">Forge InfiniBand Anti-Spoofing REST API</a>.</td>
</tr>
<tr>
<td></td>
<td>Added the ability to expose the &quot;site_name&quot; field in all supported REST APIs. For more information, refer to <a href="#">REST API Complementary Information</a>.</td>
</tr>
<tr>
<td>Plugins</td>
<td>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 <a href="#">GNMI-Telemetry Plugin</a>.</td>
</tr>
<tr>
<td></td>
<td>Added support for ALM configuration for controlling isolation/de-isolation. For more information, refer to <a href="#">ALM Configurations</a>.</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td>Supported Transceivers</td>
<td>Added support for FR4 transceivers</td>
</tr>
<tr>
<td>Transceivers Burning</td>
<td>UFM supports second-source cable transceivers burn.</td>
</tr>
<tr>
<td>Module REST API</td>
<td>Added HW revision field in GET module REST API response.</td>
</tr>
<tr>
<td>Telemetry</td>
<td>Added support for the MRCS register read in UFM Telemetry.</td>
</tr>
<tr>
<td>UFM Reports</td>
<td>UFM Daily report will be disabled by default after upgrade or clean installation.</td>
</tr>
<tr>
<td>Rev 6.14.2</td>
<td></td>
</tr>
<tr>
<td>Cable and Transceivers Burning</td>
<td>UFM supports second-source cable transceivers burn.</td>
</tr>
<tr>
<td>Module REST API</td>
<td>Added HW revision field in GET module REST API response.</td>
</tr>
<tr>
<td>Telemetry</td>
<td>Added support for the MRCS register read in UFM Telemetry.</td>
</tr>
<tr>
<td>UFM Reports</td>
<td>UFM Daily report will be disabled by default after upgrade or clean installation.</td>
</tr>
<tr>
<td>Rev 6.14.0</td>
<td></td>
</tr>
<tr>
<td>UFM Upgrade</td>
<td>Added support for in-service upgrade procedure for UFM HA. Refer to the following sections: <a href="#">Upgrading UFM on Bare Metal - High Availability Upgrade</a>, <a href="#">Upgrading UFM Container in High Availability Mode</a></td>
</tr>
<tr>
<td>User Authorization</td>
<td>Added support for user-defined roles based on REST APIs subsets. Refer to <a href="#">Rest Roles Access Control</a>.</td>
</tr>
<tr>
<td>Feature</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------</td>
<td>------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>User Authentication</strong></td>
<td>Added support for user authentication based on Azure Active Directory. Refer to Azure AD</td>
</tr>
<tr>
<td><strong>Plugins Management</strong></td>
<td>Added support for loading UFM plugin to both master and standby nodes in case of UFM HA deployment. Refer to Plugin Management.</td>
</tr>
<tr>
<td><strong>Unhealthy Ports Policy Management</strong></td>
<td>Added support for unhealthy ports policy management via UFM Web UI. Refer to Health Policy Management.</td>
</tr>
<tr>
<td><strong>REST over RDMA Plugin</strong></td>
<td>Added support for remote ibdiagnet authentication. Refer to rest-rdma Plugin.</td>
</tr>
<tr>
<td><strong>SHARP Reservation</strong></td>
<td>Added support for synchronous SHARP reservation REST API (in addition to the existing asynchronous REST API). Refer to NVIDIA SHARP REST API.</td>
</tr>
<tr>
<td><strong>Secondary Telemetry</strong></td>
<td>Added support for secondary telemetry running by default upon UFM startup, fetching NVIDIA Amber counters. Refer to Secondary Telemetry.</td>
</tr>
<tr>
<td><strong>PCI Analysis</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>UFM System Dump</strong></td>
<td>Added human readable time to the dmsg de-message output as part of UFM system dump.</td>
</tr>
<tr>
<td><strong>Factory Reset</strong></td>
<td>Added support for UFM Factory Reset. Refer to UFM Factory Reset.</td>
</tr>
<tr>
<td><strong>Network Fast Recovery</strong></td>
<td>Added the ability to automatically isolate a malfunctioning switch port as detected by the switch. Refer to Enabling Network Fast Recovery</td>
</tr>
<tr>
<td><strong>Multi-Subnet UFM</strong></td>
<td>Added support for multiple UFM instances, wherein multiple instances are aggregated, managed and controlled by a centralized UFM instance. Refer to Multi-Subnet UFM.</td>
</tr>
<tr>
<td><strong>Switch ASIC Failure Detection</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>UFM High-Availability Enhancements</strong></td>
<td>Added support for configuring high-availability with dual-link connections to improve the high-availability robustness.</td>
</tr>
<tr>
<td><strong>Automatic Switch Grouping</strong></td>
<td>Added support for enabling automatic grouping of 1U switches by UFM, as per a pre-defined user-configured mapping. Refer to Appendix - Switch Grouping.</td>
</tr>
<tr>
<td><strong>SHARP Trees APIs</strong></td>
<td>Incorporated support for a new UFM REST API that presents the current active SHARP trees. Refer to NVIDIA SHARP Resource Allocation REST API.</td>
</tr>
<tr>
<td><strong>SHARP Reservation APIs</strong></td>
<td>Added support for SHARP Reservation API enhancements. Refer to NVIDIA SHARP Resource Allocation REST API.</td>
</tr>
<tr>
<td><strong>Operating System Update support</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>Email Time-Zone Settings</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>Switch Connectivity Failure Indication</strong></td>
<td>Incorporated support for a new UFM event indication that identifies failed communication with a specified managed switch. Refer to Appendix - Supported Port Counters and Events.</td>
</tr>
<tr>
<td>Feature</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Dynamic Telemetry</td>
<td>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.</td>
</tr>
<tr>
<td>TFS (Telemetry Fluent Streaming) Plugin</td>
<td>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.</td>
</tr>
<tr>
<td>DTS (DPU Telemetry) Plugin</td>
<td>Added support for displaying DPUs data within the UFM Web UI. Refer to DTS Plugin.</td>
</tr>
<tr>
<td>Cyber-AI Plugin</td>
<td>Added support for displaying Cyber-AI software within the UFM Web UI. Refer to UFM Cyber-AI Plugin.</td>
</tr>
<tr>
<td>Packet Mirroring Collector (PMC) Plugin</td>
<td>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.</td>
</tr>
<tr>
<td>SNMP Traps Listener Plugin</td>
<td>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.</td>
</tr>
<tr>
<td>Bright Cluster Integration Plugin</td>
<td>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.</td>
</tr>
<tr>
<td>UFM System Dump</td>
<td>UFM System Dump collection enhancement. Refer to UFM System Dump Tab.</td>
</tr>
<tr>
<td>Expanding Non-Blocking Fabric (NDT Plugin extension)</td>
<td>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.</td>
</tr>
<tr>
<td>PDR (Packet Drop Rate) Plugin</td>
<td>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.</td>
</tr>
</tbody>
</table>

**Rev 6.12.0**

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Managed Switches - Sysinfo Mechanism</td>
<td>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</td>
<td></td>
</tr>
<tr>
<td>REST over RDMA Plugin</td>
<td>Introduced security improvements (allowed read-only options in remote ibdiagnet) and added support for Telemetry API. Refer to rest-rdma Plugin.</td>
<td></td>
</tr>
<tr>
<td>Events and Notifications</td>
<td>Added support for indicating potential switch ASIC failure by detecting a defined percentage of unhealthy switch ports. Refer to Additional Configuration (Optional)</td>
<td></td>
</tr>
<tr>
<td>SHARP AM Multi-Port</td>
<td>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</td>
<td></td>
</tr>
<tr>
<td>UFM System Dump</td>
<td>Added support for downloading the generated UFM system dump. Refer to UFM System Dump Tab</td>
<td></td>
</tr>
<tr>
<td>UFM REST API</td>
<td>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</td>
<td></td>
</tr>
<tr>
<td>UFM System Dump Improvements</td>
<td>UFM System Dump Improvements including Creating New System Dump API</td>
<td></td>
</tr>
<tr>
<td>Feature</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>----------------------------------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>UFM SLURM Integration</td>
<td>Enhanced UFM SLURM integration; allow flexible configuration of PKey and SHARP resources usage. Refer to <a href="#">Appendix - UFM SLURM Integration</a>.</td>
<td></td>
</tr>
<tr>
<td>UFM HA</td>
<td>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 <a href="#">Configuring UFM Docker in HA Mode</a> and <a href="#">Installing UFM Server Software for High Availability</a>.</td>
<td></td>
</tr>
<tr>
<td>Managed Switch Operations</td>
<td>Added support for persistent enablement/disability of managed switches' ports. Refer to <a href="#">Ports Window</a>.</td>
<td></td>
</tr>
<tr>
<td>UFM SDK</td>
<td>Created a script to get TopX data by category. Refer to <a href="#">UFM Aggregation TopX README.md</a>.</td>
<td></td>
</tr>
<tr>
<td>Proxy Authentication</td>
<td>Added option to delegate authentication to a proxy. Refer to <a href="#">Delegate Authentication to a Proxy</a>.</td>
<td></td>
</tr>
<tr>
<td>UFM Initial Settings</td>
<td>Removed the requirement to set the IPoIB address to the main IB interface used by UFM/SM (gv.cfg → fabric_interface).</td>
<td></td>
</tr>
<tr>
<td>Port auto-isolation</td>
<td>Symbol BER <strong>warning</strong> does not trigger port auto-isolation, only symbol BER <strong>error</strong></td>
<td></td>
</tr>
<tr>
<td>MFT Package</td>
<td>Integrated with MFT version 4.23.0-104</td>
<td></td>
</tr>
<tr>
<td><strong>Rev 6.11.0</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UFM Discovery and Device Management</td>
<td>• InBand autos discovery of switches' IP addresses using <code>ibdiagnet</code></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Discovering the device's PSID and FW version using <code>ibdiagnet</code> by default instead of using an SM vendor plugin</td>
<td></td>
</tr>
<tr>
<td>CPU Affinity</td>
<td>Enabling the user to control CPU affinity of UFM's major processes</td>
<td></td>
</tr>
<tr>
<td>gRPC API</td>
<td>Added support for streaming UFM REST API data over gRPC as part of new UFM plugin. Refer to <a href="#">GRPC-Streamer Plugin</a>.</td>
<td></td>
</tr>
<tr>
<td>Telemetry</td>
<td>• Added support for flexible counters infrastructure (ability to change counter sets that are sampled by the UFM)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 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 <a href="#">Secondary Telemetry</a>.</td>
<td></td>
</tr>
<tr>
<td>EFS UFM Plugin</td>
<td>Added support for streaming UFM events data to FluentD destination as part of a new UFM plugin. Refer to <a href="#">UFM Telemetry Fluentd Streaming (TFS) Plugin</a>.</td>
<td></td>
</tr>
<tr>
<td>General UI Enhancements</td>
<td>• Displayed columns of all tables are persistent per user, with the option to restore defaults. Refer to <a href="#">Displayed Columns</a>.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Improved look and feel in Network Map. Refer to <a href="#">Network Map</a>.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Added Reveal Uptime to the general tab in the devices information tabs. Refer to <a href="#">Device General Tab</a>.</td>
<td></td>
</tr>
<tr>
<td>High Availability Deployment</td>
<td>• 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 <a href="#">Installing UFM Server Software for High Availability</a>.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Changed UFM HA package installation command parameters. For more information, refer to <a href="#">Installing UFM Server Software for High Availability</a>.</td>
<td></td>
</tr>
<tr>
<td>REST APIs</td>
<td>Added support for PKey filtering for default session data. Refer to <a href="#">Get Default Monitoring Session Data by PKey Filtering</a>.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Added support for filtering session data by groups. Refer to <a href="#">Monitoring Sessions REST API</a>.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Added support for resting all unhealthy ports at once. Refer to <a href="#">Mark All Unhealthy Ports as Healthy at Once</a>.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Added support for presenting system uptime in UFM REST API. Refer to <a href="#">Systems REST API</a>.</td>
<td></td>
</tr>
</tbody>
</table>
**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)

---

<table>
<thead>
<tr>
<th>System health enhancements</th>
<th>Add support for the periodic fabric health report, and reflected the ports’ results in UFM's dashboard</th>
</tr>
</thead>
<tbody>
<tr>
<td>UFM Plugins Management</td>
<td>Add support for plugin management via UFM web UI</td>
</tr>
<tr>
<td>UFM Extended Status</td>
<td>- Add support for showing UFM's current processes status (via shell script)</td>
</tr>
<tr>
<td></td>
<td>- Added REST API for exposing UFM readiness</td>
</tr>
<tr>
<td>Failover to Other Ports</td>
<td>Add support for SM and UFM Telemetry failover to other ports on the local machine</td>
</tr>
<tr>
<td>UFM Appliance Upgrade</td>
<td>Added a set of REST APIs for supporting the UFM Appliance upgrade</td>
</tr>
<tr>
<td>Configuration Audit</td>
<td>Add support for tracking changes made in major UFM configuration files (UFM, SM, SHARP, Telemetry)</td>
</tr>
<tr>
<td>UFM Plugins</td>
<td>Add support for new SDK plugins</td>
</tr>
<tr>
<td>Telemetry</td>
<td>Add support for statistics processing based on UFM telemetry csv format</td>
</tr>
<tr>
<td>UFM High Availability Installation</td>
<td>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</td>
</tr>
</tbody>
</table>

---

### 1.7 Bug Fixes History

<table>
<thead>
<tr>
<th>Ref. #</th>
<th>Description</th>
<th>Keywords</th>
<th>Discovered in Release</th>
</tr>
</thead>
<tbody>
<tr>
<td>37549</td>
<td>Fixed issue where following the UFM HA upgrade from version 5.0.1-2 to version 5.3.1-2, the <code>ufm_ha_cluster_config</code> command wiped the root partition</td>
<td>UFM HA Upgrade, <code>ufm_ha_cluster_config</code>, Root Partition, Wipe</td>
<td>6.15.2</td>
</tr>
<tr>
<td>37521</td>
<td>Fixed intermittent UFM REST API failures</td>
<td>REST API, Failure</td>
<td>6.15.1</td>
</tr>
<tr>
<td>37588</td>
<td>Fixed <code>manage_the_unmanaged</code> tool failure</td>
<td>manage_the_unmanaged, Failure</td>
<td>6.15.2</td>
</tr>
<tr>
<td>Revision</td>
<td>Description</td>
<td>Keywords</td>
<td>Discovered in Release</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
<td>----------</td>
<td>-----------------------</td>
</tr>
<tr>
<td>Rev 6.15.1</td>
<td>Fixed the issue in congestion control, where <em>cc-policy.conf</em> file remains unchanged following the upgrade of the container version (with no changes made by the user)</td>
<td>Congestion Control, cc-policy.conf, Upgrade, Container</td>
<td>6.16.0-4</td>
</tr>
<tr>
<td>36701 83</td>
<td>Monitoring endpoint not returning counters for an active interface</td>
<td>Monitoring, Active Interface, Counters</td>
<td>v6.15.0</td>
</tr>
<tr>
<td>36701 82</td>
<td>Inconsistent port format type returned from the UFM</td>
<td>Inconsistent, Port, Format Type</td>
<td>v6.14.1</td>
</tr>
<tr>
<td>36669 44</td>
<td>Port auto isolation failed to activate when a port consistently exhibited a high Symbol BER (1e-7)</td>
<td>Port Auto Isolation, Symbol BER</td>
<td>v6.13.1</td>
</tr>
<tr>
<td>36653 16</td>
<td>The UFM REST API endpoint <code>/ufmRest/resources/ports</code> provide inaccurate port state information</td>
<td>Ports REST API, Port State</td>
<td>v6.14.1</td>
</tr>
<tr>
<td>36041 94</td>
<td>UFM Fabric Validation &quot;$CheckPortCounters&quot; failure</td>
<td>Fabric Validation, CheckPortCounters</td>
<td>v6.13.2</td>
</tr>
<tr>
<td>Rev 6.15.0</td>
<td>UFM Web UI does not display Network Map (stuck with &quot;please wait&quot; message)</td>
<td>Web UI, Network Map</td>
<td>v6.14.1</td>
</tr>
<tr>
<td>36445 53</td>
<td>When querying the ports, adding a <code>cable_info=true</code> as an argument will give cable information per port</td>
<td>Ports, Query, cable_info=true</td>
<td>v6.14.0</td>
</tr>
<tr>
<td>36042 12</td>
<td>Broken links REST API</td>
<td>REST API, Broken link</td>
<td>v6.13.2</td>
</tr>
<tr>
<td>36041 83</td>
<td>UFM error UFM NOT performed OpenSM polling for fabric changes more than 230742 seconds</td>
<td>OpenSM, UFM Error</td>
<td>v6.13.2-5</td>
</tr>
<tr>
<td>Issue</td>
<td>Description</td>
<td>Keywords</td>
<td>Discovered in release</td>
</tr>
<tr>
<td>-------</td>
<td>-------------</td>
<td>----------</td>
<td>-----------------------</td>
</tr>
<tr>
<td>35878</td>
<td>OpenSM restarted when backup UFM lost power</td>
<td>OpenSM, Restart</td>
<td>v6.9</td>
</tr>
<tr>
<td>35774</td>
<td>UFM REST API returns wrong switch type for NDR unmanaged switch</td>
<td>Unmanaged Switch, NDR, REST API</td>
<td>v6.13.1</td>
</tr>
<tr>
<td>35758</td>
<td>UFM event is not generated for a switch down</td>
<td>UFM Event, Switch Down</td>
<td>v6.13.1</td>
</tr>
<tr>
<td>36284</td>
<td>UFM Web UI timezone issue when selecting Local Time</td>
<td>Timezone, Web UI, Local Time</td>
<td>v6.14.1-5</td>
</tr>
<tr>
<td>35658</td>
<td>UFM container CLI bugs</td>
<td>CLI, Container</td>
<td>v6.13.2-5</td>
</tr>
</tbody>
</table>

---

**Rev 6.14.0**

<table>
<thead>
<tr>
<th>Issue</th>
<th>Description</th>
<th>Keywords</th>
<th>Discovered in release</th>
</tr>
</thead>
<tbody>
<tr>
<td>35907</td>
<td>After upgrading UFM new telemetry data is not being collected and presented in UI Telemetry tab.</td>
<td>Telemetry, Coredump</td>
<td>6.14.0</td>
</tr>
</tbody>
</table>

---

**Rev 6.13.2**

<table>
<thead>
<tr>
<th>Issue</th>
<th>Description</th>
<th>Keywords</th>
<th>Discovered in Release</th>
</tr>
</thead>
<tbody>
<tr>
<td>32288</td>
<td>ufm-prolog.sh failure: hostnames are not found in the fabric after reboot</td>
<td>Hostnames, <code>ufm-prolog.sh</code>, reboot</td>
<td>6.10.0</td>
</tr>
<tr>
<td>34956</td>
<td>UFM Enterprise v6.13.1 server hangs intermittently, blocking UFM REST server, and UFM GUI</td>
<td>UFM REST, UFM GUI</td>
<td>6.13.1</td>
</tr>
</tbody>
</table>

---

N/A | Reverted setGuidsForPkey APIs for supporting SHARP reservation (in case it is enabled) | setGuidsForPkey, SHARP Reservation | 6.13.1 |
### Rev 6.13.1

<table>
<thead>
<tr>
<th>ID</th>
<th>Description</th>
<th>Keywords</th>
<th>Discovered in Release</th>
</tr>
</thead>
<tbody>
<tr>
<td>3459431</td>
<td>Description: UFM System Dump cannot be extracted from UFM 3.0 Enterprise Appliance host when running in high-availability mode.</td>
<td>System Dump, High-Availability</td>
<td>6.12.0</td>
</tr>
<tr>
<td>3461658</td>
<td>Description: The network fast recovery configuration (<code>/opt/ufm/files/conf/opensm/fast_recovery.conf</code>) is missing when UFM is deployed in Docker Container mode.</td>
<td>Network Fast Recovery; Docket Container; Missing Configuration</td>
<td>6.12.0</td>
</tr>
<tr>
<td>3461058</td>
<td>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</td>
<td>Dynamic, Telemetry, Log-rotate</td>
<td>6.13.0</td>
</tr>
</tbody>
</table>

### Rev 6.13.0

<table>
<thead>
<tr>
<th>ID</th>
<th>Description</th>
<th>Keywords</th>
<th>Discovered in Release</th>
</tr>
</thead>
<tbody>
<tr>
<td>3410826</td>
<td>Description: Rectified inability to modify user password</td>
<td>User Password, Update, Fail</td>
<td>6.12.1</td>
</tr>
<tr>
<td>3383916</td>
<td>Description: Fixed Client CTRL+C server disruption</td>
<td>Client CTRL+C, Server functionality</td>
<td>Rest Over RDMA Image 1.0.0-21</td>
</tr>
<tr>
<td>3375414</td>
<td>Description: Fixed improper functionality of UFM UI Dashboard</td>
<td>UI Dashboard</td>
<td>6.11.0</td>
</tr>
<tr>
<td>3342713</td>
<td>Description: Fixed UFM Health configuration for periodic restarts of the telemetry</td>
<td>UFM Health, Telemetry, Periodic restarts</td>
<td>6.11.1</td>
</tr>
<tr>
<td>3361160</td>
<td>Description: Fixed UFM long upgrade time due to a large historical Telemetry database file</td>
<td>Long Upgrade Time, Historical Telemetry, Database File</td>
<td>6.11.0</td>
</tr>
<tr>
<td>3268270</td>
<td>Description: Show managed switches inventory data (Sysinfo) immediately after UFM initialization</td>
<td>Managed Switches, Inventory, Sysinfo</td>
<td>6.11.0</td>
</tr>
<tr>
<td>3338613</td>
<td>Description: Fixed UFM log rotation for supported Ubuntu OSs</td>
<td>Log rotation, Ubuntu</td>
<td>6.11.0</td>
</tr>
<tr>
<td>3338600</td>
<td>Description: Fixed UFM UI lockdown by adding protection to the failed path on backend side</td>
<td>UFM UI, lockdown</td>
<td>6.11.0</td>
</tr>
<tr>
<td>ID</td>
<td>Description</td>
<td>Keywords</td>
<td>Discovered in Release</td>
</tr>
<tr>
<td>-------</td>
<td>------------------------------------------------------------------------------</td>
<td>-----------------------------------------------</td>
<td>-----------------------</td>
</tr>
<tr>
<td>32761</td>
<td>Description: Fixed remote syslog configuration in UFM Web UI to be persistent</td>
<td>Remote Syslog, Web UI</td>
<td>6.11.0</td>
</tr>
<tr>
<td>32340</td>
<td>Description: UFM WebUI unresponsive after failover issue</td>
<td>UFM, WebUI, failover</td>
<td>6.10.0</td>
</tr>
<tr>
<td>31995</td>
<td>Description: Incorrect Tier reporting in the UFM events</td>
<td>Tier, Incorrect Report</td>
<td>6.10.0</td>
</tr>
<tr>
<td>31070</td>
<td>Description: Using GET All Modules REST API (GET /ufmRest/resources/modules), returns N/A in device_name.</td>
<td>Modules, N/A, device_name</td>
<td>6.9</td>
</tr>
<tr>
<td>30768</td>
<td>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 &quot;invalid date.&quot;</td>
<td>WEB UI, &quot;invalid date&quot;</td>
<td>6.9</td>
</tr>
<tr>
<td>30601</td>
<td>Description: UFM WEB UI - Ports REST API returns tier parameters as N/A in response</td>
<td>WEB UI, tier, N/A</td>
<td>6.9</td>
</tr>
<tr>
<td>30526</td>
<td>Description: UFM monitoring mode is not working</td>
<td>Monitoring, mode</td>
<td>6.9</td>
</tr>
<tr>
<td>30311</td>
<td>Description: Network map showing a link between QM8790 and Manta Ray leaf having BW of &gt;20,000 Gb/s</td>
<td>Network Map, BW, 20,000</td>
<td>6.8.0</td>
</tr>
<tr>
<td>30033</td>
<td>Description: UFM Starting and Stopping On Its Own Since Merge</td>
<td>Start, Stop</td>
<td>6.7.0</td>
</tr>
<tr>
<td>29682</td>
<td>Description: Fabric health Old Alerts and events do not clear</td>
<td>Fabric Health, Alerts, clear</td>
<td>6.8.0</td>
</tr>
<tr>
<td>29579</td>
<td>Description: BER Not Being Read or Reported</td>
<td>BER, Not, Reported</td>
<td>6.8.0</td>
</tr>
<tr>
<td>30322</td>
<td>Description: UFM UFMAPL_4.7.0.3_UFM_6.8.0.6 lists one of my skyways as “host” instead of “gateway”</td>
<td>skyway, gateway, host</td>
<td>6.8.0</td>
</tr>
<tr>
<td>ID</td>
<td>Description</td>
<td>Keywords</td>
<td>Date</td>
</tr>
<tr>
<td>------</td>
<td>------------------------------------------------------------------------------</td>
<td>-----------------------------------</td>
<td>------------</td>
</tr>
<tr>
<td>29664</td>
<td>Description: UFM Fabric health BER_CHECK warnings</td>
<td>Fabric Health, BER, check</td>
<td>6.8.0</td>
</tr>
<tr>
<td>28012</td>
<td>Description: UFM failed to serve incoming REST API requests</td>
<td>REST API, hang, unresponsive</td>
<td>6.7.0</td>
</tr>
<tr>
<td>27820</td>
<td>Description: UFM APL 4.6 BER not reported (None) in event logs</td>
<td>BER, events, log</td>
<td>6.7.0</td>
</tr>
<tr>
<td>27447</td>
<td>Description: UFM health test: CheckSMConnectivityOnStandby should consider multiple GUIDs on a port</td>
<td>UFM Health, SM connectivity, multiple guids</td>
<td>6.7.0</td>
</tr>
<tr>
<td>28302</td>
<td>Description: UFM (container) is not starting after server reboot</td>
<td>UFM Container, reboot</td>
<td>6.7.0</td>
</tr>
<tr>
<td>28048</td>
<td>Description: UFM WEB GUI becomes Unresponsive and Event/REST API log stops printing</td>
<td>Web UI, unresponsive</td>
<td>6.7.0</td>
</tr>
<tr>
<td>26993</td>
<td>Description: IPMI console login connects to CentOS (UM docker OS) instead of Ubuntu (host OS) after UFM docker installation.</td>
<td>IPMI; CentOS; Login</td>
<td>6.6.1</td>
</tr>
<tr>
<td>26380</td>
<td>Description: Wrong module (line/spine) label appears in effective BER event.</td>
<td>Module; Effective; BER; Event</td>
<td>6.4.1</td>
</tr>
<tr>
<td>26186</td>
<td>Description: UFM failover is not working when bond0 is configured with IPoIB.</td>
<td>Failover, Bond; IPoIB</td>
<td>6.6.1</td>
</tr>
<tr>
<td>26155</td>
<td>Description: UFM software no longer supports license type 'UFM APPLIANCE'.</td>
<td>License; UFM Appliance</td>
<td>6.5.2</td>
</tr>
<tr>
<td>25896</td>
<td>Description: UFM stopped to discover topology on SuperPOD environment.</td>
<td>Stopped; discover</td>
<td>6.5.2</td>
</tr>
<tr>
<td>23351</td>
<td>Description: Memory leak discovered in ModelMain.py process.</td>
<td>Memory leak</td>
<td>6.5.1</td>
</tr>
<tr>
<td>ID</td>
<td>Description</td>
<td>Keywords</td>
<td>Discovered in Release</td>
</tr>
<tr>
<td>-------</td>
<td>-----------------------------------------------------------------------------</td>
<td>-----------------------------------------------</td>
<td>-----------------------</td>
</tr>
<tr>
<td>23000</td>
<td>CMP python error</td>
<td>Python, error</td>
<td>6.5.1</td>
</tr>
<tr>
<td>23736</td>
<td>UFM license check of UFM permanent license generates invalid license status</td>
<td>Permanent license, UFM health report</td>
<td>6.5.1</td>
</tr>
<tr>
<td>21257</td>
<td>Some commands appear for users with monitor privileges which are not functional. It is recommended not to use this user role.</td>
<td>Monitor, permissions, user</td>
<td>4.2.0</td>
</tr>
<tr>
<td>19673</td>
<td>Email sender address cannot contain more than one period (&quot;.&quot;) in the domain name.</td>
<td>Email, sender, period</td>
<td>6.3</td>
</tr>
<tr>
<td>20694</td>
<td>SMTP server username cannot have more than 20 characters.</td>
<td>Email</td>
<td>6.3</td>
</tr>
<tr>
<td>19143</td>
<td>MellanoxCare service can now communicate with UFM (valid only when http communication is configured between MCare and UFM).</td>
<td>MellanoxCare, http, https</td>
<td>6.2</td>
</tr>
<tr>
<td>17830</td>
<td>Opening UFM web UI in monitoring mode is now supported.</td>
<td>Web UI, monitoring mode</td>
<td>6.2</td>
</tr>
<tr>
<td>ID</td>
<td>Description</td>
<td>Keywords</td>
<td>Discovered in</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
<td>----------------------------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>16918</td>
<td>UFM Agent now is now part of the UFM web UI.</td>
<td>UFM Agent</td>
<td>6.1</td>
</tr>
<tr>
<td>17932</td>
<td>UFM/module temperature thresholds notifications.</td>
<td>Temperature thresholds</td>
<td>6.1</td>
</tr>
<tr>
<td>16786</td>
<td>Fixed an issue where UFM HA prerequisite script was checking for wrong Virtual IP port argument.</td>
<td>UFM HA, prerequisite, Virtual IP, port</td>
<td>6.1</td>
</tr>
<tr>
<td>17062</td>
<td>Fixed an issue where MLNX_OS credentials were missing at the device &quot;access_credentials&quot; menu (the issue was detected on old Java based GUI). At the new UFM Web UI - MLNX_OS credentials are represented by HTTP credentials.</td>
<td>MLNX_OS, credentials</td>
<td>6.1</td>
</tr>
<tr>
<td>14865</td>
<td>Fixed an issue where CentOS 7.5 was not recognized as RHEL 7 flavor upon installation.</td>
<td>Installation, CentOS, RHEL</td>
<td>6.0</td>
</tr>
<tr>
<td>13582</td>
<td>Fixed the issue where ibdiagnet’s unresponsiveness when using the get_physical_info flag caused UFM to hang.</td>
<td>ibdiagnet</td>
<td>5.10</td>
</tr>
<tr>
<td>12940</td>
<td>Fixed the issue where partition configuration was lost after upgrading to UFM version 5.9.6 and restarting the server.</td>
<td>partitions.conf, PKey, configuration</td>
<td>5.9.6</td>
</tr>
<tr>
<td>12765</td>
<td>Updated report execution command in order to avoid the following false warning of wrong link speed during topology comparison.</td>
<td>Topology compare report</td>
<td>5.9.6</td>
</tr>
</tbody>
</table>
1.8 Known Issues History

<table>
<thead>
<tr>
<th>Ref #</th>
<th>Issue</th>
</tr>
</thead>
</table>
| 3791820 | **Description:** Configuring the collection of SLVL on the secondary telemetry will result in SLVL data being sampled at a reduced rate.  
**Keywords:** SLVL, Multi-Rate, Reduced Rate  
**Workaround:** Edit the `launch_ibdiagnet_config.ini` file and restart the UFM telemetry.  
1. Edit the `launch_ibdiagnet_config.ini` file by running the following command:  
   ```  
   vi /opt/ufm/files/conf/secondary_telemetry_defaults/launch_ibdiagnet_config.ini  
   ```  
   Comment the following line:  
   ```  
   #base_freq=1  
   ```  
2. Restart UFM telemetry:  
   ```  
   /etc/init.d/ufmd ufm_telemetry_stop  
   /etc/init.d/ufmd ufm_telemetry_start  
   ```  
**Discovered in Release:** 6.15.0 |
| 3775405 | **Description:** Upon UFM startup, an empty temporary folder will be created at /tmp folder every 10 minutes (due to periodic telemetry status check)  
**Keywords:** Empty folder, temporary, /tmp |
<table>
<thead>
<tr>
<th>Ref #</th>
<th>Issue</th>
</tr>
</thead>
<tbody>
<tr>
<td>36</td>
<td><strong>Workaround:</strong> Add <code>rm -f /tmp/tmp*</code> to crontab to run daily or change instances_sessionscompatibility_interval parameter in gv.cfg to 30/60 minutes <strong>Discovery in Release:</strong> v6.15.0</td>
</tr>
<tr>
<td>3560659</td>
<td><strong>Description:</strong> Modifying the mtu_limit parameter for [MngNetwork] in gv.cfg does not accurately reflect changes upon restarting UFM. <strong>Keywords:</strong> mtu_limit, MngNetwork, gv.cfg, UFM restart <strong>Workaround:</strong> UFM needs to be restarted twice in order for the changes to take effect. <strong>Discovery in Release:</strong> v6.15.0</td>
</tr>
<tr>
<td>3729822</td>
<td><strong>Description:</strong> The Logs API temporarily returns an empty response when SM log file contains messages from both previous year (2023) and current year (2024). <strong>Keywords:</strong> Logs API, Empty response, Logs file <strong>Workaround:</strong> N/A (issue will be automatically resolved after the problematic SM log file, which includes messages from 2023 and 2024 years, will be rotated) <strong>Discovery in Release:</strong> v6.15.0</td>
</tr>
<tr>
<td>3675071</td>
<td><strong>Description:</strong> UFM stops gracefully after the b2b primary cable is physically disconnected <strong>Keywords:</strong> UFM HA, B2B, Primary Cable Disconnection <strong>Workaround:</strong> N/A <strong>Discovery in Release:</strong> 6.14.1</td>
</tr>
<tr>
<td>N/A</td>
<td><strong>Description:</strong> Execution of UFM Fabric Health Report (via UFM Web UI / REST API) will trigger ibdiagnet to use SLRG register which might cause some of the switch and HCA’s firmware to stick and cause the HCA’s ports to stay at &quot;Init&quot; state. <strong>Keywords:</strong> Fabric Health Report, SLRG register, &quot;Init&quot; state, Switch, HCA <strong>Discovery in Release:</strong> 6.14.0</td>
</tr>
<tr>
<td>3538640</td>
<td><strong>Description:</strong> Fixed ALM plugin log rotate function. <strong>Keywords:</strong> ALM, Plugin, Log rotate <strong>Discovery in Release:</strong> 6.13.0</td>
</tr>
<tr>
<td>3532191</td>
<td><strong>Description:</strong> Fixed UFM hanging (database is locked) after corrective restart of UFM health. <strong>Keywords:</strong> Hanging, Database, Locked <strong>Discovery in Release:</strong> 6.13.0</td>
</tr>
<tr>
<td>3555583</td>
<td><strong>Description:</strong> Resolved REST API links’ inability to return hostname for computer nodes. <strong>Keywords:</strong> REST API, Links, Hostname, Computer Nodes <strong>Discovery in Release:</strong> 6.12.1</td>
</tr>
<tr>
<td>3549795</td>
<td><strong>Description:</strong> Fixed ufm_ha_cluster status to show DRBD sync status. <strong>Keywords:</strong> ufm_ha_cluster, DRBD, Sync Status <strong>Discovery in Release:</strong> 6.13.0</td>
</tr>
<tr>
<td>3549793</td>
<td><strong>Description:</strong> Fixed UFM HA installation failure. <strong>Keywords:</strong> HA, Installation <strong>Discovery in Release:</strong> 6.13.0</td>
</tr>
<tr>
<td>Ref #</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>3547517</td>
<td>Fixed UFM logs REST API returning empty result when SM logs exist on the disk.</td>
</tr>
<tr>
<td>3546178</td>
<td>Fixed SHARP jobs failure when SHARP reservation feature is enabled.</td>
</tr>
<tr>
<td>3541477</td>
<td>Fixed UFM module temperature alerting on wrong thresholds.</td>
</tr>
<tr>
<td>3191419</td>
<td>Fixed UFM default session API returning port counter values as NULL.</td>
</tr>
<tr>
<td>3560659</td>
<td>Fixed proper update in [MngNetwork] mtu_limit in gv.cfg when restarting UFM.</td>
</tr>
<tr>
<td>3534374</td>
<td>Fixed configure_ha_nodes.sh failure when deploying UFM6.13.x HA on Ubuntu22.04</td>
</tr>
<tr>
<td>3496853</td>
<td>Fixed daily report not being sent properly.</td>
</tr>
<tr>
<td>3469639</td>
<td>Fixed REST RDMA server failure every couple of days, causing inability to retrieve ibdiagnet data.</td>
</tr>
<tr>
<td>3455767</td>
<td>Fixed incorrect combination of multiple devices in monitoring.</td>
</tr>
<tr>
<td>3511410</td>
<td>Collect system dump for DGX host does not work due to missing sshpass utility.</td>
</tr>
<tr>
<td>3432385</td>
<td>UFM does not support HDR switch configured with hybrid split mode, where some of the ports are split and some are not.</td>
</tr>
<tr>
<td>Ref #</td>
<td>Issue</td>
</tr>
<tr>
<td>--------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| 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:
  ```bash
  /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                                                                                                                                                                                                                                                                                                                                                                                   |
| 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 in 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                                                                                                                                                                                                                                                                                                                                                           |
| 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
  ```bash
  pcs cluster start --all
  ```
  3. Follow instructions in [Split-Brain Recovery in HA Installation](#).  
  **Keywords:** HA, Back-to-back Interface                                                                                                                                                                                                                                                                                                                                                                          |
<table>
<thead>
<tr>
<th>Ref #</th>
<th>Description</th>
<th>Workaround</th>
<th>Keywords</th>
</tr>
</thead>
<tbody>
<tr>
<td>3361160</td>
<td>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.</td>
<td>To preserve the historical telemetry database data while upgrading 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.</td>
<td>UFM Historical Telemetry Database, Cleanup, Upgrade</td>
</tr>
<tr>
<td>3346321</td>
<td>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</td>
<td>N/A</td>
<td>Multiport SM, Failover, Secondary port</td>
</tr>
<tr>
<td>3240664</td>
<td>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.</td>
<td>N/A</td>
<td>UFM Upgrade</td>
</tr>
<tr>
<td>3242332</td>
<td>Upgrading MLNX_OFED uninstalls UFM</td>
<td>Upgrade UFM to a newer version (v6.11.0 or newer), then upgrade MLNX_OFED</td>
<td>MLNX_OFED, Uninstall, UFM</td>
</tr>
<tr>
<td>3237353</td>
<td>Upgrading from UFM v6.10 removes MLNX_OFED crucial packages</td>
<td>Reinstall MLNX_OFED/UFM</td>
<td>MLNX_OFED, Upgrade, Packages</td>
</tr>
<tr>
<td>N/A</td>
<td>Running UFM software with external UFM-SM is no longer supported</td>
<td>N/A</td>
<td>External UFM-SM</td>
</tr>
<tr>
<td>3144732</td>
<td>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.</td>
<td>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.</td>
<td>Ubuntu 22, sysdump, sshpass</td>
</tr>
<tr>
<td>3129490</td>
<td>HA uninstall procedure might get stuck on Ubuntu 20.04 due to multipath daemon running on the host.</td>
<td>Stop the multipath daemon before running the HA uninstall script on Ubuntu 20.04.</td>
<td>HA uninstall, multipath daemon, Ubuntu 20.04</td>
</tr>
<tr>
<td>3147196</td>
<td>Running the upgrade procedure on bare metal Ubuntu 18.04 in HA mode might fail.</td>
<td>For instructions on how to apply the upgrade for bare metal Ubuntu 18.04, refer to High Availability Upgrade for Ubuntu 18.04.</td>
<td>Upgrade, Ubuntu 18.04, Docker Container, failure</td>
</tr>
<tr>
<td>3145058</td>
<td>Running upgrade procedure on UFM Docker Container in HA mode might fail.</td>
<td>For instructions on how to apply the upgrade for UFM Docker Container in HA, refer to Upgrade Container Procedure.</td>
<td>Upgrading UFM Docker Container in HA mode might fail.</td>
</tr>
<tr>
<td>Ref #</td>
<td>Issue</td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>----------------------------------------------------------------------</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| 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.cfg `log_dir`)  
  **Keywords:** Log, Dir, fail, start |
| 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 |
<p>| 2784560 | <strong>Description:</strong> Intentional stop for master container and start it again or reboot of master server will damage the HA failover option |</p>
<table>
<thead>
<tr>
<th>Ref #</th>
<th>Issue</th>
</tr>
</thead>
</table>
| 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                                                                                                          |
| 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 last 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 a 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 REST API (GET /ufmRest/monitoring/session/0/data) will not include port statistics for 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 of master server will damage the HA failover option  
**Workaround:** manually Restart UFM cluster  
**Keywords:** UFM Container; Reboot, Failover                                                                                                  |
<table>
<thead>
<tr>
<th>Ref #</th>
<th>Description</th>
<th>Workaround</th>
<th>Keywords</th>
<th>Discovered in release</th>
</tr>
</thead>
<tbody>
<tr>
<td>2872513</td>
<td>Description: after rebooting master container, Failover will be triggered twice (once to the standby and then back again to the master container)</td>
<td>N/A</td>
<td>UFM Container, reboot, failover</td>
<td>6.7</td>
</tr>
<tr>
<td>2863388</td>
<td>Description: Fail to get cables info for NDR Splitted Port.</td>
<td>N/A</td>
<td>Cable, NDR, Split</td>
<td></td>
</tr>
<tr>
<td>N/A</td>
<td>Description: In case of using SM mkey per port, several UFM operations might fail (get cable info, get system dump, switch FW upgrade)</td>
<td>N/A</td>
<td>SM, mkey per port,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>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.</td>
<td>N/A</td>
<td>telemetry, history, bond, failure</td>
<td>6.7</td>
</tr>
<tr>
<td></td>
<td>Description: SHARP Aggregation Manager over UCX is not supported.</td>
<td>N/A</td>
<td>UCX, SHARP AM</td>
<td>6.6.1</td>
</tr>
<tr>
<td>2459320</td>
<td>Description: Docker upgrade to UFM6.6.1 from UFM6.6.0 is not supported.</td>
<td>N/A</td>
<td>Docker; upgrade</td>
<td>6.6.1</td>
</tr>
<tr>
<td></td>
<td>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.”</td>
<td>N/A</td>
<td>System dump, UFM Appliance host</td>
<td>6.5.2</td>
</tr>
<tr>
<td>2100564</td>
<td>Description: For modular dual-management switch systems, switch information is not presented correctly if the primary management module fails and the secondary takes over.</td>
<td>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.</td>
<td>Modular switch, dual-management, virtual IP, box IP</td>
<td>6.4.1</td>
</tr>
<tr>
<td>Ref #</td>
<td>Issue</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>----------</td>
<td>----------------------------------------------------------------------</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| 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 |
| -        | **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 |
2 UFM Overview

2.1 Scale-Out Your Fabric with Unified Fabric Manager

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.

2.2 UFM Benefits

<table>
<thead>
<tr>
<th>Benefit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Central Console for Fabric Management</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>In-Depth Fabric Visibility and Control</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>Advanced Traffic Analysis</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>Enables Multiple Isolated Application Environments on a Shared Fabric</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>Service-Oriented Automatic Resource Provisioning</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>Quick Resolution of Fabric Problems</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>Seamless Failover Handling</strong></td>
<td>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.</td>
</tr>
</tbody>
</table>
2.3 Main Functionality Modules

<table>
<thead>
<tr>
<th>Module</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fabric Dashboard</td>
<td>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.</td>
</tr>
<tr>
<td>Fabric Segmentation (PKey Management)</td>
<td>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).</td>
</tr>
<tr>
<td>Fabric Discovery and Physical View</td>
<td>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.</td>
</tr>
<tr>
<td>Central Device Management</td>
<td>UFM provides the ability to centrally access switches and hosts, and perform maintenance tasks such as firmware and software upgrade, shutdown and restart.</td>
</tr>
<tr>
<td>Monitoring</td>
<td>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.</td>
</tr>
<tr>
<td>Configuration</td>
<td>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.</td>
</tr>
<tr>
<td>Fabric Health</td>
<td>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.</td>
</tr>
<tr>
<td>Logging</td>
<td>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.</td>
</tr>
<tr>
<td>Module</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>High Availability</td>
<td>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 NVIDIA’s High Availability switching solutions allows for non-disruptive operation of complex and demanding data center environments.</td>
</tr>
</tbody>
</table>

Please refer to the following sections for UFM’s main functionalities:
- Events and Alarms
- Reports
- Telemetry

### 2.4 UFM Communication Requirements

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

#### 2.4.1 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.

*Note:* UFM Server Communication with Clients

#### 2.4.1.1 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.
2.4.1.2 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.

2.4.1.3 Summary of UFM Server Communication with Clients

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

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

2.4.2 UFM Server Communication with InfiniBand Switches

2.4.2.1 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.

2.4.2.2 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).

### 2.4.2.3 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.

### 2.4.2.4 Summary of UFM Server Communication with InfiniBand Switches

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

### 2.4.3 UFM Server Communication with InfiniBand Hosts

*UFM Server Communication with InfiniBand Hosts*
2.4.3.1 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.

2.4.3.2 UFM Server Communication with InfiniBand Hosts

<table>
<thead>
<tr>
<th>Affected Service</th>
<th>Network</th>
<th>Address / Service / Port</th>
<th>Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>InfiniBand Management / Monitoring</strong></td>
<td><strong>InfiniBand</strong></td>
<td><strong>Management Datagrams</strong></td>
<td>Bi-directional</td>
</tr>
</tbody>
</table>

2.4.4 UFM Server High Availability (HA) Active—Standby Communication

*UFM Server HA Active—Standby Communication*
2.4.4.1 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).

<table>
<thead>
<tr>
<th>Affected Service</th>
<th>Network</th>
<th>Address / Service / Port</th>
<th>Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>UFM HA heartbeat</td>
<td>Out-of-band management*</td>
<td>UDP / 694</td>
<td>Bi-directional</td>
</tr>
<tr>
<td>UFM HA DRBD</td>
<td>Out-of-band management*</td>
<td>TCP / 8888</td>
<td>Bi-directional</td>
</tr>
</tbody>
</table>

*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.

2.5 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*
2.5.1 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.

2.5.2 Client Tier API

Third-party clients are managed by the REST API.

2.5.3 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.

2.5.4 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.

2.5.5 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.

2.5.6 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.

2.5.7 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.
2.5.8 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.

2.5.9 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.

2.5.10 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.

2.6 Getting Familiar with UFM's Data Model

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.

2.6.1 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.

2.6.2 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

<table>
<thead>
<tr>
<th>Icon</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>N/A</td>
<td>Port Object</td>
<td>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.</td>
</tr>
<tr>
<td>Icon</td>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>---------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>N/A</td>
<td>Module Object</td>
<td>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.</td>
</tr>
<tr>
<td><img src="icon.png" alt="Icon" /></td>
<td>Link Object</td>
<td>Represents the physical connection between two active ports.</td>
</tr>
<tr>
<td>N/A</td>
<td>Cable Object</td>
<td>Represents the physical cable or the transceiver connected to one of the link edges.</td>
</tr>
<tr>
<td><img src="icon.png" alt="Icon" /></td>
<td>Computer Object</td>
<td>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.</td>
</tr>
<tr>
<td><img src="icon.png" alt="Icon" /></td>
<td>Switch Object</td>
<td>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.</td>
</tr>
<tr>
<td><img src="icon.png" alt="Icon" /></td>
<td>Rack Object</td>
<td>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.</td>
</tr>
</tbody>
</table>

## 2.7 UFM Web UI Overview

The UFM Web User Interface (GUI) lets you access UFM through a web browser, where you can visualize your network and interact with the display using a keyboard and mouse.

The UFM Web UI is supported on Google Chrome and TBD. It is designed to be viewed on a display with a minimum resolution of 1920 × 1080 pixels.

- [Access the WebUI](#)
- [WebUI Layout](#)
- [Set User Preferences](#)

### 2.7.1 Access the WebUI

#### 2.7.1.1 UFM Web UI Supported Browsers

UFM Web UI is supported on all the following web browsers: Internet Explorer, Firefox, Chrome and Opera.

For optimal UFM Web UI performance, make sure you are using the latest version available of Google Chrome.

For more information, see *UFM User Manual*.

#### 2.7.1.2 Launching UFM Web UI Session

Before accessing the UFM Web UI:
• If required, you can change the configuration of the connection (port and protocol) between the UFM server and the APACHE server in the file `gv.cfg`:
  • `ws_protocol = http` or `https`
  • Setting the parameter `ws_protocol` to `http` allows unsecured access
  • Setting the parameter `ws_protocol` to `https` denies unsecured access.
  • `ws_port = port number`

➢ To launch a UFM Web UI session, do the following:
  1. Launch the Web UI by entering the following URL in your browser:
     http://<UFM_server_IP>/ufm
     https://<UFM_server_IP>/ufm

  2. In the Login page, enter your User Name and your predefined user Password and click Login.

Once you have entered your user name and password, the main window shows the UFM Dashboard. For more information, see the [Fabric Dashboard](#).

### 2.7.2 WebUI Layout

The UFM WebUI contains two main areas:

1. Top Bar - Contains local time zone and user information on the top right side of the screen.
2. Sidebar Menu - Contains a taskbar accessible from a sidebar menu on the left side of the screen. For more information on each tab, refer to [UFM Web UI](#).
2.7.2.1 Top Bar

Each user can customize the UFM display, time zone, and date format, change their account password, and manage their preferences. For details, refer to Set User Preferences.

2.7.2.2 Sidebar Menu

<table>
<thead>
<tr>
<th>Tab Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Dashboard" /></td>
<td>Provides a summary view of the fabric status.</td>
</tr>
<tr>
<td><img src="image" alt="Network Map" /></td>
<td>Provides a hierarchical topology view of the fabric.</td>
</tr>
<tr>
<td><img src="image" alt="Managed Elements" /></td>
<td>Provides information on all fabric devices. This information is presented in a table format.</td>
</tr>
<tr>
<td>Tab Icon</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td><img src="image" alt="Logical Elements" /></td>
<td>Provides information on all logical servers. This information is presented in a table format.</td>
</tr>
<tr>
<td><img src="image" alt="Events &amp; Alarms" /></td>
<td>Provides information on the events &amp; alarms generated by the system.</td>
</tr>
<tr>
<td><img src="image" alt="Telemetry" /></td>
<td>Enables establishing monitoring sessions on devices or ports.</td>
</tr>
<tr>
<td><img src="image" alt="System Health" /></td>
<td>Enables running and viewing fabric reports, UFM reports, and system logs. You can also back up UFM configuration files.</td>
</tr>
<tr>
<td><img src="image" alt="Jobs" /></td>
<td>Provides information on all jobs created, as a result of UFM actions.</td>
</tr>
<tr>
<td><img src="image" alt="Settings" /></td>
<td>Enables configuring UFM server and UFM fabric settings, including events policy, device access, network management, subnet manager, and user management</td>
</tr>
</tbody>
</table>

### 2.7.3 Set User Preferences

This section describes how to customize your UFM display settings and change your password.

#### 2.7.3.1 Dark/Light Theme

1. Select Preferences.

   ![Preferences](image)

   - **Enable Dark Mode**

2. In User Preferences, enable dark mode for UFM presentation in a dark theme. The following figure shows the dark theme:
2.7.3.2 Time Zone Converter

Allows you to unify all times in UFM like events and alarms, ibdagnet, telemetry and logs. You can switch between local and machine time.

In the status bar drop-down menu, switch between local and server/machine time.
2.7.3.3 Table Enhancements

2.7.3.3.1 Look and Feel Improvements

In the screenshots, the difference between Server Time and Local Time is 6 hours.

2.7.3.3.2 Displayed Columns
2.7.3.4 Export All Data as CSV

There are two options for exporting as CSV:

- All Data: all data returned from server.
- Displayed Data: only displayed rows.

⚠️ Displayed columns of all tables are persistent per user, with the option to restore defaults.
3 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.

3.1 Prerequisites for UFM Server Software Installation

Please refer to Installation Notes for information on system prerequisites.

3.2 UFM Installation Steps

To install the UFM software:

1. Download the UFM software and license file
2. Install the UFM server software and activate the license file
3. Perform initial configuration
4. Run the UFM server software

3.3 UFM Installation Steps

3.3.1 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.

3.3.1.1 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.

3.3.1.2 Downloading UFM Software

⚠️ 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.

![Software Downloads Image](image)

2. Save the file on your local drive.
3. Click Close.

### 3.3.2 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

⚠️ **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
3.3.2.1 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.

<table>
<thead>
<tr>
<th>RedHat 7</th>
<th>RedHat 8</th>
<th>RedHat 9</th>
<th>Ubuntu 18.04</th>
<th>Ubuntu 20.04</th>
<th>Ubuntu 22.04</th>
</tr>
</thead>
<tbody>
<tr>
<td>acl</td>
<td>acl</td>
<td>acl</td>
<td>acl</td>
<td>acl</td>
<td>acl</td>
</tr>
<tr>
<td>apr-util-openssl</td>
<td>apr-util-openssl</td>
<td>apr-util-openssl</td>
<td>apache2</td>
<td>apache2</td>
<td>apache2</td>
</tr>
<tr>
<td>bc</td>
<td>bc</td>
<td>bc</td>
<td>bc</td>
<td>bc</td>
<td>bc</td>
</tr>
<tr>
<td>cairo</td>
<td>gnuTLS</td>
<td>gnuTLS</td>
<td>chrpath</td>
<td>chrpath</td>
<td>chrpath</td>
</tr>
<tr>
<td>gnuTLS</td>
<td>httpd</td>
<td>httpd</td>
<td>cron</td>
<td>cron</td>
<td>cron</td>
</tr>
<tr>
<td>httpd</td>
<td>iptables</td>
<td>iptables</td>
<td>gawk</td>
<td>gawk</td>
<td>gawk</td>
</tr>
<tr>
<td>iptables</td>
<td>jansson</td>
<td>jansson</td>
<td>lftp</td>
<td>lftp</td>
<td>lftp</td>
</tr>
<tr>
<td>lftp</td>
<td>lftp</td>
<td>lftp</td>
<td>libcurl4</td>
<td>libcurl4</td>
<td>libcurl4</td>
</tr>
<tr>
<td>libxml2</td>
<td>libnsl</td>
<td>libnsl</td>
<td>logrotate</td>
<td>logrotate</td>
<td>logrotate</td>
</tr>
<tr>
<td>libxml2</td>
<td>libxml2</td>
<td>libxml2</td>
<td>python3</td>
<td>python3</td>
<td>python3</td>
</tr>
<tr>
<td>mod_session</td>
<td>libxml2</td>
<td>libxml2</td>
<td>qperf</td>
<td>qperf</td>
<td>qperf</td>
</tr>
<tr>
<td>mod_ssl</td>
<td>mod_session</td>
<td>mod_session</td>
<td>rsync</td>
<td>rsync</td>
<td>rsync</td>
</tr>
<tr>
<td>net-snmp</td>
<td>mod_ssl</td>
<td>mod_ssl</td>
<td>snmpd</td>
<td>snmpd</td>
<td>snmpd</td>
</tr>
<tr>
<td>net-snmp</td>
<td>net-snmp</td>
<td>net-snmp</td>
<td>sqlite3</td>
<td>sqlite3</td>
<td>sqlite3</td>
</tr>
<tr>
<td>net-snmp-utils</td>
<td>net-snmp-utils</td>
<td>net-snmp-utils</td>
<td>sshpass</td>
<td>sshpass</td>
<td>sshpass</td>
</tr>
<tr>
<td>net-tools</td>
<td>net-snmp-utils</td>
<td>net-snmp-utils</td>
<td>ssl-cert</td>
<td>ssl-cert</td>
<td>ssl-cert</td>
</tr>
<tr>
<td>php</td>
<td>net-tools</td>
<td>net-tools</td>
<td>sudo</td>
<td>sudo</td>
<td>sudo</td>
</tr>
<tr>
<td>psmisc</td>
<td>php</td>
<td>php</td>
<td>telnet</td>
<td>telnet</td>
<td>telnet</td>
</tr>
<tr>
<td>python3</td>
<td>psmisc</td>
<td>psmisc</td>
<td>zip</td>
<td>zip</td>
<td>zip</td>
</tr>
<tr>
<td>python3-libs</td>
<td>python36</td>
<td>python3</td>
<td>zip</td>
<td>zip</td>
<td>zip</td>
</tr>
<tr>
<td>qperf</td>
<td>qperf</td>
<td>qperf</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>rsync</td>
<td>rsync</td>
<td>rsync</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sqlite</td>
<td>sqlite</td>
<td>sqlite</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sshpass</td>
<td>sshpass</td>
<td>sshpass</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sudo</td>
<td>sudo</td>
<td>sudo</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>telnet</td>
<td>telnet</td>
<td>telnet</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>zip</td>
<td>zip</td>
<td>zip</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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).
- SELinux is disabled.
- 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).

![Warning]

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:
  ```bash
  rpm -e opensm-3.3.9.MLNX_20111006_e52d5fc-0.1
  ```
  Ubuntu:
  ```bash
  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:

```bash
ufma_interfaces = ib0,eth0
```

### 3.3.2.2 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.

![Warning]

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.

3.3.2.3 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

3.3.2.3.1 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).
1. On both servers, Install UFM Enterprise in Stand Alone (SA) mode.

   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 drbd-utils kmod-drbd84
   OR
   yum install pcs pacemaker drbd-utils kmod-drbd90
   ```

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

   ```
   wget https://www.mellanox.com/downloads/UFM/ufm_ha_5.5.0-9.tgz
   ```

   For Sha256:

   ```
   wget https://download.nvidia.com/ufm/ufm_ha/ufm_ha_5.5.0-9.sha256
   ```

   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)**
a. 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 192.168.10.3 \
--standby-secondary-ip 192.168.10.2 \
--no-vip
```

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

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

⚠️ 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.

⚠️ 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.

⚠️ 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.

b. 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:

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

a. [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
```

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

⚠️ 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:

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

a. [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
```

b. [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
```

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
```
3.3.2.3.2 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:
   
   ```bash
   ./install.sh
   ```

   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).

   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.

3.3.2.4 Installing UFM Docker Container Mode

3.3.2.4.1 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.

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):
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

3.3.2.4.2 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.

3.3.2.4.3 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
```

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>
```

3.3.2.4.4 Step 2: Installing UFM Docker

3.3.2.4.4.1 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.

   **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

<table>
<thead>
<tr>
<th>Flag</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-f</code></td>
<td><code>--fabric-interface</code></td>
<td>IB fabric interface name.</td>
</tr>
<tr>
<td><code>-g</code></td>
<td><code>--mgmt-interface</code></td>
<td>Management interface name.</td>
</tr>
<tr>
<td><code>-h</code></td>
<td><code>--help</code></td>
<td>Show help</td>
</tr>
<tr>
<td><code>-m</code></td>
<td><code>--multisubnet-consumer</code></td>
<td>UFM Multisubnet Consumer mode</td>
</tr>
</tbody>
</table>

### 3.3.2.4.5 Installation Modes

UFM Enterprise installer supports several deployment modes:

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

### 3.3.2.4.6 Installing UFM on Docker Container - High Availability Mode

#### 3.3.2.4.6.1 Pre-deployments requirements

- Install pacemaker, pcs, and drbd-utils on both servers
  - For Ubuntu:
    ```bash
    apt install pcs pacemaker drbd-utils
    ```
  - For CentOS/Red Hat:
    ```bash
    yum install pcs pacemaker drbd-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.

### 3.3.2.4.6.2 Installing UFM Containers

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

```bash
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
```

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

```bash
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
```

### 3.3.2.4.6.3 Downloading UFM HA Package

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

```bash
wget https://www.mellanox.com/downloads/UFM/ufm_ha_5.4.0-9.tgz
```

For Sha256:

```bash
wget https://download.nvidia.com/ufm/ufm_ha/ufm_ha_5.4.0-9.sha256
```

### 3.3.2.4.6.4 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`

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

### 3.3.2.4.6.5 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:

   ```bash
   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
   ```

   - The script `configure_ha_nodes.sh` is located under `/usr/local/bin/`, therefore, by default, you do not need to use the full path to run it.
   - The `--cluster-password` must be at least 8 characters long.
   - 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.

   - `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:

   ```bash
   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:

```bash
ufm_ha_cluster config -h
```

To configure HA, follow the below instructions:
Please change the variables in the commands below based on your setup.

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

   ```bash
   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:

   ```bash
   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)

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:

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

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

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

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

   ```bash
   ufm_ha_cluster config -r master \
   -e 10.212.145.6 \
   -l 10.212.145.5 \
   -i 10.212.145.00 \
   --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:

```bash
ufm_ha_cluster status
```

IPv6 Example:

```bash
```

Starting HA Cluster

- To start UFM HA cluster:

  ```bash
  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

3.3.2.4.7 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:

```bash
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:

```bash
systemctl daemon-reload
```

4. To Start UFM Enterprise service run:

```bash
systemctl start ufm-enterprise
```

3.3.2.5 Replacing the Standby Node in HA Mode

3.3.2.5.1 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 -1 <local primary ip address> -e <peer primary ip address> -E <peer secondary ip address> -p <clust
3.4 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.

![Warning](image)

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.

3.4.1 Licensing

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

![About](image)

Nvidia Mellanox'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.

Version: 6.5.2-5

Advanced

Copyright © 2013 - 2020 Nvidia Mellanox Technologies
All rights reserved

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 "ufmllicense" 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 output example when running the CLI command `ufmllicense` in SA and HA Modes.

HA Mode Output Example:
3.5 UFM Configuration

3.5.1 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)

3.5.1.1 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.

To remove a license:

Delete the license file from /opt/ufm/files/licenses.
The fabric interface should be set to one of the InfiniBand IPoIB interfaces, which connect the UFM/SM to the fabric:

\[
fabric\_interface = ib0
\]

- By default, \texttt{fabric\_interface} is set to \texttt{ib0}
- \texttt{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).

3.5.2 Additional Configuration (Optional)

- 3.5.2.1 General Settings in \texttt{gv.cfg}
  - 3.5.2.1.1 Enabling SHARP Aggregation Manager
  - 3.5.2.1.2 Enabling Predefined Groups
  - 3.5.2.1.3 Enabling Multi-NIC Host Grouping
  - 3.5.2.1.4 Defining Node Description Black-List
  - 3.5.2.1.5 Running UFM Over IPv6 Network Protocol
  - 3.5.2.1.6 Adding SM Plugin (e.g. lossymgr) to \texttt{event\_plugin\_name} Option
  - 3.5.2.1.7 Multi-port SM
  - 3.5.2.1.8 Configuring UDP Buffer
  - 3.5.2.1.9 Virtualization
  - 3.5.2.1.10 Static SM LID
  - 3.5.2.1.11 Configuring Log Rotation
  - 3.5.2.1.12 Configuring UFM Logging
  - 3.5.2.1.13 Configuring UFM Over Static IPv4 Address
  - 3.5.2.1.14 Configuring Syslog
  - 3.5.2.1.15 Configuration Examples in \texttt{gv.cfg}
  - 3.5.2.1.16 Dynamic Telemetry
  - 3.5.2.1.17 SM Trap Handler Configuration
  - 3.5.2.1.18 CPU Affinity on UFM
  - 3.5.2.1.19 Quality of Service (QoS) Support
  - 3.5.2.1.20 UFM Failover to Another Port
  - 3.5.2.1.21 Configure Managed Switches Info Persistency
  - 3.5.2.1.22 Configuring Partial Switch ASIC Failure Events
  - 3.5.2.1.23 Enabling Network Fast Recovery
  - 3.5.2.1.24 Disabling Rest Roles Access Control
  - 3.5.2.1.25 Enabling/Disabling Authentication
    - 3.5.2.1.25.1 Kerberos Authentication
    - 3.5.2.1.25.2 UFM Authentication Server
    - 3.5.2.1.25.3 Azure AD Authentication

- 3.5.2.2 Setting up Telemetry in UFM
  - 3.5.2.2.1 Enabling UFM Telemetry
  - 3.5.2.2.2 Changing UFM Telemetry Default Configuration
  - 3.5.2.2.3 Supporting Generic Counters Parsing and Display
3.5.2.2.4 Supporting Multiple Telemetry Instances Fetch
3.5.2.2.5 Low-Frequency (Secondary) Telemetry
   3.5.2.2.5.1 Low-Frequency (Secondary) Telemetry Exposing IPv6 Counters
   3.5.2.2.5.2 Stopping Telemetry Endpoint Using CLI Command

3.5.2.1 General Settings in gv.cfg

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

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

### 3.5.2.1.1 Enabling SHARP Aggregation Manager

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

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

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

### 3.5.2.1.2 Enabling Predefined Groups

```
enable_predefined_groups = true
```

⚠️ 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.

### 3.5.2.1.3 Enabling Multi-NIC Host Grouping

```
multinic_host_enabled = true
```

⚠️ 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.

⚠️ It is recommended to set the value of this parameter before running UFM for the first time.
3.5.2.1.4 Defining Node Description Black-List

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
```

3.5.2.1.5 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
```

3.5.2.1.6 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.

3.5.2.1.7 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 =
```

⚠️ 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.

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

3.5.2.1.8 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 default, 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
```

3.5.2.1.9 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
```

3.5.2.1.10 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
```

⚠️ 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.
3.5.2.1.11 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
```

3.5.2.1.12 Configuring UFM Logging

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

<table>
<thead>
<tr>
<th>Field</th>
<th>Default Value</th>
<th>Value Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>level</td>
<td>WARNING</td>
<td>CRITICAL, ERROR, WARNING, INFO, DEBUG</td>
<td>The definition of the maub logging level for UFM components.</td>
</tr>
<tr>
<td>smclient_level</td>
<td>WARNING</td>
<td>CRITICAL, ERROR, WARNING, INFO, DEBUG</td>
<td>The logging level for SM client log messages</td>
</tr>
<tr>
<td>event_log_level</td>
<td>INFO</td>
<td>CRITICAL, ERROR, WARNING, INFO, DEBUG</td>
<td>The logging level for UFM events log messages</td>
</tr>
<tr>
<td>rest_log_level</td>
<td>INFO</td>
<td>CRITICAL, ERROR, WARNING, INFO, DEBUG</td>
<td>Logging level for REST API related log messages</td>
</tr>
<tr>
<td>authentication_service_log_level</td>
<td>INFO</td>
<td>CRITICAL, ERROR, WARNING, INFO, DEBUG</td>
<td>logging level for UFM authentication log messages</td>
</tr>
<tr>
<td>log_dir</td>
<td>/opt/ufm/files/log</td>
<td>N/A</td>
<td>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 ufmap user (ufmap 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.</td>
</tr>
<tr>
<td>max_history_lines</td>
<td>100000</td>
<td>N/A</td>
<td>The maximum number of lines in log files to be shown in UI output for UFM logging.</td>
</tr>
</tbody>
</table>

```
[Logging]
# Optional logging levels: CRITICAL, ERROR, WARNING, INFO, DEBUG.
level = WARNING
smclient_level = WARNING
event_log_level = INFO
```

85
3.5.2.1.13 Configuring UFM Over Static IPv4 Address

Follow this procedure 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
   ```

3.5.2.1.14 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.
<table>
<thead>
<tr>
<th>Field</th>
<th>Default Value</th>
<th>Value Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>syslog</td>
<td>false</td>
<td>True or False</td>
<td>Enables/disables UFM syslog option</td>
</tr>
<tr>
<td>syslog_addr</td>
<td>/dev/log # for remote rsyslog_hostname:514</td>
<td>N/A</td>
<td>UFM syslog configuration (syslog_addr) For working with local syslog, set value to: /dev/log For working with external machine, set value to: host:port Important note: the default remote syslog server port is 514 As the UFM log messages could be sent to remote server, change the rsyslog configuration on the remote server The /etc/rsyslog.conf file should be edited and two sections should be uncommented as shown below: # Provides UDP syslog reception $ModLoad imudp $UDPServerRun 514 # Provides TCP syslog reception $ModLoad imtcp $InputTCPServerRun 514 Restart the remote syslog service, run: service rsyslog restart</td>
</tr>
<tr>
<td>ufm_syslog</td>
<td>false</td>
<td>True or False</td>
<td>Sets syslog option for main UFM process logging messages - False - Not to send. True: Send</td>
</tr>
<tr>
<td>smclient_syslog</td>
<td>false</td>
<td>True or False</td>
<td>Sets syslog option for OpenSM logging messages - False - Not to send. True: Send</td>
</tr>
<tr>
<td>event_syslog</td>
<td>false</td>
<td>True or False</td>
<td>Sets syslog option for events logging messages - False - Not to send. True: Send</td>
</tr>
<tr>
<td>rest_syslog</td>
<td>false</td>
<td>True or False</td>
<td>Sets syslog option for UFM REST API logging messages - False - Not to send. True: Send</td>
</tr>
<tr>
<td>authentication_syslog</td>
<td>false</td>
<td>True or False</td>
<td>Set syslog option for UFM authentication logging messages. False - Not to send. True: Send</td>
</tr>
<tr>
<td>Field</td>
<td>Default Value</td>
<td>Value Options</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>---------------</td>
<td>--------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>syslog_level</td>
<td>WARNING</td>
<td>CRITICAL, ERROR, WARNING, INFO, DEBUG</td>
<td>Sets global syslog messages logging level. The syslog level is common for all the UFM components. The syslog level that is sent to syslog is the highest among the syslog level and component log level defined in the above section.</td>
</tr>
<tr>
<td>syslog_facility</td>
<td>LOG_USER</td>
<td>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</td>
<td>Includes the remote syslog package header value for log message facility.</td>
</tr>
</tbody>
</table>

```bash
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
```

3.5.2.1.15 Configuration Examples in gv.cfg

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

- Polling interval for Fabric Dashboard information

  ```text
  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.

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

- HTTP/HTTPS Port Configuration

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

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

3.5.2.1.16 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.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>max_instances</td>
<td>Maximum number of simultaneous running UFM Telemetries.</td>
<td>5</td>
</tr>
<tr>
<td>new_instance_delay</td>
<td>Delay time between the start of two UFM Telemetry instances, in minutes.</td>
<td>5</td>
</tr>
<tr>
<td>update_discovery_delay</td>
<td>The time to wait before updating the discovery file of each telemetry instance if the fabric has changed, in minutes.</td>
<td>10</td>
</tr>
<tr>
<td>endpoint_timeout</td>
<td>Telemetry endpoint timeout, in seconds.</td>
<td>5</td>
</tr>
<tr>
<td>bringup_timeout</td>
<td>Telemetry bringup tool timeout, in seconds.</td>
<td>60</td>
</tr>
<tr>
<td>initial_exposed_port</td>
<td>Initial port for the available range of ports (range(initial_exposed_port, initial_exposed_port + max_instances)).</td>
<td>9003</td>
</tr>
<tr>
<td>instances_sessions_compatibility_interval</td>
<td>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</td>
<td>10</td>
</tr>
</tbody>
</table>
3.5.2.1.17  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

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

3.5.2.1.18  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.

3.5.2.1.19  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
```

⚠️ 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.

### 3.5.2.1.20 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.

3.5.2.1.21 Configure 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 file:

```
[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
# 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 be 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
```

3.5.2.1.22 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 to 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 | 2021-01-02 10:54:32 | Unhealthy Port | default/1/ Switch: 1-20/1-20 | Port Peer Port is considered as unhealthy due to MANUAL

Critical | 2021-01-02 10:54:32 | Port ASIC Fails | default/1/ Switch: 1-20/1-20 | Port Peer Port is considered as unhealthy due to MANUAL

```

93
3.5.2.1.23 Enabling Network Fast Recovery

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 and corresponding action protocols. These rules permit the switch to promptly react to substrand link 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 & 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 in the `gv.cfg` file. By doing this, you can specify which 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]
is_fast_recovery_enabled = false
# This will be supported by the Network Fast Recovery.
network_fast_recovery_conditions =
  SWITCH_DECISION_CREDIT_WATCHDOG,SWITCH_DECISION_RAW_BER,SWITCH_DECISION_EFFECTIVE_BER,SWITCH_DECISION_SYMBOL_BER
```

The "Unhealthy Ports" page provides visibility of these ports. If desired, the user can mark a port as healthy, triggering a restart of that specific port on the switch.

The trigger that initiated the isolation of ports can be viewed under the "Condition" column, as seen below.
3.5.2.1.24 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
```

3.5.2.1.25 Enabling/Disabling Authentication

3.5.2.1.25.1 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 principal 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_sign_up`: 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.

3.5.2.1.25.2 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
```

3.5.2.1.25.3 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 =
```

3.5.2.2 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.

3.5.2.2.1 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
```
3.5.2.2.2 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.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>hca</td>
<td>-</td>
</tr>
<tr>
<td>scope_file</td>
<td>-</td>
</tr>
<tr>
<td>plugin.env.PROMETHEUS_ENDPOINT</td>
<td>The port on which HTTP endpoint is configured</td>
</tr>
<tr>
<td>plugin.env.PROMETHEUS_INDEXES</td>
<td>Configures how data is indexed and stored in memory</td>
</tr>
<tr>
<td>config_watch_enabled=true</td>
<td>Configures network watcher to inform ibdiagnet that network topology has</td>
</tr>
<tr>
<td></td>
<td>changed (as ibdiagnet lacks the ability to re-discover network changes)</td>
</tr>
<tr>
<td>plugin.env.PROMETHEUS_CSET_DIR</td>
<td>Specifies where the counterset files, which define the data to</td>
</tr>
<tr>
<td></td>
<td>be retrieved and the corresponding counter names.</td>
</tr>
<tr>
<td>num_iterations</td>
<td>The number of iterations to run before ‘restarting’, i.e. rediscovering</td>
</tr>
<tr>
<td></td>
<td>fabric.</td>
</tr>
<tr>
<td>plugin.env.CLX_RESTART_FILE</td>
<td>A file that is ‘touched’ to indicate that an ibdiagnet restart is necessary</td>
</tr>
</tbody>
</table>

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

3.5.2.2.3 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).

3.5.2.2.4 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_url” 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://<IP>:<PORT>/csv/<CSET_NAME>. For example http://10.10.10.10:9001/csv/minimal http://10.10.10.10:9002/csv/test.

⚠️ 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.

3.5.2.2.5 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_per_slvl.xcset`)

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_freq_debug_per_slvl.xcset`.

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

3.5.2.2.5.1 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.

3.5.2.2.5.2 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_ibdiagnet_config.ini
  ```
  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` -
  ```bash
v1 /opt/ufm/files/conf/secondary_telemetry/prometheus_configs/cset/<name for your choice>.xcset
  ```


• Existing `xcset` -
  ```bash
v1 /opt/ufm/files/conf/secondary_telemetry/prometheus_configs/cset/low_freq_debug.xcset
  ```

• 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:
  ```bash
/etc/init.d/ufmd ufm_telemetry_stop
/etc/init.d/ufmd ufm_telemetry_start
  ```

### 3.6 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
3.6.1 Running UFM Server Software in Management Mode

After installing, run the UFM Server by invoking:

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

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

3.6.2 Running UFM Software in High Availability Mode

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

```bash
ufm_ha_cluster start
```

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

**ufm_ha_cluster Command Options**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>start</td>
<td>Starts UFM HA cluster.</td>
</tr>
<tr>
<td>stop</td>
<td>Stops UFM HA cluster.</td>
</tr>
<tr>
<td>failover</td>
<td>Initiates failover (change mastership from local server to remote server).</td>
</tr>
<tr>
<td>takeover</td>
<td>Initiates takeover (change mastership from remote server to local server).</td>
</tr>
<tr>
<td>status</td>
<td>Shows current HA cluster status.</td>
</tr>
<tr>
<td>cleanup</td>
<td>Cleans the HA configurations on this node.</td>
</tr>
<tr>
<td>help</td>
<td>Displays help text.</td>
</tr>
</tbody>
</table>

3.6.3 HTTP/HTTPS Configuration

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

After installation, the user can change 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.

3.6.4 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.

3.6.5 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.

3.6.6 UFM Authentication Server

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

3.6.6.1 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`.

### 3.6.7 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:

#### 3.6.7.1 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.

![Azure Portal App Registrations](image)

4. Provide the application details:
   a. **Name**: Enter a descriptive name.
   b. **Supported account types**: Account types that are allowed to login and use the registered application.
c. Redirect URL: select the app type Web, and Add the following redirect URL https://<ufm_server>/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:
9. Click on "Overview" in the side menu to view the application information, such as tenant ID, client ID, and other details.

3.6.7.2 Enable Azure Authentication From UFM

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

3.6.7.3 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:
3.6.7.4 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.

3.6.7.4.1 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:

```bash
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.

```bash
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.

3.6.7.4.2 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:

   ```bash
   # 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:

   ```conf
   [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):

   ```bash
   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.
3.6.8 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

⚠️ 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.

⚠️ 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.

3.6.9 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` directory.

  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 gvm_github]# /opt/ufm/scripts/show_ufm_status.sh
```

<table>
<thead>
<tr>
<th>ModelMain</th>
<th>Process is: [Running]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Opensm</td>
<td>Process is: [Running]</td>
</tr>
<tr>
<td>SHARP</td>
<td>Process is: [Running]</td>
</tr>
<tr>
<td>Unhealthy Ports</td>
<td>Process is: [Running]</td>
</tr>
<tr>
<td>Daily Report</td>
<td>Process is: [Running]</td>
</tr>
<tr>
<td>UFM Health</td>
<td>Process is: [Running]</td>
</tr>
<tr>
<td>UFM Telemetry</td>
<td>Process is: [Running]</td>
</tr>
</tbody>
</table>

```
[root@r-ufm77 gvm_github]# /opt/ufm/scripts/show_ufm_status.sh -e
```

<table>
<thead>
<tr>
<th>ModelMain</th>
<th>Process is: [Running]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Opensm</td>
<td>Process is: [Running]</td>
</tr>
<tr>
<td>SHARP</td>
<td>Process is: [Running]</td>
</tr>
<tr>
<td>Unhealthy Ports</td>
<td>Process is: [Running]</td>
</tr>
<tr>
<td>Daily Report</td>
<td>Process is: [Running]</td>
</tr>
<tr>
<td>UFM Health</td>
<td>Process is: [Running]</td>
</tr>
<tr>
<td>UFM Telemetry</td>
<td>Process is: [Running]</td>
</tr>
</tbody>
</table>

3.7 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.

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](#)

3.7.1 Upgrading UFM on Bare Metal Server

3.7.1.1 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.X-XXX.el7.x86_64.tgz
   - RedHat 8/CentOS 8/OEL 8: ufm-X.X.X-XXX.el8.x86_64.tgz
   - Ubuntu 18.04: ufm-X.X.X-XXX.ubuntu18.x86_64.tgz
   - Ubuntu 20.04: ufm-X.X.X-XXX.ubuntu20.x86_64.tgz
   - Ubuntu 22.04: ufm-X.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
   ```
   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).
   a. Upgrade from the previous version: the existing UFM data and configuration are preserved.
   b. 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
   ```
   `/etc/init.d/ufmd start` - Available for backward compatibility.
7. After the upgrade, remove the temporary directory

3.7.1.2 Upgrading UFM on Bare Metal - High Availability Upgrade

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:
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
```

**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).

6. On both servers, download latest UFM-HA package

```
wget https://www.mellanox.com/downloads/UFM/ufm_ha_5.5.0-9.tgz
```

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

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

```
# ufm_ha_cluster stop
```

9. On both the Master and Standby servers (starting with the Master), run the upgrade command from within the extracted HA package located in /tmp:

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

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

```
# ufm_ha_cluster start
```

### 3.7.2 Upgrading UFM on Docker Container

**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.
3.7.2.1 Upgrading UFM on Docker Container in Standalone Mode

1. Stop the UFM Enterprise service. Run:

   ```bash
   systemctl stop ufm-enterprise
   ```

2. Remove the existing docker image. Run:

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

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

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

4. Run the docker upgrade command:

   ```bash
   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:

   ```bash
   systemctl daemon-reload
   ```

6. Start UFM Enterprise service:

   ```bash
   systemctl start ufm-enterprise
   ```

3.7.2.2 Upgrading UFM Container in High Availability Mode

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).

1. Remove the old docker image from the standby server. Run:

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

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

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

   **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.**

   ```bash
   ufm_ha_cluster failover
   ```
Repeat steps 1-2 on the un-upgraded node (previous Master node).

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

```
wget https://www.mellanox.com/downloads/UFM/ufm_ha_5.5.0-9.tgz
```

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

```
# ufm_ha_cluster stop
```

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

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

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

```
# ufm_ha_cluster start
```

### 3.8 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](#)

#### 3.8.1 Uninstalling UFM in Standalone Mode

To uninstall the UFM Server:

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

```
Child interfaces are not deleted.
```

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

#### 3.8.2 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_cluster cleanup
```

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

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

3.8.3 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_cluster cleanup
```

2. Run:

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

3. Run the following on the master and slave:

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

3.9 UFM Configuration Backup and Restore

3.9.1 Overview

UFM migration enables backup and restores UFM configuration files.

3.9.2 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

⚠️ The user may also backup the UFM historical telemetry data (`-t` argument).
3.9.2.1 UFM (Bare Metal)

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

### 3.9.2.1.1 Optional Arguments

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-h</code></td>
<td><code>--help</code> show this help message and exit</td>
</tr>
<tr>
<td><code>-f</code></td>
<td><code>--backup-file BACKUP_FILE</code> full path of zip file to be generated</td>
</tr>
<tr>
<td><code>-t</code></td>
<td><code>--telemetry</code> backup UFM historical telemetry</td>
</tr>
</tbody>
</table>

3.9.2.2 UFM Docker Container

1. Backup UFM configuration. Run:
   ```bash
docker exec ufm /opt/ufm/scripts/ufm_backup.sh
   ```

2. Copy the backup file from UFM docker container to the host. Run:
   ```bash
docker cp ufm:/root/<backup file> <path on host>
   ```

3.9.2.3 UFM Appliance

1. Backup UFM configuration. Run:
   ```bash
   ufm data backup [with-telemetry]
   ```

2. Upload the backup file to a remote host. Run:
   ```bash
   ufm data upload <backup file> <upload URL>
   ```

⚠️ More details can be found in the log file `/tmp/ufm_backup.log`.

3.9.3 Restore UFM Configuration

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

3.9.3.1 UFM Bare Metal

```bash
/opt/ufm/scripts/ufm_restore.sh --help
```
3.9.3.1 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       |

3.9.3.2 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
```

3.9.3.3 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
```

⚠️ 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`
3.10 UFM Factory Reset

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

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

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.

### 3.10.1 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:

   ```
   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/
   ```

**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.

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

• `ufma_interfaces`
• `mgmt_interface`

UFM configuration upgrade during restore is not supported in UFM Appliance GEN2/GEN2.5
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:
   
   ```bash
   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):
   
   ```bash
   ufm_ha_cluster stop
   # validate that ufm is not running
   ufm_ha_cluster status
   ```

2. Run the `ufm_factory_reset.sh` script:

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

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

   Flag:

<table>
<thead>
<tr>
<th>Flag</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-y</td>
<td>Optional</td>
<td>Does not require user confirmation.</td>
</tr>
</tbody>
</table>
4 Manage Users

4.1 Accounts and Roles

Each UFM user can be assigned to one of the following four roles:

- **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.

4.2 User Account Management

By default and upon every UFM deployment, the admin user (System Admin) is generated to allow initial access to the UFM.

A user with system Administration rights can manage other users' accounts, including the creation, deletion, and modification of accounts.

To edit existing user accounts, right-click the account from the list of user accounts and perform the desired action (Change Password/Remove).
4.2.1 Add New User

1. Click the “New” button.

2. Fill in the required fields in the dialog box.

4.2.2 Edit/Remove Existing User

Right-click the account from the list of user accounts and perform the desired action (Change Password/Remove).
4.3 Authentication Methods

- 4.3.1 UFM Authentication Server
  - 4.3.1.1 Configurations of the UFM Authentication Server
- 4.3.2 Token-Based Authentication
- 4.3.3 Proxy Authentication
  - 4.3.3.1 Delegating Authentication to a Proxy
- 4.3.4 Azure AD Authentication
  - 4.3.4.1 Register UFM in Azure AD Portal
  - 4.3.4.2 Enable Azure Authentication From UFM
  - 4.3.4.3 Azure Authentication Login Page
- 4.3.5 Kerberos Authentication
  - 4.3.5.1 Setting Up Kerberos Server Machine
  - 4.3.5.2 Setting Up Kerberos Client Machine

UFM User Authentication is based on standard Apache User Authentication or Internal UFM Authentication Server.

Each Web Service client application must authenticate against the UFM server to gain access to the system.

The available authentication methods supported by UFM are as follows:
<table>
<thead>
<tr>
<th>Authentication Method</th>
<th>Description</th>
<th>UFM Related Prefix</th>
<th>REST API Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic Authentication</td>
<td>Based on user and password, provided by the client. This method is enabled by default.</td>
<td>/ufmRest</td>
<td>Basic Authentication</td>
</tr>
<tr>
<td>Session-Based Authentication</td>
<td>A stateful authentication technique where sessions are used to keep track of the authenticated user. This method is enabled by default and is used by the UFM WebUI.</td>
<td>/ufmRestV2</td>
<td>Session-Based Authentication</td>
</tr>
<tr>
<td>Client-Based Authentication</td>
<td>Refers to an end user's device proving its own identity by providing a digital certificate that can be verified by a server in order to gain access to UFM resources</td>
<td>/ufmRest</td>
<td>Client-Based Authentication</td>
</tr>
<tr>
<td>Token-Based Authentication</td>
<td>Token-based authentication is a protocol which allows users to verify their identity, and in return receive a unique access token. To use UFM, the user should create a token using UFM Web UI or UFM REST API</td>
<td>/ufmRestV3</td>
<td>Token-Based Authentication</td>
</tr>
<tr>
<td>Proxy Authentication</td>
<td>Proxy authentication delegates the user authentication to a remote Proxy server.</td>
<td>/ufmRestV3</td>
<td>N/A</td>
</tr>
<tr>
<td>Azure AD Authentication</td>
<td>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.</td>
<td>/ufmRestV2</td>
<td>N/A</td>
</tr>
<tr>
<td>Kerberos Authentication</td>
<td>Kerberos is a protocol designed to authenticate service requests between trusted hosts over an untrusted network</td>
<td>/ufmRestKrb</td>
<td>N/A</td>
</tr>
</tbody>
</table>

There are two optional services which can provide authentication handling of UFM.

1. Apache Web Server (used by default) - Standard Apache web server and supports the above-mentioned authentication methods.
2. UFM Authentication Server - a centralized HTTP server, is responsible for managing various authentication methods supported by UFM.

### 4.3.1 UFM Authentication Server

#### 4.3.1.1 Configurations of the UFM Authentication Server

The UFM Authentication Server is designed to be configurable and is initially turned on by default.

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.

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

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

### 4.3.2 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.

⚠️ 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):

<table>
<thead>
<tr>
<th>Name</th>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
</table>
| Revoke | ![Revoke Icon] | Revoke a specific token.  
⚠️ The revoked token will no longer be valid. |
| Delete | ![Delete Icon] | Delete a specific token. |
| Copy | ![Copy Icon] | Copy specific token into the clipboard. |
4.3.3 Proxy Authentication

4.3.3.1 Delegating Authentication to a Proxy

To allow a custom user authentication, you can configure UFM to delegate the user authentication to a remote Proxy server. The remote Proxy server is written by the user, thus, allowing flexibility on deciding how the authentication is performed.

By default, the feature is disabled. To activate the feature, configure `auth_proxy_enabled` with `true`.

Proxy should use `ufmRestV3` to send requests to UFM. The request header should contain a username and role. The available roles are `System_Admin`, `Fabric_Admin`, `Fabric_Operator`, and `Monitoring_Only`. If the request header is sent without a username or a role, it is rejected by the UFM.

For example:

```ini
[AuthProxy]
# Defaults to false, but set to true to enable this feature
auth_proxy_enabled = true
# HTTP Header name that will contain the username
auth_proxy_header_name = X_WEBAUTH_USER
# HTTP Header name that will contain the user roles. The available roles are as follows: System_Admin, Fabric_Admin, Fabric_Operator, and Monitoring_Only
auth_proxy_header_role = X_WEBAUTH_ROLE
# Set to `true` to enable auto sign up of users who do not exist in UFM DB. Defaults to `true`.
auth_proxy_auto_sign_up = true
# Limit where auth proxy requests come from by configuring a list of IP addresses. This can be used to prevent users spoofing the X_WEBAUTH_USER header.
# This option is required
auth_proxy_whitelist = 192.168.1.1, 192.168.1.0/24, 2001::23, 2001:r0/120
```

The following chart describes the flow:

- 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.
4.3.4 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:

4.3.4.1 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:
   a. Name: Enter a descriptive name.
   b. Supported account types: Account types that are allowed to login and use the registered application.
c. Redirect URL: select the app type Web, and Add the following redirect URL https://<ufm_server>/auth/login

Then, click Register. The app’s Overview page opens.

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

Add a client secret

<table>
<thead>
<tr>
<th>Description</th>
<th>UFM_APP_sec</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expires</td>
<td>Recommended: 180 days (6 months)</td>
</tr>
</tbody>
</table>

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

4.3.4.2 Enable Azure Authentication From UFM

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

4.3.4.3 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:
4.3.5 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.

4.3.5.1 Setting Up Kerberos Server Machine

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

1. Install the required packages:

   ```bash
   # 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:

   ```ini
   [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:

   ```bash
   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:

```bash
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.

```bash
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.

4.3.5.2 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:

   ```bash
   # 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:

   ```ini
   [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):

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'
UFM Server Health Monitoring

The UFM Server Health Monitoring module is a standalone module that monitors UFM resources and processes according to the settings in the /opt/ufm/files/conf/UFMHealthConfiguration.xml file.

For example:

- Each monitored resource or process has its own failure condition (number of retries and/or timeout), which you can configure.
- If a test fails, UFM will perform a corrective operation, if defined for the process, for example, to restart the process. You can change the configured corrective operation. If the corrective operation is set to "None", after the defined number of failures, the give-up operation is performed.
- If a test reaches the configured threshold for the number of retries, the health monitoring initiates the give-up operation defined for the process, for example, UFM failover or stop.
- By default, events and alarms are sent when a process fails, and they are also recorded in the internal log file.

Each process runs according to its own defined schedule, which you can change in the configuration file.

Changes to the configuration file take effect only after a UFM Server restart. (It is possible to kill and run in background the process nohup python /opt/ufm/ufmhealth/UfmHealthRunner.pyo &.)

You can also use the configuration file to improve disk space management by configuring:
- How often to purge MySQL binary log files.
- When to delete compressed UFM log files (according to free disk space).

The settings in the /opt/ufm/files/conf/UFMHealthConfiguration.xml file are also used to generate the UFM Health Report.

The following section describes the configuration file options for UFM server monitoring.

5.1 UFM Health Configuration

The UFM health configuration file contains three sections:
- Supported Operations—This section describes all the operations that can be used in tests, and their parameters.
- Supported Tests—This section describes all the tests. Each test includes:
  - The main test operation.
  - A corrective operation, if the main operation fails.
  - A give-up operation, if the main operation continues to fail after the corrective operation and defined number of retries.
  The number of retries and timeout is also configured for each test operation.
- Test Schedule - This section lists the tests in the order in which they are performed and their configured frequency.

The following table describes the default settings in the /opt/ufm/files/conf/UFMHealthConfiguration.xml file for each test. The tests are listed in the order in which they are performed in the default configuration file.
You might need to modify the default values depending on the size of your fabric.

For example, in a large fabric, the SM might not be responsive for `sminfo` for a long time; therefore, it is recommended to increase the values for timeout and number of retries for `SMResponseTest`.

Recommended configurations for `SMResponseTest` are:
- For a fabric with 5000 nodes:
  - Number of retries = 12
  - Frequency = 10
- For a fabric with 10000 nodes:
  - Number of retries = 12
  - Frequency = 20

<table>
<thead>
<tr>
<th>Test Name / Description</th>
<th>Test Operation</th>
<th>Corrective Operation (if Test Operation fails)</th>
<th>No. Retries / Give-up Operation</th>
<th>Test Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>CpuUsageTest</td>
<td>CPUTest</td>
<td>None</td>
<td>1 Retry None</td>
<td>1 minute</td>
</tr>
<tr>
<td>Checks total CPU utilization.</td>
<td>Tests that overall CPU usage does not exceed 80% (this percentage is configurable).</td>
<td>If UFM Event Burst Management is enabled, it is automatically initiated when the test operation fails.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AvailableDiskSpaceTest</td>
<td>FreeDiskTest</td>
<td>CleanDisk</td>
<td>3 Retries None</td>
<td>1 hour</td>
</tr>
<tr>
<td>Checks available disk space.</td>
<td>Tests that disk space usage for <code>/opt/ufm</code> does not exceed 90% (this percentage is configurable).</td>
<td>Delete compressed UFM log files under <code>/opt/ufm</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CheckIBFabricInterface</td>
<td>IBInterfaceTest</td>
<td>None</td>
<td>1 Retry None</td>
<td>35 seconds</td>
</tr>
<tr>
<td>Checks state of active fabric interface.</td>
<td>Tests that active fabric interface is up.</td>
<td>BringUpIBFabricInterface</td>
<td>Bring up the fabric interface</td>
<td></td>
</tr>
<tr>
<td>CheckIBFabricInterfaceStandby</td>
<td>IBInterfaceTestOnStandby</td>
<td>None</td>
<td>1 Retry</td>
<td>1 minute</td>
</tr>
<tr>
<td>MemoryTest</td>
<td>MemoryUsageTest</td>
<td>None</td>
<td>1 Retry None</td>
<td>1 minute</td>
</tr>
<tr>
<td>Checks total memory usage.</td>
<td>Tests that memory usage does not exceed 90% (this percentage is configurable).</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SMProcessTest</td>
<td>SMRunningTest</td>
<td>RestartProcess</td>
<td>1 Retry UFMFailoverOrDoNothing</td>
<td>10 seconds</td>
</tr>
<tr>
<td>Checks status of the OpenSM service.</td>
<td>Tests that the SM process is running.</td>
<td>Restart the SM process</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SMResponseTest</td>
<td>SMTest</td>
<td>None</td>
<td>9 Retries UFMFailoverOrDoNothing</td>
<td>10 seconds</td>
</tr>
<tr>
<td>Checks responsiveness of SM (when SM process is running).</td>
<td>Tests SM responsiveness by sending the sminfo query to SM.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Test Name / Description</td>
<td>Test Operation</td>
<td>Corrective Operation (if Test Operation fails)</td>
<td>No. Retries / Give-up Operation</td>
<td>Test Frequency</td>
</tr>
<tr>
<td>-------------------------</td>
<td>-----------------------------------------------------</td>
<td>-----------------------------------------------</td>
<td>--------------------------------</td>
<td>----------------</td>
</tr>
<tr>
<td>IbpmTest</td>
<td>ProcessIsRunningTest</td>
<td>RestartProcess</td>
<td>3 Retries</td>
<td>1 minute</td>
</tr>
<tr>
<td></td>
<td>Tests that the IBPM service is running.</td>
<td>Restart the IBPM service</td>
<td>None</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ModelMainTest</td>
<td>ProcessIsRunningTest</td>
<td>RestartProcess</td>
<td>3 Retries</td>
<td>20 seconds</td>
</tr>
<tr>
<td></td>
<td>Tests that the UFM service is running.</td>
<td>Restart the UFM service</td>
<td>UFMFailoverOrDoNothing</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HttpdTest</td>
<td>ProcessIsRunningTest</td>
<td>RestartProcess</td>
<td>3 Retries</td>
<td>20 seconds</td>
</tr>
<tr>
<td></td>
<td>Tests that the httpd service is running.</td>
<td>Restart the httpd service</td>
<td>UFMFailoverOrDoNothing</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MySqlTest</td>
<td>ConnectToMySql</td>
<td>None</td>
<td>1 Retry</td>
<td>20 seconds</td>
</tr>
<tr>
<td></td>
<td>Tests that the MySql service is running.</td>
<td></td>
<td>UFMFailoverOrDoNothing</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CleanMySql</td>
<td>AlwaysFailTest</td>
<td>PurgeMySqlLogs</td>
<td>1 Retry</td>
<td>24 hours</td>
</tr>
<tr>
<td>Purges MySql Logs</td>
<td>Fails the test in order to perform the corrective action.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UFMServerVersionTest</td>
<td>UfmVersionTest</td>
<td>None</td>
<td>1 Retry</td>
<td>24 hours</td>
</tr>
<tr>
<td>Checks UFM software version and build.</td>
<td>Returns UFM software version information.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UFMServerLicenseTest</td>
<td>UfmLicenseTest</td>
<td>None</td>
<td>1 Retry</td>
<td>24 hours</td>
</tr>
<tr>
<td>Checks UFM License information.</td>
<td>Returns UFM License information.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UFMServerHAConfigurationTest</td>
<td>UfmHAConfigurationTest</td>
<td>None</td>
<td>1 Retry</td>
<td>24 hours</td>
</tr>
<tr>
<td>(HA only) Checks the configuration on master and standby.</td>
<td>Returns information about the master and standby UFM servers.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UFMMemoryTest</td>
<td>UfmMemoryUsageTest</td>
<td>None</td>
<td>1 Retry</td>
<td>1 minute</td>
</tr>
<tr>
<td>Checks available UFM memory.</td>
<td>Tests that UFM memory usage does not exceed 80% (this percentage is configurable).</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UFMCPUUsageTest</td>
<td>CPUTest</td>
<td>None</td>
<td>1 Retry</td>
<td>1 minute</td>
</tr>
<tr>
<td>Checks UFM CPU utilization.</td>
<td>Tests that UFM CPU usage does not exceed 60% (this percentage is configurable).</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CheckDrbdTcpConnectionPerformanceTest (HA only) Checks the tcp connection between master and standby.</td>
<td>TcpConnectionPerformanceTest</td>
<td>None</td>
<td>2 Retry</td>
<td>10 minutes</td>
</tr>
<tr>
<td></td>
<td>Tests that bandwidth is greater than 100 Mb/sec and latency is less than 70 usec (configurable).</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
5.1.1 UFM Core Files Tracking

To receive a notification every time OpenSM or ibpm creates a core dump, please refer to the list of all current core dumps of OpenSM and ibpm in the UFM health report.

To receive core dump notifications, do the following:

1. Set the core_dumps_directory field in the gv.cfg file to point to the location where all core dumps are created (by default, this location is set to /tmp).

   core_dumps_directory = /tmp

2. Set the naming convention for the core dump file. The name must include the directory configured in the step above.
   The convention we recommend is:

   ```
   echo "*/tmp/%t.core.%e.%p.%h" > /proc/sys/kernel/core_pattern
   ```

3. Make sure core dumps directory setting is persistent between reboots. Add the kernel.core_pattern parameter with the desired file name format to the /etc/sysctl.conf file. Example:

   ```
   kernel.core_pattern=/tmp/%t.core.%e.%p.%h
   ```

4. Configure the core file size to be unlimited.

   ```
   ulimit -c unlimited
   ```

5. (Only on UFM HA master) Update the UFM configuration file gv.cfg to enable core dump tracking.

   ```
   track_core_dumps = yes
   ```

5.2 Example of Health Configuration

The default configuration for the overall memory test in the opt/ufm/files/conf/UFMHealthConfiguration.xml file is:

```xml
<Test Name="MemoryTest" NumOfRetriesBeforeGiveup="3" RetryTimeoutInSeconds="10">
  <TestOperation Name="MemoryUsageTest">
    <Parameters>
      <Parameter Name="ThresholdInPercents" Value="90"/>
    </Parameters>
  </TestOperation>
  <CorrectiveOperation Name="None"/>
  <GiveupOperation Name="None"/>
</Test>
```
This configuration tests the available memory. If memory usage exceeds 90%, the test is repeated up to 3 times at 10 second intervals, or until memory usage drops to below 90%. No corrective action is taken and no action is taken after 3 retries.

To test with a usage threshold of 80%, and to initiate UFM failover or stop UFM after three retries, change the configuration to:

```xml
<Test Name="MemoryTest" NumOfRetriesBeforeGiveup="3" RetryTimeoutInSeconds="10">
  <TestOperation Name="MemoryUsageTest">
    <Parameters>
      <Parameter Name="ThresholdInPercents" Value="80"/>
    </Parameters>
  </TestOperation>
  <CorrectiveOperation Name="None"/>
  <GiveupOperation Name="UFMFailoverOrStop"/>
</Test>
```

5.2.1 Event Burst Management

UFM event burst management can lower the overall CPU usage following an event burst by suppressing events. Event burst management is configured in the `gv.cfg` configuration file.

When the overall CPU usage exceeds the threshold configured by the CpuUsageTest in the `/opt/ufm/files/conf/UFMHealthConfiguration.xml` file, a High CPU Utilization event occurs.

This event initiates the UFM event burst management, which:

- Suppresses events. The default level of suppression enables critical events only.
- If, after a specified period of time (30 seconds, by default), no further High CPU Utilization event occurs, the UFM server enables all events.

To modify Event burst management configuration, change the following parameters in the `gv.cfg` file:

```plaintext
# The events' level in case events are suppressed (the possible levels are disable_all_events,
# enable_critical_events, and enable_all_events)
# The entire feature can be turned off using the level *enable_all_events*
suppress_events_level = enable_critical_events
# The amount of time in seconds which events are suppressed
suppress_events_timeout = 30
```
6 Events and Alarms

UFM offers comprehensive diagnostics for your InfiniBand fabric, covering a range of categories:

1. Fabric configurations
2. Fabric topology
3. Hardware issues
4. Communication errors
5. Maintenance
6. Security
7. Switch module status
8. NVIDIA SHARP notifications

Events are notifications generated by UFM, indicating issues within the mentioned categories in the InfiniBand fabric. On the other hand, alerts are urgent notifications derived from events (many events can be configured as alarms based on customer preferences).

These detections are performed both before running applications and during standard operation. They help troubleshoot and notify network administrators of potential network issues before they escalate.

Events can originate from various sources:

- SM traps
- SHARP AM traps
- UFM internal analysis, encompassing:
  - Internal detection of topology changes
  - Internal fabric analysis (based on IBDiagnet)
  - Internal monitoring of managed switches
  - Maintenance activities (device action tracking, licensing, cable integrity)
- Threshold-crossing events determined by telemetry counter readings

<table>
<thead>
<tr>
<th></th>
<th>WebUI</th>
<th>REST API</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Events</strong></td>
<td>UFM events can be viewed via the Events and Alarms WebUI view. Refer to Events &amp; Alarms.</td>
<td>Events REST API</td>
</tr>
<tr>
<td></td>
<td>For device-specific events, refer to the Events &amp; Alarms.</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>Configuration of events is managed within the Events Policy Tab in the Settings window</td>
<td>Events Policy REST API</td>
</tr>
<tr>
<td><strong>Alarms</strong></td>
<td>UFM alarms can be viewed via the Events and Alarms WebUI view. Refer to Events &amp; Alarms.</td>
<td>Alarms REST API</td>
</tr>
<tr>
<td></td>
<td>Configuration of alarms is managed within the Events Policy Tab in the Settings window</td>
<td>N/A</td>
</tr>
</tbody>
</table>

For showing all the UFM supported events, refer to Threshold-Crossing Events Reference.
6.1 Threshold-Crossing Events Reference

This reference lists the threshold-based events that UFM supports, each including a set of attributes listed below. You can view these messages through TBD. For details about configuring notifications for these events, refer to TBD.

- Event ID: Unique event identifier
- Event Name: Short description of the event
- To Log: A flag which defines whether the event will be sent to UFM events log or not (1 or 0, respectively).
- Alarm: A flag which defines whether UFM alarm is generated when the specified event is triggered by UFM (1 means an alarm is generated). Alarms are used to allow a notification with significant indication.
- Default Severity: Indicates the alarm severity (Info, Warning, Minor, Critical)
- Default Threshold: Event dependent threshold (for example, event occurrence, counter threshold or temperature threshold).
- Default TTL: 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.
- Related Object: The object (context) to which the UFM event is related to (port, switch, gateway, grid, etc).
- Category: Indicates the category to which the event is related to.
- Source: Indicates the origin source of the specified event (SM, Telemetry, Licensing, UFM)

For information about defining event policy, see Configuring Event Management.

<table>
<thead>
<tr>
<th>Event ID</th>
<th>Event Name</th>
<th>T o Log</th>
<th>Alarm</th>
<th>Default Severity</th>
<th>Default Threshold</th>
<th>Related Object</th>
<th>Category</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>64</td>
<td>GID Address In Service</td>
<td>0</td>
<td>Info</td>
<td>1</td>
<td>300</td>
<td>Port</td>
<td>Fabric Notification</td>
<td>SM</td>
</tr>
<tr>
<td>65</td>
<td>GID Address Out of Service</td>
<td>0</td>
<td>Warn</td>
<td>1</td>
<td>300</td>
<td>Port</td>
<td>Fabric Notification</td>
<td>SM</td>
</tr>
<tr>
<td>66</td>
<td>New MCast Group Created</td>
<td>0</td>
<td>Info</td>
<td>1</td>
<td>300</td>
<td>Port</td>
<td>Fabric Notification</td>
<td>SM</td>
</tr>
<tr>
<td>67</td>
<td>MCast Group Deleted</td>
<td>0</td>
<td>Info</td>
<td>1</td>
<td>300</td>
<td>Port</td>
<td>Fabric Notification</td>
<td>SM</td>
</tr>
<tr>
<td>110</td>
<td>Symbol Error</td>
<td>1</td>
<td>Warn</td>
<td>200</td>
<td>300</td>
<td>Port</td>
<td>Hardware</td>
<td>Telemetry</td>
</tr>
<tr>
<td>111</td>
<td>Link Error Recovery</td>
<td>1</td>
<td>Minor</td>
<td>1</td>
<td>300</td>
<td>Port</td>
<td>Hardware</td>
<td>Telemetry</td>
</tr>
<tr>
<td>Event ID</td>
<td>Event Name</td>
<td>To Log</td>
<td>Alarm</td>
<td>Default Severity</td>
<td>Default Threshold</td>
<td>Related Object</td>
<td>Category</td>
<td>Source</td>
</tr>
<tr>
<td>----------</td>
<td>------------------------------------------------</td>
<td>--------</td>
<td>-------</td>
<td>------------------</td>
<td>------------------</td>
<td>---------------</td>
<td>--------------</td>
<td>----------------</td>
</tr>
<tr>
<td>112</td>
<td>Link Downed</td>
<td>1</td>
<td>1</td>
<td>Critical</td>
<td>300</td>
<td>Port</td>
<td>Hardware</td>
<td>Telemetry</td>
</tr>
<tr>
<td>113</td>
<td>Port Receive Errors</td>
<td>1</td>
<td>1</td>
<td>Minor</td>
<td>300</td>
<td>Port</td>
<td>Hardware</td>
<td>Telemetry</td>
</tr>
<tr>
<td>114</td>
<td>Port Receive Remote Physical Errors</td>
<td>0</td>
<td>0</td>
<td>Minor</td>
<td>300</td>
<td>Port</td>
<td>Hardware</td>
<td>Telemetry</td>
</tr>
<tr>
<td>115</td>
<td>Port Receive Switch Relay Errors</td>
<td>1</td>
<td>1</td>
<td>Minor</td>
<td>999</td>
<td>Port</td>
<td>Fabric Configurati on</td>
<td>Telemetry</td>
</tr>
<tr>
<td>116</td>
<td>Port Xmit Discards</td>
<td>1</td>
<td>1</td>
<td>Minor</td>
<td>200</td>
<td>Port</td>
<td>Communication Error</td>
<td>Telemetry</td>
</tr>
<tr>
<td>117</td>
<td>Port Xmit Constraint Errors</td>
<td>1</td>
<td>1</td>
<td>Minor</td>
<td>200</td>
<td>Port</td>
<td>Communication Error</td>
<td>Telemetry</td>
</tr>
<tr>
<td>118</td>
<td>Port Receive Constraint Errors</td>
<td>1</td>
<td>1</td>
<td>Minor</td>
<td>200</td>
<td>Port</td>
<td>Communication Error</td>
<td>Telemetry</td>
</tr>
<tr>
<td>119</td>
<td>Local Link Integrity Errors</td>
<td>1</td>
<td>1</td>
<td>Minor</td>
<td>5</td>
<td>Port</td>
<td>Hardware</td>
<td>Telemetry</td>
</tr>
<tr>
<td>120</td>
<td>Excessive Buffer Overrun Errors</td>
<td>1</td>
<td>1</td>
<td>Minor</td>
<td>100</td>
<td>Port</td>
<td>Communication Error</td>
<td>Telemetry</td>
</tr>
<tr>
<td>121</td>
<td>VL15 Dropped</td>
<td>1</td>
<td>1</td>
<td>Minor</td>
<td>50</td>
<td>Port</td>
<td>Communication Error</td>
<td>Telemetry</td>
</tr>
<tr>
<td>122</td>
<td>Congested Bandwidth (%) Threshold Reached</td>
<td>1</td>
<td>1</td>
<td>Minor</td>
<td>10</td>
<td>Port</td>
<td>Hardware</td>
<td>Telemetry</td>
</tr>
<tr>
<td>123</td>
<td>Port Bandwidth (%) Threshold Reached</td>
<td>1</td>
<td>1</td>
<td>Minor</td>
<td>95</td>
<td>Port</td>
<td>Communication Error</td>
<td>Telemetry</td>
</tr>
<tr>
<td>130</td>
<td>Non-optimal link width</td>
<td>1</td>
<td>1</td>
<td>Minor</td>
<td>0</td>
<td>Port</td>
<td>Hardware</td>
<td>SM</td>
</tr>
<tr>
<td>134</td>
<td>T4 Port Congested Bandwidth</td>
<td>1</td>
<td>1</td>
<td>Warning</td>
<td>10</td>
<td>Port</td>
<td>Communication Error</td>
<td>Telemetry</td>
</tr>
<tr>
<td>141</td>
<td>Flow Control Update Watchdog Timer Expired</td>
<td>1</td>
<td>0</td>
<td>Warning</td>
<td>1</td>
<td>Port</td>
<td>Hardware</td>
<td>SM</td>
</tr>
<tr>
<td>144</td>
<td>Capability Mask Modified</td>
<td>1</td>
<td>0</td>
<td>Info</td>
<td>1</td>
<td>Port</td>
<td>Fabric Notificati on</td>
<td>SM</td>
</tr>
<tr>
<td>145</td>
<td>System Image GUID changed</td>
<td>1</td>
<td>0</td>
<td>Info</td>
<td>1</td>
<td>Port</td>
<td>Communication Error</td>
<td>SM</td>
</tr>
<tr>
<td>156</td>
<td>Link Speed Enforcement Disabled</td>
<td>1</td>
<td>0</td>
<td>Critical</td>
<td>0</td>
<td>Site</td>
<td>Fabric Notificati on</td>
<td>SM</td>
</tr>
<tr>
<td>250</td>
<td>Running in Limited Mode</td>
<td>1</td>
<td>1</td>
<td>Critical</td>
<td>0</td>
<td>Grid</td>
<td>Maintenance</td>
<td>Licensing</td>
</tr>
<tr>
<td>251</td>
<td>Switching to Limited Mode</td>
<td>1</td>
<td>1</td>
<td>Critical</td>
<td>0</td>
<td>Grid</td>
<td>Maintenance</td>
<td>Licensing</td>
</tr>
<tr>
<td>Event ID</td>
<td>Event Name</td>
<td>T o l o g</td>
<td>Alar m</td>
<td>Default Severity</td>
<td>Default Threshold</td>
<td>Related Object</td>
<td>Category</td>
<td>Source</td>
</tr>
<tr>
<td>----------</td>
<td>------------</td>
<td>----------</td>
<td>--------</td>
<td>------------------</td>
<td>-------------------</td>
<td>----------------</td>
<td>----------</td>
<td>--------</td>
</tr>
<tr>
<td>252</td>
<td>License Expired</td>
<td>1</td>
<td>1</td>
<td>Warning</td>
<td>1</td>
<td>Grid</td>
<td>Maintenance Licensing</td>
<td></td>
</tr>
<tr>
<td>253</td>
<td>Duplicated licenses</td>
<td>1</td>
<td>0</td>
<td>Critical</td>
<td>1</td>
<td>Grid</td>
<td>Maintenance Licensing</td>
<td></td>
</tr>
<tr>
<td>254</td>
<td>License Limit Exceeded</td>
<td>1</td>
<td>0</td>
<td>Critical</td>
<td>1</td>
<td>Grid</td>
<td>Maintenance Licensing</td>
<td></td>
</tr>
<tr>
<td>255</td>
<td>License is About to Expire</td>
<td>1</td>
<td>0</td>
<td>Warning</td>
<td>1</td>
<td>Grid</td>
<td>Maintenance Licensing</td>
<td></td>
</tr>
<tr>
<td>256</td>
<td>Bad M_Key</td>
<td>1</td>
<td>0</td>
<td>Minor</td>
<td>1</td>
<td>300</td>
<td>Port Security SM</td>
<td></td>
</tr>
<tr>
<td>257</td>
<td>Bad P_Key</td>
<td>1</td>
<td>0</td>
<td>Minor</td>
<td>1</td>
<td>300</td>
<td>Port Security SM</td>
<td></td>
</tr>
<tr>
<td>258</td>
<td>Bad Q_Key</td>
<td>1</td>
<td>0</td>
<td>Minor</td>
<td>1</td>
<td>300</td>
<td>Port Security SM</td>
<td></td>
</tr>
<tr>
<td>259</td>
<td>Bad P_Key Switch External Port</td>
<td>1</td>
<td>0</td>
<td>Critical</td>
<td>1</td>
<td>300</td>
<td>Port Security SM</td>
<td></td>
</tr>
<tr>
<td>328</td>
<td>Link is Up</td>
<td>1</td>
<td>0</td>
<td>Info</td>
<td>1</td>
<td>0</td>
<td>Link Fabric Topology SM</td>
<td></td>
</tr>
<tr>
<td>329</td>
<td>Link is Down</td>
<td>1</td>
<td>0</td>
<td>Warning</td>
<td>1</td>
<td>0</td>
<td>Site Fabric Topology SM</td>
<td></td>
</tr>
<tr>
<td>331</td>
<td>Node is Down</td>
<td>1</td>
<td>0</td>
<td>Warning</td>
<td>1</td>
<td>0</td>
<td>Site Fabric Topology SM</td>
<td></td>
</tr>
<tr>
<td>332</td>
<td>Node is Up</td>
<td>1</td>
<td>0</td>
<td>Info</td>
<td>1</td>
<td>300</td>
<td>Site Fabric Topology SM</td>
<td></td>
</tr>
<tr>
<td>336</td>
<td>Port Action Succeeded</td>
<td>1</td>
<td>0</td>
<td>Info</td>
<td>1</td>
<td>0</td>
<td>Port Maintenance UFM</td>
<td></td>
</tr>
<tr>
<td>337</td>
<td>Port Action Failed</td>
<td>1</td>
<td>0</td>
<td>Minor</td>
<td>1</td>
<td>0</td>
<td>Port Maintenance UFM</td>
<td></td>
</tr>
<tr>
<td>338</td>
<td>Device Action Succeeded</td>
<td>1</td>
<td>0</td>
<td>Info</td>
<td>1</td>
<td>0</td>
<td>Port Maintenance UFM</td>
<td></td>
</tr>
<tr>
<td>339</td>
<td>Device Action Failed</td>
<td>1</td>
<td>0</td>
<td>Minor</td>
<td>1</td>
<td>0</td>
<td>Port Maintenance UFM</td>
<td></td>
</tr>
<tr>
<td>344</td>
<td>Partial Switch ASIC Failure</td>
<td>1</td>
<td>1</td>
<td>Critical</td>
<td>1</td>
<td>0</td>
<td>Switch Maintenance UFM</td>
<td></td>
</tr>
<tr>
<td>370</td>
<td>Gateway Ethernet Link State Changed</td>
<td>1</td>
<td>0</td>
<td>Warning</td>
<td>1</td>
<td>0</td>
<td>Gateway SM</td>
<td></td>
</tr>
<tr>
<td>371</td>
<td>Gateway Reregister Event Received</td>
<td>1</td>
<td>0</td>
<td>Warning</td>
<td>1</td>
<td>0</td>
<td>Gateway SM</td>
<td></td>
</tr>
<tr>
<td>372</td>
<td>Number of Gateways Changed</td>
<td>1</td>
<td>0</td>
<td>Warning</td>
<td>1</td>
<td>0</td>
<td>Gateway SM</td>
<td></td>
</tr>
<tr>
<td>373</td>
<td>Gateway will be Rebooted</td>
<td>1</td>
<td>0</td>
<td>Warning</td>
<td>1</td>
<td>0</td>
<td>Gateway SM</td>
<td></td>
</tr>
<tr>
<td>374</td>
<td>Gateway Reloading Finished</td>
<td>1</td>
<td>0</td>
<td>Info</td>
<td>1</td>
<td>0</td>
<td>Gateway SM</td>
<td></td>
</tr>
<tr>
<td>Event ID</td>
<td>Event Name</td>
<td>To Log</td>
<td>Alarms</td>
<td>Default Severity</td>
<td>Default Threshold</td>
<td>Related Object</td>
<td>Category</td>
<td>Source</td>
</tr>
<tr>
<td>----------</td>
<td>-------------------------------------------------</td>
<td>--------</td>
<td>--------</td>
<td>------------------</td>
<td>-------------------</td>
<td>---------------</td>
<td>--------------</td>
<td>--------</td>
</tr>
<tr>
<td>380</td>
<td>Switch Upgrade Error</td>
<td>1</td>
<td>1</td>
<td>Critical</td>
<td>0</td>
<td>Switch</td>
<td>Maintenanc e</td>
<td>UFM</td>
</tr>
<tr>
<td>381</td>
<td>Switch Upgrade Failed</td>
<td>1</td>
<td>0</td>
<td>Info</td>
<td>0</td>
<td>Switch</td>
<td>Maintenanc e</td>
<td>UFM</td>
</tr>
<tr>
<td>328</td>
<td>Module status NOT PRESENT</td>
<td>1</td>
<td>1</td>
<td>Warning</td>
<td>420</td>
<td>Switch</td>
<td>Module Status</td>
<td>UFM</td>
</tr>
<tr>
<td>383</td>
<td>Host Upgrade Failed</td>
<td>1</td>
<td>0</td>
<td>Info</td>
<td>0</td>
<td>Computer</td>
<td>Maintenanc e</td>
<td>UFM</td>
</tr>
<tr>
<td>384</td>
<td>Switch Module Powered Off</td>
<td>1</td>
<td>1</td>
<td>Info</td>
<td>420</td>
<td>Switch</td>
<td>Module Status</td>
<td>UFM</td>
</tr>
<tr>
<td>385</td>
<td>Switch FW Upgrade Started</td>
<td>1</td>
<td>0</td>
<td>Info</td>
<td>0</td>
<td>Switch</td>
<td>Maintenanc e</td>
<td>UFM</td>
</tr>
<tr>
<td>386</td>
<td>Switch SW Upgrade Started</td>
<td>1</td>
<td>0</td>
<td>Info</td>
<td>0</td>
<td>Switch</td>
<td>Maintenanc e</td>
<td>UFM</td>
</tr>
<tr>
<td>387</td>
<td>Switch Upgrade Finished</td>
<td>1</td>
<td>0</td>
<td>Info</td>
<td>0</td>
<td>Switch</td>
<td>Maintenanc e</td>
<td>UFM</td>
</tr>
<tr>
<td>388</td>
<td>Host FW Upgrade Started</td>
<td>1</td>
<td>0</td>
<td>Info</td>
<td>0</td>
<td>Compu ter</td>
<td>Maintenanc e</td>
<td>UFM</td>
</tr>
<tr>
<td>389</td>
<td>Host SW Upgrade Started</td>
<td>1</td>
<td>0</td>
<td>Info</td>
<td>0</td>
<td>Compu ter</td>
<td>Maintenanc e</td>
<td>UFM</td>
</tr>
<tr>
<td>391</td>
<td>Switch Module Removed</td>
<td>1</td>
<td>0</td>
<td>Info</td>
<td>0</td>
<td>Switch</td>
<td>Fabric Notificatio n</td>
<td>Switch</td>
</tr>
<tr>
<td>392</td>
<td>Module Temperature Threshold Reached</td>
<td>1</td>
<td>0</td>
<td>Info</td>
<td>40</td>
<td>Module</td>
<td>Hardware</td>
<td>Switch</td>
</tr>
<tr>
<td>393</td>
<td>Switch Module Added</td>
<td>1</td>
<td>0</td>
<td>Info</td>
<td>0</td>
<td>Switch</td>
<td>Fabric Notificatio n</td>
<td>Switch</td>
</tr>
<tr>
<td>394</td>
<td>Module Status FAULT</td>
<td>1</td>
<td>1</td>
<td>Critical</td>
<td>1</td>
<td>Switch</td>
<td>Module Status</td>
<td>Switch</td>
</tr>
<tr>
<td>395</td>
<td>Device Action Started</td>
<td>1</td>
<td>0</td>
<td>Info</td>
<td>0</td>
<td>Port</td>
<td>Maintenanc e</td>
<td>UFM</td>
</tr>
<tr>
<td>396</td>
<td>Site Action Started</td>
<td>1</td>
<td>0</td>
<td>Info</td>
<td>0</td>
<td>Port</td>
<td>Maintenanc e</td>
<td>UFM</td>
</tr>
<tr>
<td>397</td>
<td>Site Action Failed</td>
<td>1</td>
<td>0</td>
<td>Minor</td>
<td>0</td>
<td>Port</td>
<td>Maintenanc e</td>
<td>UFM</td>
</tr>
<tr>
<td>398</td>
<td>Switch Chip Added</td>
<td>1</td>
<td>0</td>
<td>Info</td>
<td>0</td>
<td>Switch</td>
<td>Fabric Notificatio n</td>
<td>Switch</td>
</tr>
<tr>
<td>399</td>
<td>Switch Chip Removed</td>
<td>1</td>
<td>0</td>
<td>Critical</td>
<td>0</td>
<td>Switch</td>
<td>Fabric Notificatio n</td>
<td>Switch</td>
</tr>
<tr>
<td>Event ID</td>
<td>Event Name</td>
<td>To Log</td>
<td>Alarm</td>
<td>Default Severity</td>
<td>Default Threshold</td>
<td>Defau lt TTL</td>
<td>Related Object</td>
<td>Category</td>
</tr>
<tr>
<td>----------</td>
<td>------------------------------------------------</td>
<td>--------</td>
<td>-------</td>
<td>------------------</td>
<td>-------------------</td>
<td>--------------</td>
<td>----------------</td>
<td>--------------</td>
</tr>
<tr>
<td>403</td>
<td>Device Pending Reboot</td>
<td>1</td>
<td>1</td>
<td>Waring</td>
<td>0</td>
<td>300</td>
<td>Device</td>
<td>Maintenanc e</td>
</tr>
<tr>
<td>404</td>
<td>System Information is missing</td>
<td>1</td>
<td>1</td>
<td>Waring</td>
<td>1</td>
<td>300</td>
<td>Switch</td>
<td>Communication Error</td>
</tr>
<tr>
<td>405</td>
<td>Switch Identity Validation Failed</td>
<td>1</td>
<td>1</td>
<td>Waring</td>
<td>1</td>
<td>300</td>
<td>Switch</td>
<td>Communication Error</td>
</tr>
<tr>
<td>406</td>
<td>Switch System Information is missing</td>
<td>1</td>
<td>1</td>
<td>Waring</td>
<td>1</td>
<td>300</td>
<td>Switch</td>
<td>Communication Error</td>
</tr>
<tr>
<td>407</td>
<td>COMEX Ambient Temperature Threshold Reached</td>
<td>1</td>
<td>1</td>
<td>Minor</td>
<td>60</td>
<td>300</td>
<td>Switch</td>
<td>Hardware</td>
</tr>
<tr>
<td>408</td>
<td>Switch is Unresponsive</td>
<td>1</td>
<td>1</td>
<td>Critical</td>
<td>1</td>
<td>300</td>
<td>Switch</td>
<td>Communication Error</td>
</tr>
<tr>
<td>502</td>
<td>Device Upgrade Finished</td>
<td>1</td>
<td>0</td>
<td>Info</td>
<td>1</td>
<td>300</td>
<td>Device</td>
<td>Maintenanc e</td>
</tr>
<tr>
<td>506</td>
<td>Device Upgrade Finished</td>
<td>1</td>
<td>0</td>
<td>Info</td>
<td>1</td>
<td>300</td>
<td>Device</td>
<td>Maintenanc e</td>
</tr>
<tr>
<td>508</td>
<td>Core Dump Created</td>
<td>1</td>
<td>1</td>
<td>Info</td>
<td>1</td>
<td>300</td>
<td>Grid</td>
<td>Maintenanc e</td>
</tr>
<tr>
<td>510</td>
<td>SM Failover</td>
<td>0</td>
<td>1</td>
<td>Critical</td>
<td>1</td>
<td>300</td>
<td>Grid</td>
<td>Fabric Notificatio n</td>
</tr>
<tr>
<td>511</td>
<td>SM State Change</td>
<td>0</td>
<td>1</td>
<td>Info</td>
<td>1</td>
<td>300</td>
<td>Grid</td>
<td>Fabric Notificatio n</td>
</tr>
<tr>
<td>512</td>
<td>SM UP</td>
<td>0</td>
<td>1</td>
<td>Info</td>
<td>1</td>
<td>300</td>
<td>Grid</td>
<td>Fabric Notificatio n</td>
</tr>
<tr>
<td>513</td>
<td>SM System Log Message</td>
<td>0</td>
<td>1</td>
<td>Minor</td>
<td>1</td>
<td>300</td>
<td>Grid</td>
<td>Fabric Notificatio n</td>
</tr>
<tr>
<td>514</td>
<td>SM LID Change</td>
<td>0</td>
<td>1</td>
<td>Waring</td>
<td>1</td>
<td>300</td>
<td>Grid</td>
<td>Fabric Notificatio n</td>
</tr>
<tr>
<td>515</td>
<td>Fabric Health Report Info</td>
<td>1</td>
<td>1</td>
<td>Info</td>
<td>1</td>
<td>300</td>
<td>Grid</td>
<td>Fabric Notificatio n</td>
</tr>
<tr>
<td>516</td>
<td>Fabric Health Report Warning</td>
<td>1</td>
<td>1</td>
<td>Waring</td>
<td>1</td>
<td>300</td>
<td>Grid</td>
<td>Fabric Notificatio n</td>
</tr>
<tr>
<td>517</td>
<td>Fabric Health Report Error</td>
<td>1</td>
<td>1</td>
<td>Critical</td>
<td>1</td>
<td>300</td>
<td>Grid</td>
<td>Fabric Notificatio n</td>
</tr>
<tr>
<td>518</td>
<td>UFM-related process is down</td>
<td>1</td>
<td>1</td>
<td>Critical</td>
<td>1</td>
<td>300</td>
<td>Grid</td>
<td>Maintenanc e</td>
</tr>
<tr>
<td>Event ID</td>
<td>Event Name</td>
<td>To Log</td>
<td>Alarm</td>
<td>Default Severity</td>
<td>Default Threshold</td>
<td>Related Object</td>
<td>Category</td>
<td>Source</td>
</tr>
<tr>
<td>----------</td>
<td>------------------------------------------------</td>
<td>--------</td>
<td>-------</td>
<td>------------------</td>
<td>-------------------</td>
<td>----------------</td>
<td>----------</td>
<td>----------</td>
</tr>
<tr>
<td>519</td>
<td>Logs purge failure</td>
<td>1</td>
<td>1</td>
<td>Minor</td>
<td>1</td>
<td>300</td>
<td>Grid</td>
<td>Maintainance</td>
</tr>
<tr>
<td>520</td>
<td>Restart of UFM-related process succeeded</td>
<td>1</td>
<td>1</td>
<td>Info</td>
<td>1</td>
<td>300</td>
<td>Grid</td>
<td>Maintainance</td>
</tr>
<tr>
<td>521</td>
<td>UFM is being stopped</td>
<td>1</td>
<td>1</td>
<td>Critical</td>
<td>1</td>
<td>300</td>
<td>Grid</td>
<td>Maintainance</td>
</tr>
<tr>
<td>522</td>
<td>UFM is being restarted</td>
<td>1</td>
<td>1</td>
<td>Critical</td>
<td>1</td>
<td>300</td>
<td>Grid</td>
<td>Maintainance</td>
</tr>
<tr>
<td>523</td>
<td>UFM failover is being attempted</td>
<td>1</td>
<td>1</td>
<td>Info</td>
<td>1</td>
<td>300</td>
<td>Grid</td>
<td>Maintainance</td>
</tr>
<tr>
<td>524</td>
<td>UFM cannot connect to DB</td>
<td>1</td>
<td>1</td>
<td>Critical</td>
<td>1</td>
<td>300</td>
<td>Grid</td>
<td>Maintainance</td>
</tr>
<tr>
<td>525</td>
<td>Disk utilization threshold reached</td>
<td>1</td>
<td>1</td>
<td>Critical</td>
<td>1</td>
<td>300</td>
<td>Grid</td>
<td>Maintainance</td>
</tr>
<tr>
<td>526</td>
<td>Memory utilization threshold reached</td>
<td>1</td>
<td>1</td>
<td>Critical</td>
<td>1</td>
<td>300</td>
<td>Grid</td>
<td>Maintainance</td>
</tr>
<tr>
<td>527</td>
<td>CPU utilization threshold reached</td>
<td>1</td>
<td>1</td>
<td>Critical</td>
<td>1</td>
<td>300</td>
<td>Grid</td>
<td>Maintainance</td>
</tr>
<tr>
<td>528</td>
<td>Fabric interface is down</td>
<td>1</td>
<td>1</td>
<td>Critical</td>
<td>1</td>
<td>300</td>
<td>Grid</td>
<td>Maintainance</td>
</tr>
<tr>
<td>529</td>
<td>UFM standby server problem</td>
<td>1</td>
<td>1</td>
<td>Critical</td>
<td>1</td>
<td>300</td>
<td>Grid</td>
<td>Maintainance</td>
</tr>
<tr>
<td>530</td>
<td>SM is down</td>
<td>1</td>
<td>1</td>
<td>Critical</td>
<td>1</td>
<td>300</td>
<td>Grid</td>
<td>Maintainance</td>
</tr>
<tr>
<td>531</td>
<td>DRBD Bad Condition</td>
<td>1</td>
<td>1</td>
<td>Critical</td>
<td>1</td>
<td>300</td>
<td>Grid</td>
<td>Maintainance</td>
</tr>
<tr>
<td>532</td>
<td>Remote UFM-SM Sync</td>
<td>1</td>
<td>1</td>
<td>Info</td>
<td>1</td>
<td>0</td>
<td>Grid</td>
<td>Maintainance</td>
</tr>
<tr>
<td>533</td>
<td>Remote UFM-SM problem</td>
<td>1</td>
<td>1</td>
<td>Critical</td>
<td>1</td>
<td>0</td>
<td>Site</td>
<td>Maintainance</td>
</tr>
<tr>
<td>535</td>
<td>MH Purge Failed</td>
<td>1</td>
<td>1</td>
<td>Warning</td>
<td>1</td>
<td>300</td>
<td>Grid</td>
<td>Maintainance</td>
</tr>
<tr>
<td>536</td>
<td>UFM Health Watchdog Info</td>
<td>1</td>
<td>1</td>
<td>Info</td>
<td>1</td>
<td>300</td>
<td>Grid</td>
<td>Maintainance</td>
</tr>
<tr>
<td>537</td>
<td>UFM Health Watchdog Critical</td>
<td>1</td>
<td>1</td>
<td>Critical</td>
<td>1</td>
<td>300</td>
<td>Grid</td>
<td>Maintainance</td>
</tr>
<tr>
<td>538</td>
<td>Time Diff Between HA Servers</td>
<td>1</td>
<td>1</td>
<td>Warning</td>
<td>1</td>
<td>300</td>
<td>Grid</td>
<td>Maintainance</td>
</tr>
<tr>
<td>539</td>
<td>DRBD TCP Connection Performance</td>
<td>1</td>
<td>1</td>
<td>Warning</td>
<td>1</td>
<td>900</td>
<td>Grid</td>
<td>Maintainance</td>
</tr>
<tr>
<td>540</td>
<td>Daily Report Completed successfully</td>
<td>1</td>
<td>0</td>
<td>Info</td>
<td>1</td>
<td>300</td>
<td>Grid</td>
<td>Maintainance</td>
</tr>
<tr>
<td>Event ID</td>
<td>Event Name</td>
<td>Tolog</td>
<td>Alar m</td>
<td>Default Severity</td>
<td>Default Threshold</td>
<td>Related Object</td>
<td>Category</td>
<td>Source</td>
</tr>
<tr>
<td>---------</td>
<td>--------------------------------------------------------</td>
<td>-------</td>
<td>--------</td>
<td>------------------</td>
<td>-------------------</td>
<td>----------------</td>
<td>------------</td>
<td>--------</td>
</tr>
<tr>
<td>541</td>
<td>Daily Report Completed with Error</td>
<td>1</td>
<td>0</td>
<td>Minor</td>
<td>1</td>
<td>300</td>
<td>Grid</td>
<td>Maintenanc e</td>
</tr>
<tr>
<td>542</td>
<td>Daily Report Failed</td>
<td>1</td>
<td>0</td>
<td>Critical</td>
<td>1</td>
<td>300</td>
<td>Grid</td>
<td>Maintenanc e</td>
</tr>
<tr>
<td>543</td>
<td>Daily Report Mail Sent successfully</td>
<td>1</td>
<td>0</td>
<td>Info</td>
<td>1</td>
<td>300</td>
<td>Grid</td>
<td>Maintenanc e</td>
</tr>
<tr>
<td>544</td>
<td>Daily Report Mail Sent Failed</td>
<td>1</td>
<td>0</td>
<td>Minor</td>
<td>1</td>
<td>300</td>
<td>Grid</td>
<td>Maintenanc e</td>
</tr>
<tr>
<td>545</td>
<td>SM is not responding</td>
<td>1</td>
<td>1</td>
<td>Critical</td>
<td>1</td>
<td>300</td>
<td>Grid</td>
<td>Maintenanc e</td>
</tr>
<tr>
<td>560</td>
<td>User Connected</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Security</td>
</tr>
<tr>
<td>561</td>
<td>User Disconnected</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Security</td>
</tr>
<tr>
<td>602</td>
<td>UFM Server Failover</td>
<td>1</td>
<td>1</td>
<td>Critical</td>
<td>1</td>
<td>0</td>
<td>Site</td>
<td>Fabric Notificatio n</td>
</tr>
<tr>
<td>603</td>
<td>Events Suppression</td>
<td>1</td>
<td>0</td>
<td>Critical</td>
<td>0</td>
<td>300</td>
<td>Site</td>
<td>Maintenanc e</td>
</tr>
<tr>
<td>604</td>
<td>Report Succeeded</td>
<td>1</td>
<td>1</td>
<td>Info</td>
<td>1</td>
<td>300</td>
<td>Grid</td>
<td>Maintenanc e</td>
</tr>
<tr>
<td>605</td>
<td>Report Failed</td>
<td>1</td>
<td>1</td>
<td>Critical</td>
<td>1</td>
<td>300</td>
<td>Grid</td>
<td>Maintenanc e</td>
</tr>
<tr>
<td>606</td>
<td>Correction Attempts Paused</td>
<td>1</td>
<td>0</td>
<td>Warning</td>
<td>1</td>
<td>0</td>
<td>Site</td>
<td>Fabric Notificatio n</td>
</tr>
<tr>
<td>701</td>
<td>Non-optimal Link Speed</td>
<td>1</td>
<td>1</td>
<td>Minor</td>
<td>1</td>
<td>0</td>
<td>Port</td>
<td>Hardware</td>
</tr>
<tr>
<td>702</td>
<td>Unhealthy IB Port</td>
<td>1</td>
<td>1</td>
<td>Warning</td>
<td>1</td>
<td>0</td>
<td>Port</td>
<td>Hardware</td>
</tr>
<tr>
<td>703</td>
<td>Fabric Collector Connected</td>
<td>1</td>
<td>0</td>
<td>Info</td>
<td>1</td>
<td>0</td>
<td>Grid</td>
<td>Maintenanc e</td>
</tr>
<tr>
<td>704</td>
<td>Fabric Collector Disconnected</td>
<td>1</td>
<td>1</td>
<td>Critical</td>
<td>1</td>
<td>0</td>
<td>Grid</td>
<td>Maintenanc e</td>
</tr>
<tr>
<td>750</td>
<td>High data retransmission count on port</td>
<td>1</td>
<td>1</td>
<td>Warning</td>
<td>500</td>
<td>1</td>
<td>Port</td>
<td>Hardware</td>
</tr>
<tr>
<td>901</td>
<td>Fabric Configuration Started</td>
<td>0</td>
<td>1</td>
<td>Info</td>
<td>1</td>
<td>0</td>
<td>Grid</td>
<td>Fabric Notificatio n</td>
</tr>
<tr>
<td>902</td>
<td>Fabric Configuration Completed</td>
<td>0</td>
<td>1</td>
<td>Info</td>
<td>1</td>
<td>0</td>
<td>Grid</td>
<td>Fabric Notificatio n</td>
</tr>
<tr>
<td>903</td>
<td>Fabric Configuration Failed</td>
<td>0</td>
<td>1</td>
<td>Critical</td>
<td>1</td>
<td>0</td>
<td>Grid</td>
<td>Fabric Notificatio n</td>
</tr>
<tr>
<td>Event ID</td>
<td>Event Name</td>
<td>To Log</td>
<td>Alarm</td>
<td>Default Severity</td>
<td>Default Threshold</td>
<td>Related Object</td>
<td>Category</td>
<td>Source</td>
</tr>
<tr>
<td>----------</td>
<td>------------------------------------------------</td>
<td>--------</td>
<td>-------</td>
<td>------------------</td>
<td>-------------------</td>
<td>----------------</td>
<td>---------------</td>
<td>--------</td>
</tr>
<tr>
<td>904</td>
<td>Device Configuration Failure</td>
<td>0</td>
<td>1</td>
<td>Critical</td>
<td>1</td>
<td>0</td>
<td>Device</td>
<td>Fabric Notification</td>
</tr>
<tr>
<td>905</td>
<td>Device Configuration Timeout</td>
<td>0</td>
<td>1</td>
<td>Critical</td>
<td>1</td>
<td>0</td>
<td>Device</td>
<td>Fabric Notification</td>
</tr>
<tr>
<td>906</td>
<td>Provisioning Validation Failure</td>
<td>0</td>
<td>1</td>
<td>Critical</td>
<td>1</td>
<td>0</td>
<td>Grid</td>
<td>Fabric Notification</td>
</tr>
<tr>
<td>907</td>
<td>Switch is Down</td>
<td>1</td>
<td>1</td>
<td>Critical</td>
<td>1</td>
<td>0</td>
<td>Site</td>
<td>Fabric Topology</td>
</tr>
<tr>
<td>908</td>
<td>Switch is Up</td>
<td>1</td>
<td>1</td>
<td>Info</td>
<td>1</td>
<td>300</td>
<td>Site</td>
<td>Fabric Topology</td>
</tr>
<tr>
<td>909</td>
<td>Director Switch is Down</td>
<td>1</td>
<td>1</td>
<td>Critical</td>
<td>1</td>
<td>300</td>
<td>Site</td>
<td>Fabric Topology</td>
</tr>
<tr>
<td>910</td>
<td>Director Switch is Up</td>
<td>1</td>
<td>1</td>
<td>Info</td>
<td>1</td>
<td>0</td>
<td>Site</td>
<td>Fabric Topology</td>
</tr>
<tr>
<td>911</td>
<td>Module Temperature Low Threshold Reached</td>
<td>1</td>
<td>1</td>
<td>Warning</td>
<td>60</td>
<td>300</td>
<td>Module</td>
<td>Hardware</td>
</tr>
<tr>
<td>912</td>
<td>Module Temperature High Threshold Reached</td>
<td>1</td>
<td>1</td>
<td>Critical</td>
<td>60</td>
<td>300</td>
<td>Module</td>
<td>Hardware</td>
</tr>
<tr>
<td>913</td>
<td>Module High Voltage</td>
<td>1</td>
<td>1</td>
<td>Warning</td>
<td>10</td>
<td>420</td>
<td>Switch</td>
<td>Module Status</td>
</tr>
<tr>
<td>914</td>
<td>Module High Current</td>
<td>1</td>
<td>1</td>
<td>Warning</td>
<td>10</td>
<td>420</td>
<td>Switch</td>
<td>Module Status</td>
</tr>
<tr>
<td>915</td>
<td>BER_ERROR</td>
<td>1</td>
<td>1</td>
<td>Critical</td>
<td>1e-8</td>
<td>420</td>
<td>Port</td>
<td>Hardware</td>
</tr>
<tr>
<td>916</td>
<td>BER_WARNING</td>
<td>1</td>
<td>1</td>
<td>Warning</td>
<td>1e-13</td>
<td>420</td>
<td>Port</td>
<td>Hardware</td>
</tr>
<tr>
<td>917</td>
<td>SYMBOL_BER_ERROR</td>
<td>1</td>
<td>1</td>
<td>Critical</td>
<td>10</td>
<td>420</td>
<td>Port</td>
<td>Hardware</td>
</tr>
<tr>
<td>918</td>
<td>High Symbol BER reported</td>
<td>1</td>
<td>1</td>
<td>Warning</td>
<td>10</td>
<td>420</td>
<td>Port</td>
<td>Hardware</td>
</tr>
<tr>
<td>919</td>
<td>Cable Temperature High</td>
<td>1</td>
<td>1</td>
<td>Critical</td>
<td>0</td>
<td>0</td>
<td>Port</td>
<td>Hardware</td>
</tr>
<tr>
<td>920</td>
<td>Cable Temperature Low</td>
<td>1</td>
<td>1</td>
<td>Critical</td>
<td>0</td>
<td>0</td>
<td>Port</td>
<td>Hardware</td>
</tr>
<tr>
<td>1300</td>
<td>SM_SAKEY_VIOLATION</td>
<td>1</td>
<td>1</td>
<td>Warning</td>
<td>5300</td>
<td></td>
<td>Port</td>
<td>Security</td>
</tr>
<tr>
<td>1301</td>
<td>SM_SGID_SPOOFED</td>
<td>1</td>
<td>1</td>
<td>Warning</td>
<td>5300</td>
<td></td>
<td>Port</td>
<td>Security</td>
</tr>
<tr>
<td>1302</td>
<td>SM_RATE_LIMIT_EXCEEDED</td>
<td>1</td>
<td>1</td>
<td>Warning</td>
<td>5300</td>
<td></td>
<td>Port</td>
<td>Security</td>
</tr>
<tr>
<td>Event ID</td>
<td>Event Name</td>
<td>To Log</td>
<td>Alarm</td>
<td>Default Severity</td>
<td>Default Threshold</td>
<td>Related Object</td>
<td>Category</td>
<td>Source</td>
</tr>
<tr>
<td>----------</td>
<td>------------------------------------------------</td>
<td>--------</td>
<td>-------</td>
<td>------------------</td>
<td>-------------------</td>
<td>----------------</td>
<td>----------</td>
<td>--------</td>
</tr>
<tr>
<td>1303</td>
<td>SM_MULTICAST_GROUPS_LIMIT_EXCEEDED</td>
<td>1</td>
<td>1</td>
<td>Warning</td>
<td>5300</td>
<td>Port</td>
<td>Security</td>
<td>SM</td>
</tr>
<tr>
<td>1304</td>
<td>SM_SERVICES_LIMIT_EXCEEDED</td>
<td>1</td>
<td>1</td>
<td>Warning</td>
<td>5300</td>
<td>Port</td>
<td>Security</td>
<td>SM</td>
</tr>
<tr>
<td>1305</td>
<td>SM_EVENT_SUBSCRIPTION_LIMIT_EXCEEDED</td>
<td>1</td>
<td>1</td>
<td>Warning</td>
<td>5300</td>
<td>Port</td>
<td>Security</td>
<td>SM</td>
</tr>
<tr>
<td>1306</td>
<td>Unallowed SM was detected in the fabric</td>
<td>1</td>
<td>1</td>
<td>Warning</td>
<td>0</td>
<td>Port</td>
<td>Fabric Notification</td>
<td>SM</td>
</tr>
<tr>
<td>1307</td>
<td>SMinfo SET request was received from unallowed SM</td>
<td>1</td>
<td>1</td>
<td>Warning</td>
<td>0</td>
<td>Port</td>
<td>Fabric Notification</td>
<td>SM</td>
</tr>
<tr>
<td>1309</td>
<td>SM was detected with non-matching SMKey</td>
<td>1</td>
<td>1</td>
<td>Warning</td>
<td>0</td>
<td>Port</td>
<td>Fabric Notification</td>
<td>SM</td>
</tr>
<tr>
<td>1310</td>
<td>Duplicated node GUID was detected</td>
<td>1</td>
<td>1</td>
<td>Critical</td>
<td>1</td>
<td>Device</td>
<td>Fabric Notification</td>
<td>SM</td>
</tr>
<tr>
<td>1311</td>
<td>Duplicated port GUID was detected</td>
<td>1</td>
<td>1</td>
<td>Critical</td>
<td>1</td>
<td>Port</td>
<td>Fabric Notification</td>
<td>SM</td>
</tr>
<tr>
<td>1312</td>
<td>Switch was Rebooted</td>
<td>1</td>
<td>1</td>
<td>Info</td>
<td>1</td>
<td>Device</td>
<td>Fabric Notification</td>
<td>UFM</td>
</tr>
<tr>
<td>1315</td>
<td>Topo Config File Error</td>
<td>1</td>
<td>1</td>
<td>Critical</td>
<td>1</td>
<td>Grid</td>
<td>Fabric Notification</td>
<td>UFM</td>
</tr>
<tr>
<td>1316</td>
<td>Topo Config Subnet Mismatch</td>
<td>1</td>
<td>1</td>
<td>Critical</td>
<td>1</td>
<td>Grid</td>
<td>Fabric Notification</td>
<td>Topodiff</td>
</tr>
<tr>
<td>1400</td>
<td>High Ambient Temperature</td>
<td>1</td>
<td>1</td>
<td>Warning</td>
<td>0</td>
<td>Switch</td>
<td>Hardware</td>
<td>Switch</td>
</tr>
<tr>
<td>1401</td>
<td>High Fluid Temperature</td>
<td>1</td>
<td>1</td>
<td>Warning</td>
<td>0</td>
<td>Switch</td>
<td>Hardware</td>
<td>Switch</td>
</tr>
<tr>
<td>1402</td>
<td>Low Fluid Level</td>
<td>1</td>
<td>1</td>
<td>Warning</td>
<td>0</td>
<td>Switch</td>
<td>Hardware</td>
<td>Switch</td>
</tr>
<tr>
<td>1403</td>
<td>Low Supply Pressure</td>
<td>1</td>
<td>1</td>
<td>Warning</td>
<td>0</td>
<td>Switch</td>
<td>Hardware</td>
<td>Switch</td>
</tr>
<tr>
<td>1404</td>
<td>High Supply Pressure</td>
<td>1</td>
<td>1</td>
<td>Warning</td>
<td>0</td>
<td>Switch</td>
<td>Hardware</td>
<td>Switch</td>
</tr>
<tr>
<td>1405</td>
<td>Low Return Pressure</td>
<td>1</td>
<td>1</td>
<td>Warning</td>
<td>0</td>
<td>Switch</td>
<td>Hardware</td>
<td>Switch</td>
</tr>
<tr>
<td>1406</td>
<td>High Return Pressure</td>
<td>1</td>
<td>1</td>
<td>Warning</td>
<td>0</td>
<td>Switch</td>
<td>Hardware</td>
<td>Switch</td>
</tr>
<tr>
<td>Event ID</td>
<td>Event Name</td>
<td>T o L o g</td>
<td>Alar m</td>
<td>Defau lt Seve rity</td>
<td>Defau lt Thres hold</td>
<td>Defau lt TTL</td>
<td>Related Obje ct</td>
<td>Category</td>
</tr>
<tr>
<td>----------</td>
<td>-------------------------------------------</td>
<td>----------</td>
<td>--------</td>
<td>------------------</td>
<td>---------------------</td>
<td>-------------</td>
<td>----------------</td>
<td>-------------</td>
</tr>
<tr>
<td>1407</td>
<td>High Differential Pressure</td>
<td>1</td>
<td>1</td>
<td>Warning</td>
<td>0</td>
<td>86400</td>
<td>Switch</td>
<td>Hardware</td>
</tr>
<tr>
<td>1408</td>
<td>Low Differential Pressure</td>
<td>1</td>
<td>1</td>
<td>Warning</td>
<td>0</td>
<td>86400</td>
<td>Switch</td>
<td>Hardware</td>
</tr>
<tr>
<td>1409</td>
<td>System Fail Safe</td>
<td>1</td>
<td>1</td>
<td>Warning</td>
<td>0</td>
<td>86400</td>
<td>Switch</td>
<td>Hardware</td>
</tr>
<tr>
<td>1410</td>
<td>Fault Critical</td>
<td>1</td>
<td>1</td>
<td>Critical</td>
<td>0</td>
<td>86400</td>
<td>Switch</td>
<td>Hardware</td>
</tr>
<tr>
<td>1411</td>
<td>Fault Pump1</td>
<td>1</td>
<td>1</td>
<td>Critical</td>
<td>0</td>
<td>86400</td>
<td>Switch</td>
<td>Hardware</td>
</tr>
<tr>
<td>1412</td>
<td>Fault Pump2</td>
<td>1</td>
<td>1</td>
<td>Critical</td>
<td>0</td>
<td>86400</td>
<td>Switch</td>
<td>Hardware</td>
</tr>
<tr>
<td>1413</td>
<td>Fault Fluid Level Critical</td>
<td>1</td>
<td>1</td>
<td>Critical</td>
<td>0</td>
<td>86400</td>
<td>Switch</td>
<td>Hardware</td>
</tr>
<tr>
<td>1414</td>
<td>Fault Fluid Over Temperature</td>
<td>1</td>
<td>1</td>
<td>Critical</td>
<td>0</td>
<td>86400</td>
<td>Switch</td>
<td>Hardware</td>
</tr>
<tr>
<td>1415</td>
<td>Fault Primary DC</td>
<td>1</td>
<td>1</td>
<td>Critical</td>
<td>0</td>
<td>86400</td>
<td>Switch</td>
<td>Hardware</td>
</tr>
<tr>
<td>1416</td>
<td>Fault Redundant DC</td>
<td>1</td>
<td>1</td>
<td>Critical</td>
<td>0</td>
<td>86400</td>
<td>Switch</td>
<td>Hardware</td>
</tr>
<tr>
<td>1417</td>
<td>Fault Fluid Leak</td>
<td>1</td>
<td>1</td>
<td>Critical</td>
<td>0</td>
<td>86400</td>
<td>Switch</td>
<td>Hardware</td>
</tr>
<tr>
<td>1418</td>
<td>Fault Sensor Failure</td>
<td>1</td>
<td>1</td>
<td>Critical</td>
<td>0</td>
<td>86400</td>
<td>Switch</td>
<td>Hardware</td>
</tr>
<tr>
<td>1419</td>
<td>Cooling Device Monitoring Error</td>
<td>1</td>
<td>0</td>
<td>Critical</td>
<td>0</td>
<td>1</td>
<td>Grid</td>
<td>Hardware</td>
</tr>
<tr>
<td>1420</td>
<td>Cooling Device Communication Error</td>
<td>1</td>
<td>1</td>
<td>Critical</td>
<td>0</td>
<td>86400</td>
<td>Switch</td>
<td>Hardware</td>
</tr>
<tr>
<td>1500</td>
<td>New cable detected</td>
<td>1</td>
<td>0</td>
<td>Info</td>
<td>1</td>
<td>0</td>
<td>Link</td>
<td>Security</td>
</tr>
<tr>
<td>1502</td>
<td>Cable detected in a new location</td>
<td>1</td>
<td>0</td>
<td>Warning</td>
<td>1</td>
<td>0</td>
<td>Link</td>
<td>Security</td>
</tr>
<tr>
<td>1503</td>
<td>Duplicate Cable Detected</td>
<td>1</td>
<td>0</td>
<td>Critical</td>
<td>1</td>
<td>0</td>
<td>Link</td>
<td>Security</td>
</tr>
<tr>
<td>1315</td>
<td>Topo Config File Error</td>
<td>1</td>
<td>1</td>
<td>Critical</td>
<td>1</td>
<td>0</td>
<td>Grid</td>
<td>Fabric Notificatio n</td>
</tr>
<tr>
<td>1504</td>
<td>SHARP Allocation Succeeded</td>
<td>1</td>
<td>1</td>
<td>Info</td>
<td>1</td>
<td>0</td>
<td>Grid</td>
<td>SHARP</td>
</tr>
<tr>
<td>1505</td>
<td>SHARP Allocation Failed</td>
<td>1</td>
<td>0</td>
<td>Warning</td>
<td>1</td>
<td>0</td>
<td>Grid</td>
<td>SHARP</td>
</tr>
<tr>
<td>1506</td>
<td>SHARP Deallocation Succeeded</td>
<td>1</td>
<td>0</td>
<td>Info</td>
<td>1</td>
<td>0</td>
<td>Grid</td>
<td>SHARP</td>
</tr>
<tr>
<td>1507</td>
<td>SHARP Deallocation Failed</td>
<td>1</td>
<td>0</td>
<td>Warning</td>
<td>1</td>
<td>0</td>
<td>Grid</td>
<td>SHARP</td>
</tr>
<tr>
<td>Event ID</td>
<td>Event Name</td>
<td>Tolo</td>
<td>Alarm</td>
<td>Default Severity</td>
<td>Default Threshold</td>
<td>Relat Objct</td>
<td>Category</td>
<td>Source</td>
</tr>
<tr>
<td>---------</td>
<td>-----------------------------------------------</td>
<td>------</td>
<td>-------</td>
<td>------------------</td>
<td>------------------</td>
<td>------------</td>
<td>----------</td>
<td>--------</td>
</tr>
<tr>
<td>1508</td>
<td>Device Collect System Dump Started</td>
<td>1</td>
<td>0</td>
<td>Info</td>
<td>1</td>
<td>300</td>
<td>Device</td>
<td>Maintena</td>
</tr>
<tr>
<td>1509</td>
<td>Device Collect System Dump Finished</td>
<td>1</td>
<td>0</td>
<td>Info</td>
<td>1</td>
<td>300</td>
<td>Device</td>
<td>Maintena</td>
</tr>
<tr>
<td>1510</td>
<td>Device Collect System Dump Error</td>
<td>1</td>
<td>0</td>
<td>Critical</td>
<td>1</td>
<td>300</td>
<td>Device</td>
<td>Maintena</td>
</tr>
<tr>
<td>1511</td>
<td>Virtual Port Added</td>
<td>1</td>
<td>0</td>
<td>Info</td>
<td>1</td>
<td>0</td>
<td>Port</td>
<td>Fabric Notificatio</td>
</tr>
<tr>
<td>1512</td>
<td>Virtual Port Removed</td>
<td>1</td>
<td>0</td>
<td>Warning</td>
<td>1</td>
<td>0</td>
<td>Port</td>
<td>Fabric Notificatio</td>
</tr>
<tr>
<td>1513</td>
<td>Burn Cables Transceivers Started</td>
<td>1</td>
<td>0</td>
<td>Info</td>
<td>1</td>
<td>0</td>
<td>Device</td>
<td>Maintena</td>
</tr>
<tr>
<td>1514</td>
<td>Burn Cables Transceivers Finished</td>
<td>1</td>
<td>0</td>
<td>Info</td>
<td>1</td>
<td>0</td>
<td>Device</td>
<td>Maintena</td>
</tr>
<tr>
<td>1515</td>
<td>Burn Cables Transceivers Failed</td>
<td>1</td>
<td>0</td>
<td>Warning</td>
<td>1</td>
<td>0</td>
<td>Device</td>
<td>Maintena</td>
</tr>
<tr>
<td>1516</td>
<td>Activate Cables Transceivers FW Started</td>
<td>1</td>
<td>0</td>
<td>Info</td>
<td>1</td>
<td>0</td>
<td>Device</td>
<td>Maintena</td>
</tr>
<tr>
<td>1517</td>
<td>Activate Cables Transceivers FW Failed</td>
<td>1</td>
<td>0</td>
<td>Warning</td>
<td>1</td>
<td>0</td>
<td>Device</td>
<td>Maintena</td>
</tr>
<tr>
<td>1520</td>
<td>Aggregation Node Discovery Failed</td>
<td>1</td>
<td>0</td>
<td>Critical</td>
<td>1</td>
<td>0</td>
<td>SHARP AM</td>
<td>SHARP</td>
</tr>
<tr>
<td>1521</td>
<td>Job Started</td>
<td>1</td>
<td>0</td>
<td>Info</td>
<td>1</td>
<td>0</td>
<td>SHARP AM</td>
<td>SHARP</td>
</tr>
<tr>
<td>1522</td>
<td>Job Ended</td>
<td>1</td>
<td>0</td>
<td>Info</td>
<td>1</td>
<td>0</td>
<td>SHARP AM</td>
<td>SHARP</td>
</tr>
<tr>
<td>1523</td>
<td>Job Start Failed</td>
<td>1</td>
<td>0</td>
<td>Critical</td>
<td>1</td>
<td>0</td>
<td>SHARP AM</td>
<td>SHARP</td>
</tr>
<tr>
<td>1524</td>
<td>Job Error</td>
<td>1</td>
<td>0</td>
<td>Critical</td>
<td>1</td>
<td>0</td>
<td>SHARP AM</td>
<td>SHARP</td>
</tr>
<tr>
<td>1525</td>
<td>Trap QP Error</td>
<td>1</td>
<td>0</td>
<td>Critical</td>
<td>1</td>
<td>0</td>
<td>SHARP AM</td>
<td>SHARP</td>
</tr>
<tr>
<td>1526</td>
<td>Trap Invalid Request</td>
<td>1</td>
<td>0</td>
<td>Critical</td>
<td>1</td>
<td>0</td>
<td>SHARP AM</td>
<td>SHARP</td>
</tr>
<tr>
<td>1527</td>
<td>Trap Sharp Error</td>
<td>1</td>
<td>0</td>
<td>Critical</td>
<td>1</td>
<td>0</td>
<td>SHARP AM</td>
<td>SHARP</td>
</tr>
<tr>
<td>1528</td>
<td>Trap QP Alloc timeout</td>
<td>1</td>
<td>0</td>
<td>Critical</td>
<td>1</td>
<td>0</td>
<td>SHARP AM</td>
<td>SHARP</td>
</tr>
<tr>
<td>1529</td>
<td>Trap AMKey Violation</td>
<td>1</td>
<td>0</td>
<td>Critical</td>
<td>1</td>
<td>0</td>
<td>SHARP AM</td>
<td>SHARP</td>
</tr>
<tr>
<td>Event ID</td>
<td>Event Name</td>
<td>Alarm</td>
<td>Default Severity</td>
<td>Default Threshold</td>
<td>Related Object</td>
<td>Category</td>
<td>Source</td>
<td></td>
</tr>
<tr>
<td>----------</td>
<td>------------</td>
<td>-------</td>
<td>------------------</td>
<td>-------------------</td>
<td>----------------</td>
<td>----------</td>
<td>--------</td>
<td></td>
</tr>
<tr>
<td>1530</td>
<td>Unsupported Trap</td>
<td>1</td>
<td>1</td>
<td>SHARP</td>
<td>SHARP</td>
<td>SHARP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1531</td>
<td>Reservation Updated</td>
<td>1</td>
<td>0</td>
<td>Info</td>
<td>SHARP</td>
<td>SHARP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1532</td>
<td>Sharp is not Responding</td>
<td>1</td>
<td>1</td>
<td>Critical</td>
<td>SHARP</td>
<td>SHARP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1533</td>
<td>Agg Node Active</td>
<td>1</td>
<td>0</td>
<td>Info</td>
<td>SHARP</td>
<td>SHARP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1534</td>
<td>Agg Node Inactive</td>
<td>1</td>
<td>0</td>
<td>Warning</td>
<td>SHARP</td>
<td>SHARP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1535</td>
<td>Trap AMKey Violation Triggered by AM</td>
<td>1</td>
<td>0</td>
<td>Warning</td>
<td>SHARP</td>
<td>SHARP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1550</td>
<td>Guids Were Added to Pkey</td>
<td>1</td>
<td>0</td>
<td>Info</td>
<td>Port</td>
<td>Fabric Notification</td>
<td>UFM</td>
<td></td>
</tr>
<tr>
<td>1551</td>
<td>Guids Were Removed from Pkey</td>
<td>1</td>
<td>0</td>
<td>Info</td>
<td>Port</td>
<td>Fabric Notification</td>
<td>UFM</td>
<td></td>
</tr>
<tr>
<td>1600</td>
<td>VS/CC Classes Key Violation</td>
<td>1</td>
<td>1</td>
<td>Warning</td>
<td>Security</td>
<td>SM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1602</td>
<td>PCI Speed Degradation Warning</td>
<td>1</td>
<td>1</td>
<td>Warning</td>
<td>Port</td>
<td>Fabric Notification</td>
<td>UFM</td>
<td></td>
</tr>
<tr>
<td>1603</td>
<td>PCI Width Degradation Warning</td>
<td>1</td>
<td>1</td>
<td>Warning</td>
<td>Port</td>
<td>Fabric Notification</td>
<td>UFM</td>
<td></td>
</tr>
</tbody>
</table>
### 7 Reports

UFM reports provide summarized information about selected topics. Through the different UFM WebUI tabs, you can create reports that run a series of checks on UFM components.

The table below summarizes the types of reports and provides useful links for more information.

<table>
<thead>
<tr>
<th>Report Type</th>
<th>Description</th>
<th>WebUI</th>
<th>REST API</th>
</tr>
</thead>
<tbody>
<tr>
<td>UFM Health Report</td>
<td>A report that run a series of checks related to UFM server health, including CPU, memory, license, configurations and disk monitoring.</td>
<td>UFM Health Tab</td>
<td>Reports REST API</td>
</tr>
</tbody>
</table>
| Fabric Health Report      | Through the Fabric Health tab, you can access the fabric health reports. There are two kinds of reports:  
  • Custom Reports - The user can generate a report that runs a series of checks on the fabric on demand.  
  • Periodic Reports - An automatically generated report that is periodically generated by the UFM. | Fabric Health Tab | Reports REST API Periodic Fabric Health REST API |
| Topology Comparison Report| Reports on topology comparison, available as:  
  • Periodic Comparison - allows users to compare the current fabric topology with a preset master topology.  
  • Custom Comparison - compares user-defined topology with the current fabric topology. | Topology Compare Tab | Topology Compare REST API       |
| Daily Reports             | The Daily Report feature collects, analyzes, and reports the most significant issues of the fabric in the last 24 hours | Daily Reports Tab | N/A                           |
# Telemetry

UFM Telemetry allows the collection and monitoring of InfiniBand fabric port statistics, such as network bandwidth, congestion, errors, latency, and more.

UFM provides a range of telemetry capabilities:

- Real-time monitoring views
- Monitoring of multiple attributes
- Intelligent Counters for error and congestion counters
- InfiniBand port-based error counters
- InfiniBand congestion XmitWait counter-based congestion measurement
- InfiniBand port-based bandwidth data

The telemetry session panels support the following actions:

- Rearrangement via a straightforward drag-and-drop function
- Resizing by hovering over the panel's border

UFM Telemetry data is collected via UFM telemetry instances invoked during UFM startup.

<table>
<thead>
<tr>
<th>Telemetry Instance</th>
<th>Description</th>
<th>REST API</th>
</tr>
</thead>
</table>
| **High-Frequency (Primary) Telemetry Instance** | A default telemetry session that collects a predefined set of ~30 counters covering bandwidth, congestion, and error metrics, which UFM analyzes and reports. These counters are used for:  
  - Default Telemetry Session - An ongoing session used by the UFM to display UFM WebUI dashboard charts information and for monitoring and analyzing ports threshold events (the session interval is 30 secs by default)  
  - Real-Time Telemetry - allows users to define live telemetry sessions for monitoring small subsets of devices or ports and a selected set of counters. For more information, refer to [Telemetry - User-Defined Sessions](#)  
  - Historical Telemetry - based on the primary telemetry and collects statistical data from all fabric ports and stores them in an internal UFM SQLite database (the session interval is 5 mins by default) | For Default and Real-time Telemetry: [Monitoring REST API](#)  
For Historical Telemetry: [History Telemetry Sessions REST API](#) |
| **Low-Frequency (Secondary) Telemetry Instance** | Operates automatically upon UFM startup, offering an extended scope of 120 counters. For a list of the Secondary Telemetry Fields, refer to [Low-Frequency (Secondary) Telemetry Fields](#) | N/A |

For direct telemetry endpoint access, which exposes the list of supported counters:

For the High-Frequency (Primary) Telemetry Instance, run the following command:
For the Low-Frequency (Secondary) Telemetry Instance, run the following command:

curl http://r-ufm114:9002/csv/cset/low_freq_debug

8.1 Historical Telemetry Collection in UFM

8.1.1 Storage Considerations

UFM periodically collects fabric port statistics and saves them in its SQLite database. Before starting up UFM Enterprise, please consider the following disk space utilization for various fabric sizes and duration.

The measurements in the table below were taken with sampling interval set to once per 30 seconds. Be aware that the default sampling rate is once per 300 seconds. Disk utilization calculation should be adjusted accordingly.

<table>
<thead>
<tr>
<th>Number of Nodes</th>
<th>Ports per Node</th>
<th>Storage per Hour</th>
<th>Storage per 15 Days</th>
<th>Storage per 30 Days</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>8</td>
<td>1.6 MB</td>
<td>576 MB (0.563 GB)</td>
<td>1152 MB (1.125 GB)</td>
</tr>
<tr>
<td>100</td>
<td>8</td>
<td>11 MB</td>
<td>3960 MB (3.867 GB)</td>
<td>7920 MB (7.734 GB)</td>
</tr>
<tr>
<td>500</td>
<td>8</td>
<td>50 MB</td>
<td>18000 MB (17.58 GB)</td>
<td>36000 MB (35.16 GB)</td>
</tr>
<tr>
<td>1000</td>
<td>8</td>
<td>100 MB</td>
<td>36000 MB (35.16 GB)</td>
<td>72000 MB (70.31 GB)</td>
</tr>
</tbody>
</table>

8.2 High-Frequency (Primary) Telemetry Fields

The following is a list of available counters which includes a variety of metrics related to timestamps, port and node information, error statistics, firmware versions, temperatures, cable details, power levels, and various other telemetry-related data.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>timestamp</td>
<td></td>
</tr>
<tr>
<td>source_id</td>
<td></td>
</tr>
<tr>
<td>tag</td>
<td></td>
</tr>
<tr>
<td>node_guid</td>
<td>node GUID</td>
</tr>
<tr>
<td>port_guid</td>
<td>Port GUID</td>
</tr>
<tr>
<td>port_num</td>
<td>Port Number</td>
</tr>
<tr>
<td>PortXmitDataExtended</td>
<td>Transmitted data rate per egress port in bytes passing through the port during the sample period</td>
</tr>
<tr>
<td>PortRcvDataExtended</td>
<td>The received data on the ingress port in bytes during the sample period</td>
</tr>
<tr>
<td>Field Name</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>PortXmitPktsExtended</td>
<td>Total number of packets transmitted on the port.</td>
</tr>
<tr>
<td>PortRcvPktsExtended</td>
<td>Total number of packets received on the port.</td>
</tr>
<tr>
<td>SymbolErrorCounterExtended</td>
<td>This counter provides information on error bits that were not corrected by phy correction mechanisms.</td>
</tr>
<tr>
<td>LinkErrorRecoveryCounterExtended</td>
<td>Total number of times the Port Training state machine has successfully completed the link error recovery process.</td>
</tr>
<tr>
<td>LinkDownedCounterExtended</td>
<td>Perf.PortCounters</td>
</tr>
<tr>
<td>PortRcvErrorsExtended</td>
<td>Total number of packets containing an error that were received on the port.</td>
</tr>
<tr>
<td>PortRcvRemotePhysicalErrorsExtended</td>
<td>Total number of packets marked with the EBP delimiter received on the port.</td>
</tr>
<tr>
<td>PortRcvSwitchRelayErrorsExtended</td>
<td>Total number of packets received on the port that were discarded because they could not be forwarded by the switch relay.</td>
</tr>
<tr>
<td>PortXmitDiscardsExtended</td>
<td>Total number of outbound packets discarded by the port because the port is down or congested.</td>
</tr>
<tr>
<td>PortXmitConstraintErrorsExtended</td>
<td>Total number of packets not transmitted from the switch physical port.</td>
</tr>
<tr>
<td>PortRcvConstraintErrorsExtended</td>
<td>Total number of packets received on the switch physical port that are discarded.</td>
</tr>
<tr>
<td>LocalLinkIntegrityErrorsExtended</td>
<td>The number of times that the count of local physical errors exceeded the threshold specified by LocalPhyErrors</td>
</tr>
<tr>
<td>ExcessiveBufferOverrunErrorsExtended</td>
<td>The number of times that OverrunErrors consecutive flow control update periods occurred, each having at least one overrun error</td>
</tr>
<tr>
<td>VL15DroppedExtended</td>
<td>Number of incoming VL15 packets dropped due to resource limitations (e.g., lack of buffers) in the port</td>
</tr>
<tr>
<td>PortXmitWaitExtended</td>
<td>The time an egress port had data to send but could not send it due to lack of credits or arbitration - in time ticks within the sample-time window</td>
</tr>
<tr>
<td>hist[0-4]</td>
<td>Hist[i] give the number of FEC blocks that had RS-FEC symbols errors of value i or range of errors</td>
</tr>
<tr>
<td>infiniband_CBW</td>
<td></td>
</tr>
<tr>
<td>Normalized_CBW</td>
<td></td>
</tr>
<tr>
<td>NormalizedXW</td>
<td></td>
</tr>
<tr>
<td>Normalized_XmitData</td>
<td></td>
</tr>
</tbody>
</table>

The following is a list of available counters which includes a variety of metrics related to timestamps, port and node information, error statistics, firmware versions, temperatures, cable details, power levels, and various other telemetry-related data.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>timestamp</td>
<td></td>
</tr>
<tr>
<td>Field Name</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>source_id</td>
<td></td>
</tr>
<tr>
<td>tag</td>
<td></td>
</tr>
<tr>
<td>node_guid</td>
<td>node GUID</td>
</tr>
<tr>
<td>port_guid</td>
<td>Port GUID</td>
</tr>
<tr>
<td>port_num</td>
<td>Port Number</td>
</tr>
<tr>
<td>PortXmitDataExtended</td>
<td>Transmitted data rate per egress port in bytes passing through the port during the sample period</td>
</tr>
<tr>
<td>PortRcvDataExtended</td>
<td>The received data on the ingress port in bytes during the sample period</td>
</tr>
<tr>
<td>PortXmitPktsExtended</td>
<td>Total number of packets transmitted on the port</td>
</tr>
<tr>
<td>PortRcvPktsExtended</td>
<td>Total number of packets received on the port</td>
</tr>
<tr>
<td>SymbolErrorCounterExtended</td>
<td></td>
</tr>
<tr>
<td>LinkErrorRecoveryCounterExtended</td>
<td></td>
</tr>
<tr>
<td>LinkDownedCounterExtended</td>
<td></td>
</tr>
<tr>
<td>PortRcvErrorsExtended</td>
<td></td>
</tr>
<tr>
<td>PortRcvRemotePhysicalErrorsExtended</td>
<td></td>
</tr>
<tr>
<td>PortRcvSwitchRelayErrorsExtended</td>
<td></td>
</tr>
<tr>
<td>PortXmitDiscardsExtended</td>
<td></td>
</tr>
<tr>
<td>PortXmitConstraintErrorsExtended</td>
<td></td>
</tr>
<tr>
<td>LocalLinkIntegrityErrorsExtended</td>
<td></td>
</tr>
<tr>
<td>ExcessiveBufferOverrunErrorsExtended</td>
<td></td>
</tr>
<tr>
<td>VL15DroppedExtended</td>
<td></td>
</tr>
<tr>
<td>PortXmitWaitExtended</td>
<td></td>
</tr>
<tr>
<td>hist[0-4]</td>
<td>Hist[i] give the number of FEC blocks that had RS-FEC symbols errors of value i or range of errors</td>
</tr>
<tr>
<td>infiniband_CBW</td>
<td></td>
</tr>
<tr>
<td>Normalized_CBW</td>
<td></td>
</tr>
<tr>
<td>NormalizedXW</td>
<td></td>
</tr>
<tr>
<td>Normalized_XmitData</td>
<td></td>
</tr>
</tbody>
</table>

**8.3 Low-Frequency (Secondary) Telemetry Fields**

The following is a list of available counters which includes a variety of metrics related to timestamps, port and node information, error statistics, firmware versions, temperatures, cable details, power levels, and various other telemetry-related data.
<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Node_GUID</td>
<td>node GUID</td>
</tr>
<tr>
<td>Device_ID</td>
<td>PCI device ID</td>
</tr>
<tr>
<td>node_description</td>
<td>node description</td>
</tr>
<tr>
<td>lid</td>
<td></td>
</tr>
<tr>
<td>Port_Number</td>
<td>port number</td>
</tr>
<tr>
<td>port_label</td>
<td>port label</td>
</tr>
<tr>
<td>Phy_Manager_State</td>
<td>FW Phy Manager FSM state</td>
</tr>
<tr>
<td>phy_state</td>
<td>physical state</td>
</tr>
<tr>
<td>logical_state</td>
<td>Port Logical link state</td>
</tr>
<tr>
<td>Link_speed_active</td>
<td>ib link active speed</td>
</tr>
<tr>
<td>Link_width_active</td>
<td>ib link active widthsource_id</td>
</tr>
<tr>
<td>Active_FEC</td>
<td>Active FEC</td>
</tr>
<tr>
<td>Total_Raw_BER</td>
<td>Pre-FEC monitor parameters</td>
</tr>
<tr>
<td>Effective_BER</td>
<td>Post FEC monitor parameters</td>
</tr>
<tr>
<td>Symbol_BER</td>
<td>BER after all phy correction mechanism: post FEC + PLR monitor parameters</td>
</tr>
<tr>
<td>Raw_Errors_Lane_[0-3]</td>
<td>This counter provides information on error bits that were identified on lane X. When FEC is enabled this induction corresponds to corrected errors. In PRBS test mode, indicates the number of PRBS errors on lane X.</td>
</tr>
<tr>
<td>Effective_Errors</td>
<td>This counter provides information on error bits that were not corrected by FEC correction algorithm or that FEC is not active.</td>
</tr>
<tr>
<td>Symbol_Errors</td>
<td>This counter provides information on error bits that were not corrected by phy correction mechanisms.</td>
</tr>
<tr>
<td>Time_since_last_clear_Min</td>
<td>The time passed since the last counters clear event in msec. (physical layer statistical counters)</td>
</tr>
<tr>
<td>hist[0-15]</td>
<td>Hist[i] give the number of FEC blocks that had RS-FEC symbols errors of value i or range of errors</td>
</tr>
<tr>
<td>FW_Version</td>
<td>Node FW version</td>
</tr>
<tr>
<td>Chip_Temp</td>
<td>switch temperature</td>
</tr>
<tr>
<td>Link_Down</td>
<td>Perf.PortCounters(LinkDownedCounter)</td>
</tr>
<tr>
<td>Link_Down_IB</td>
<td>Total number of times the Port Training state machine has failed the link error recovery process and downed the link.</td>
</tr>
<tr>
<td>LinkErrorRecoveryCounter</td>
<td>Total number of times the Port Training state machine has successfully completed the link error recovery process.</td>
</tr>
<tr>
<td>PlrRcvCodes</td>
<td>Number of received PLR codewords</td>
</tr>
<tr>
<td>PlrRcvCodeErr</td>
<td>The total number of rejected codewords received</td>
</tr>
<tr>
<td>PlrRcvUncorrectableCode</td>
<td>The number of uncorrectable codewords received</td>
</tr>
<tr>
<td>PlrXmitCodes</td>
<td>Number of transmitted PLR codewords</td>
</tr>
<tr>
<td>PlrXmitRetryCodes</td>
<td>The total number of codewords retransmitted</td>
</tr>
<tr>
<td>Field Name</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>PlrXmitRetryEvents</td>
<td>The total number of retransmitted event</td>
</tr>
<tr>
<td>PlrSyncEvents</td>
<td>The number of sync events</td>
</tr>
<tr>
<td>HiRetransmissionRate</td>
<td>Received bandwidth loss due to codes retransmission</td>
</tr>
<tr>
<td>PlrXmitRetryCodesWithinTSecMax</td>
<td>The maximum number of retransmitted events in t sec window</td>
</tr>
<tr>
<td>link_partner_description</td>
<td>node description of the link partner</td>
</tr>
<tr>
<td>link_partner_node_guid</td>
<td>node_guid of the link partner</td>
</tr>
<tr>
<td>link_partner_lid</td>
<td>lid of the link partner</td>
</tr>
<tr>
<td>link_partner_port_num</td>
<td>port number of the link partner</td>
</tr>
<tr>
<td>Cable_PN</td>
<td>Vendor Part Number</td>
</tr>
<tr>
<td>Cable_SN</td>
<td>Vendor Serial Number</td>
</tr>
<tr>
<td>cable_technology</td>
<td></td>
</tr>
<tr>
<td>cable_type</td>
<td>Cable/module type</td>
</tr>
<tr>
<td>cable_vendor</td>
<td></td>
</tr>
<tr>
<td>cable_length</td>
<td></td>
</tr>
<tr>
<td>cable_identifier</td>
<td></td>
</tr>
<tr>
<td>vendor_rev</td>
<td>Vendor revision</td>
</tr>
<tr>
<td>cable_fw_version</td>
<td></td>
</tr>
<tr>
<td>rx_power_lane_[0-7]</td>
<td>RX measured power</td>
</tr>
<tr>
<td>tx_power_lane_[0-7]</td>
<td>TX measured power</td>
</tr>
<tr>
<td>Module_Voltage</td>
<td>Internally measured supply voltage</td>
</tr>
<tr>
<td>Module_Temperature</td>
<td>Module temperature</td>
</tr>
<tr>
<td>fast_link_up_status</td>
<td>Indicates if fast link-up was performed in the link</td>
</tr>
<tr>
<td>time_to_link_up_ext_msec</td>
<td>Time in msec to link up from disable until phy up state. While the phy manager did not reach phy up state the timer will return 0.</td>
</tr>
<tr>
<td>Advanced_Status_Opcode</td>
<td>Status opcode: PHY FW indication</td>
</tr>
<tr>
<td>Status_Message</td>
<td>ASCII code message</td>
</tr>
<tr>
<td>down_blame</td>
<td>Which receiver caused last link down</td>
</tr>
<tr>
<td>local_reason_opcode</td>
<td>Opcode of link down reason - local</td>
</tr>
<tr>
<td>remote_reason_opcode</td>
<td>Opcode of link down reason - remote</td>
</tr>
<tr>
<td>e2e_reason_opcode</td>
<td>see local_reason_opcode for local reason opcode for remote reason opcode: local_reason_opcode+100</td>
</tr>
<tr>
<td>PortRcvRemotePhysicalErrors</td>
<td>Total number of packets marked with the EBP delimiter received on the port.</td>
</tr>
<tr>
<td>PortRcvErrors</td>
<td>Total number of packets containing an error that were received on the port.</td>
</tr>
<tr>
<td>PortXmitDiscards</td>
<td>Total number of outbound packets discarded by the port because the port is down or congested.</td>
</tr>
<tr>
<td>Field Name</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>PortRcvSwitchRelayErrors</td>
<td>Total number of packets received on the port that were discarded because they could not be forwarded by the switch relay.</td>
</tr>
<tr>
<td>ExcessiveBufferOverrunErrors</td>
<td>The number of times that OverrunErrors consecutive flow control update periods occurred, each having at least one overrun error.</td>
</tr>
<tr>
<td>LocalLinkIntegrityErrors</td>
<td>The number of times that the count of local physical errors exceeded the threshold specified by LocalPhyErrors.</td>
</tr>
<tr>
<td>PortRcvConstraintErrors</td>
<td>Total number of packets received on the switch physical port that are discarded.</td>
</tr>
<tr>
<td>PortXmitConstraintErrors</td>
<td>Total number of packets not transmitted from the switch physical port.</td>
</tr>
<tr>
<td>VL15Dropped</td>
<td>Number of incoming VL15 packets dropped due to resource limitations (e.g., lack of buffers) in the port.</td>
</tr>
<tr>
<td>PortXmitWait</td>
<td>The time an egress port had data to send but could not send it due to lack of credits or arbitration - in time ticks within the sample-time window.</td>
</tr>
<tr>
<td>PortXmitDataExtended</td>
<td>Transmitted data rate per egress port in bytes passing through the port during the sample period.</td>
</tr>
<tr>
<td>PortRcvDataExtended</td>
<td>The received data on the ingress port in bytes during the sample period.</td>
</tr>
<tr>
<td>PortXmitPktsExtended</td>
<td>Total number of packets transmitted on the port.</td>
</tr>
<tr>
<td>PortRcvPktsExtended</td>
<td>Total number of packets received on the port.</td>
</tr>
<tr>
<td>PortUniCastXmitPkts</td>
<td>Total number of unicast packets transmitted on all VLS from the port. This may include unicast packets with errors, and excludes link packets.</td>
</tr>
<tr>
<td>PortUniCastRcvPkts</td>
<td>Total number of unicast packets, including unicast packets containing errors, and excluding link packets, received from all VLS on the port.</td>
</tr>
<tr>
<td>PortMultiCastXmitPkts</td>
<td>Total number of multicast packets transmitted on all VLS from the port. This may include multicast packets with errors.</td>
</tr>
<tr>
<td>PortMultiCastRcvPkts</td>
<td>Total number of multicast packets, including multicast packets containing errors received from all VLS on the port.</td>
</tr>
<tr>
<td>SyncHeaderErrorCounter</td>
<td>Count of errored block sync header on one or more lanes.</td>
</tr>
<tr>
<td>PortSwLifetimeLimitDiscards</td>
<td>Total number of outbound packets discarded by the port because the Switch Lifetime Limit was exceeded. Applies to switches only.</td>
</tr>
<tr>
<td>PortSwHOQLifetimeLimitDiscards</td>
<td>Total number of outbound packets discarded by the port because the switch HOQ Lifetime Limit was exceeded. Applies to switches only.</td>
</tr>
<tr>
<td>rq_num_wrfe</td>
<td>Responder - number of WR flushed errors</td>
</tr>
<tr>
<td>rq_num_lle</td>
<td>Responder - number of local length errors</td>
</tr>
<tr>
<td>sq_num_wrfe</td>
<td>Requester - number of WR flushed errors</td>
</tr>
<tr>
<td>Temp_flags</td>
<td>Latched temperature flags of module</td>
</tr>
<tr>
<td>Vcc_flags</td>
<td>Latched VCC flags of module</td>
</tr>
<tr>
<td>Field Name</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>device_hw_rev</td>
<td>Node HW Revision</td>
</tr>
<tr>
<td>sw_revision</td>
<td>switch revision</td>
</tr>
<tr>
<td>sw_serial_number</td>
<td>switch serial number</td>
</tr>
<tr>
<td>measured_freq_[0-1]</td>
<td>Clock frequency measurement in last 100msec</td>
</tr>
<tr>
<td>min_freq_[0-1]</td>
<td>Minutes of clock frequency measured. Units of 0.1 KHz</td>
</tr>
<tr>
<td>max_freq_[0-1]</td>
<td>Max of clock frequency measured. Units of 0.1 KHz</td>
</tr>
<tr>
<td>max_delta_freq_[0-1]</td>
<td>Observed max delta frequency in window of 100msec. Units of 0.1 KHz</td>
</tr>
<tr>
<td>snr_media_lane_[0-7]</td>
<td>SNR value on the media lane &lt;i&gt;. In unit scale of 1/256 dB. The SNR value represents the electrical signal-to-noise ratio on an optical lane, and is defined as the minimum of the three individual eye SNR values.</td>
</tr>
<tr>
<td>snr_host_lane_[0-7]</td>
<td>SNR value on the host lane &lt;i&gt;. In unit scale of 1/256 dB. The SNR value represents the electrical signal-to-noise ratio on an optical lane, and is defined as the minimum of the three individual eye SNR values.</td>
</tr>
<tr>
<td>tx_cdr_lol</td>
<td>Bitmask for latched Tx cdr loss of lock flag per lane.</td>
</tr>
<tr>
<td>rx_cdr_lol</td>
<td>Bitmask for latched Rx cdr loss of lock flag per lane.</td>
</tr>
<tr>
<td>tx_los</td>
<td>Bitmask for latched Tx loss of signal flag per lane.</td>
</tr>
<tr>
<td>rx_los</td>
<td>Bitmask for latched Rx loss of signal flag per lane.</td>
</tr>
<tr>
<td>phy_received_bits</td>
<td>This counter provides information on the total amount of traffic (bits) received</td>
</tr>
<tr>
<td>rq_general_error</td>
<td>The total number of packets that were dropped since it contained errors. Reasons for this include: Dropped due to MPR mismatch.</td>
</tr>
</tbody>
</table>
9 UFM Web UI

This section is constituted by the following sub-sections:

- Fabric Dashboard
- Network Map
- Managed Elements
- Events & Alarms
- Telemetry - User-Defined Sessions
- System Health
- Jobs
- Settings

9.1 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.

9.1.1 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.

### New Panels

<table>
<thead>
<tr>
<th>Category</th>
<th>Panels</th>
</tr>
</thead>
<tbody>
<tr>
<td>All</td>
<td>12</td>
</tr>
<tr>
<td>Health</td>
<td>2</td>
</tr>
<tr>
<td>Monitoring</td>
<td>7</td>
</tr>
<tr>
<td>Events and Alarms</td>
<td>3</td>
</tr>
</tbody>
</table>

9.1.2 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.

9.1.3 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

9.1.4 Top N Servers/Switches by Rx or Tx Bandwidth

The Top N servers/servers 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 to 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

![](important_icon.png)

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

The following table lists the icons of this component:

<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>List view</td>
<td>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</td>
</tr>
<tr>
<td>Options</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Bar view</strong></td>
<td>Shows the top N nodes as a bar graph</td>
</tr>
<tr>
<td></td>
<td>• X axis shows the rate as a value</td>
</tr>
<tr>
<td></td>
<td>• Y axis shows the Node (server) name</td>
</tr>
<tr>
<td><strong>Drop-down menu</strong></td>
<td>Selects the number of items to display</td>
</tr>
<tr>
<td></td>
<td>Default: 10 nodes</td>
</tr>
<tr>
<td><strong>Monitoring attributes</strong></td>
<td>Selects the attribute for monitoring:</td>
</tr>
<tr>
<td></td>
<td>• TxBW - Transmit Bandwidth</td>
</tr>
<tr>
<td></td>
<td>• RxBW - Receive Bandwidth</td>
</tr>
<tr>
<td><strong>View by port/element</strong></td>
<td>Switches view to top 5 elements by bandwidth or top 5 ports by bandwidth.</td>
</tr>
<tr>
<td></td>
<td>Nodes view is presented by default.</td>
</tr>
<tr>
<td></td>
<td>• Clicking a specific port in the ports view under the port column redirects to the ports table and highlights that particular port</td>
</tr>
<tr>
<td></td>
<td>• Clicking a specific device in the devices view under the device column redirects to the Devices table and highlights that particular node</td>
</tr>
<tr>
<td><strong>Filter toggle</strong></td>
<td>Toggles the filter textbox</td>
</tr>
</tbody>
</table>

*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 15 Servers By Bandwidth](image)

9.1.5 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.

⚠️ N can be 5, 10, 15, or 20.

Top N Congested Servers by Bandwidth—List View
### Top 5 Congested Servers

<table>
<thead>
<tr>
<th>Device</th>
<th>Normalized TCBWx Congested Bandwidth (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>r-ufm254-hyp-04</td>
<td>3896</td>
</tr>
<tr>
<td>ufm-host87</td>
<td>3506</td>
</tr>
<tr>
<td>r-ufm254-hyp-03</td>
<td>3489</td>
</tr>
</tbody>
</table>

### Top 5 Congested Switches

<table>
<thead>
<tr>
<th>Device</th>
<th>Normalized TCBWx Congested Bandwidth (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>switchib</td>
<td>1541</td>
</tr>
<tr>
<td>sw-hpc62</td>
<td>991</td>
</tr>
</tbody>
</table>

*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**

<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Bar view</strong></td>
<td>Shows the top N congested devices as a bar graph</td>
</tr>
<tr>
<td><img src="image" alt="Bar" /> <img src="image" alt="List" /></td>
<td>• X axis shows the rate as a percentage</td>
</tr>
<tr>
<td></td>
<td>• Y axis shows the congested Node (server) name</td>
</tr>
<tr>
<td><strong>List view</strong></td>
<td>Shows the top N congested nodes as a list</td>
</tr>
<tr>
<td><img src="image" alt="Bar" /> <img src="image" alt="List" /></td>
<td>Each congested node is shown in a row with the name of the node</td>
</tr>
<tr>
<td></td>
<td>and its picture. It also shows the bandwidth rate</td>
</tr>
<tr>
<td><strong>Drop-down menu</strong></td>
<td>Enables selecting the number of top N congested nodes</td>
</tr>
<tr>
<td><img src="image" alt="Select" /></td>
<td>Default: 10 nodes</td>
</tr>
<tr>
<td>Options</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>View by port/element</td>
<td>Switches view to Top 5 elements By Bandwidth or Top 5 Ports By Bandwidth. Devices view is presented by default.</td>
</tr>
<tr>
<td></td>
<td>• Clicking a specific port in the Ports view under the Port column redirects to the Ports table and highlights that particular port</td>
</tr>
<tr>
<td></td>
<td>• Clicking a specific device in the Nodes view under the Device column redirects to the Devices table and highlights that particular node</td>
</tr>
</tbody>
</table>

**Monitoring attributes**  

- **TxBW**  
- **RxBW**  

- **RCBW** - Receive Congested Bandwidth (percentage)  
- **TCBW** - Transmit Congested Bandwidth (percentage)

### 9.1.6 Top N Utilized PKeys

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

⚠️ N can be 5, 10, 15, or 20.

**Top N Utilized PKeys—List View**

![List View](image)

**Top N Utilized PKeys—Bar View**

![Bar View](image)
The following table describes the options available in this component.

**Top N Utilized PKeys**

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

9.1.7 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**

<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>List view</td>
<td>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.</td>
</tr>
<tr>
<td>Bar view</td>
<td>Shows the top N alarmed devices as a bar graph.</td>
</tr>
<tr>
<td></td>
<td>• X axis shows the number of alarms</td>
</tr>
<tr>
<td></td>
<td>• Y axis shows the names of the alarmed nodes (servers)</td>
</tr>
<tr>
<td>Drop down menu</td>
<td>Enables selecting the number of top N alarmed nodes. Selects the number of items to display.</td>
</tr>
<tr>
<td></td>
<td>Default: 10 alarmed nodes</td>
</tr>
<tr>
<td>Filter toggle</td>
<td>Toggles the Filter textbox</td>
</tr>
</tbody>
</table>

**Top Alarmed Servers/Switches—List View**
### Top 5 Alarmed Servers

<table>
<thead>
<tr>
<th>Device</th>
<th>Alarms</th>
</tr>
</thead>
<tbody>
<tr>
<td>r-ufm254-hyp-03</td>
<td>9</td>
</tr>
<tr>
<td>r-ufm254-hyp-04</td>
<td>9</td>
</tr>
<tr>
<td>ufm-host87</td>
<td>7</td>
</tr>
</tbody>
</table>

### Top 5 Alarmed Switches

<table>
<thead>
<tr>
<th>Device</th>
<th>Alarms</th>
</tr>
</thead>
<tbody>
<tr>
<td>sw-hpc62</td>
<td>9</td>
</tr>
<tr>
<td>switchib</td>
<td>8</td>
</tr>
</tbody>
</table>
9.1.8 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.

9.1.9 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.
9.1.10 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

9.1.11 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.

9.1.11.1 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


9.1.11.2 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:
   a. 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
   b. 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.
9.1.11.3 Port Group Traffic Map

Traffic Map

9.1.11.4 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:

9.1.12 Events History

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’.
9.2 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).

9.2.1 Network Map Components

<table>
<thead>
<tr>
<th>Component</th>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Switches</td>
<td>![Switch Icon]</td>
<td>Represents third party switches discovered/managed by UFM</td>
</tr>
<tr>
<td>Hosts</td>
<td>![Host Icon]</td>
<td>Represents the computer (host) connected to the discovered/managed switches</td>
</tr>
<tr>
<td>Routers</td>
<td>![Router Icon]</td>
<td>Represents third party routers discovered/managed by UFM</td>
</tr>
<tr>
<td>Gateways</td>
<td>![Gateway Icon]</td>
<td>Represents third party gateways discovered/managed by UFM</td>
</tr>
<tr>
<td>Links</td>
<td>![Link Icon]</td>
<td>Represents the connections between devices on the fabric</td>
</tr>
<tr>
<td>Racks</td>
<td>![Rack Icon]</td>
<td>Represents all nodes (hosts) physically connected to a switch</td>
</tr>
</tbody>
</table>

⚠️ 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"

9.2.2 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.

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.
9.2.3 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

![Import View](image)

• "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

![Views: Default](image)

• "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.

![Save As..](image)

• Saved views can be deleted using the "x" button.

![Saved Views](image)

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.
9.2.4 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.

9.2.5 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.
9.2.6 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.
The default layout for small fabric (less than 30 nodes) is hierarchical and for large fabric is directed.

9.2.7 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](image)

*Device Severity Levels*
### 9.2.8 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.

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>✔️</td>
<td>Info</td>
</tr>
<tr>
<td>⚠️</td>
<td>Critical</td>
</tr>
<tr>
<td>🔴</td>
<td>Minor</td>
</tr>
<tr>
<td>🤔</td>
<td>Warning</td>
</tr>
</tbody>
</table>
To define a condition:
1. Select the desired counter, and click the + button.

![Counter selection screenshot]

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.

![Condition definition screenshot]

⚠️ The colors are sorted from the lowest to the highest priority (i.e., from left to right, green to red).

⚠️ 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 how the added conditions are listed in the Network Analysis section, if Link Analysis is enabled, and they are colored accordingly.
<table>
<thead>
<tr>
<th>Property</th>
<th>Source</th>
<th>Destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>System GUID</td>
<td>0x248a070300ef19a0</td>
<td>0x7cfe900300292356</td>
</tr>
<tr>
<td>Port</td>
<td>23</td>
<td>HCA-1/1</td>
</tr>
<tr>
<td>MTU</td>
<td>4096</td>
<td>4096</td>
</tr>
<tr>
<td>Width</td>
<td>4X</td>
<td>4X</td>
</tr>
<tr>
<td>Speed</td>
<td>FDR</td>
<td>FDR</td>
</tr>
<tr>
<td>Port RX Data</td>
<td>20379.9 Gb</td>
<td>5.9 Gb</td>
</tr>
<tr>
<td>Port TX Data</td>
<td>18.05 Gb</td>
<td>98.75 Gb</td>
</tr>
<tr>
<td>Port RX Data Rate</td>
<td>0 Gb/s</td>
<td>0 Gb/s</td>
</tr>
<tr>
<td>Port TX Data Rate</td>
<td>0 Gb/s</td>
<td>0 Gb/s</td>
</tr>
<tr>
<td>Port RX Packets</td>
<td>2533520412</td>
<td>45788053</td>
</tr>
<tr>
<td></td>
<td>5 Packets</td>
<td>Packets</td>
</tr>
<tr>
<td>Port TX Packets</td>
<td>1172677890</td>
<td>44948657</td>
</tr>
<tr>
<td></td>
<td>Packets</td>
<td>Packets</td>
</tr>
<tr>
<td>Port RX Packets Rate</td>
<td>2.9 Packets/s</td>
<td>2.9 Packets/s</td>
</tr>
<tr>
<td>Port TX Packets Rate</td>
<td>2.9 Packets/s</td>
<td>2.9 Packets/s</td>
</tr>
</tbody>
</table>
Please note that if the current layout and view are saved, the defined conditions are saved inside the view being saved.

9.2.9 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
9.2.10 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:

<table>
<thead>
<tr>
<th>Property</th>
<th>Source</th>
<th>Destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>System GUID</td>
<td>0x0000000000000000</td>
<td>0x0000000000000000</td>
</tr>
<tr>
<td>Port</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>MTU</td>
<td>4096</td>
<td>4096</td>
</tr>
<tr>
<td>Width</td>
<td>4X</td>
<td>4X</td>
</tr>
<tr>
<td>Speed</td>
<td>EDR</td>
<td>EDR</td>
</tr>
<tr>
<td>Port RX Data</td>
<td>614 MB</td>
<td>164 MB</td>
</tr>
<tr>
<td>Port TX Data</td>
<td>164 MB</td>
<td>614 MB</td>
</tr>
<tr>
<td>Port RX Data Rate</td>
<td>0 MB/s</td>
<td>0 MB/s</td>
</tr>
<tr>
<td>Port TX Data Rate</td>
<td>0 MB/s</td>
<td>0 MB/s</td>
</tr>
<tr>
<td>Port RX Packets</td>
<td>162888 Packets</td>
<td>59767 Packets</td>
</tr>
<tr>
<td>Port TX Packets</td>
<td>597646 Packets</td>
<td>1662723 Packets</td>
</tr>
<tr>
<td>Port RX Packets Rate</td>
<td>0.45 Packets/s</td>
<td>0.25 Packets/s</td>
</tr>
<tr>
<td>Port TX Packets Rate</td>
<td>0.25 Packets/s</td>
<td>0.45 Packets/s</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port Number</td>
<td>MCP1602-E02A</td>
</tr>
<tr>
<td>Length</td>
<td>1 m</td>
</tr>
<tr>
<td>Serial Number</td>
<td>MT17145S00778</td>
</tr>
<tr>
<td>Identifier</td>
<td>QSFP+</td>
</tr>
<tr>
<td>Technology</td>
<td>Copper cable- unequalized</td>
</tr>
<tr>
<td>Revision</td>
<td>A2</td>
</tr>
</tbody>
</table>

9.2.11 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.
9.2.11.1 Supported Actions for Internally Managed Switches

<table>
<thead>
<tr>
<th>Element Type</th>
<th>Supported Actions</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Managed Switch</td>
<td>Reboot</td>
<td>Reboot the switch software</td>
</tr>
<tr>
<td></td>
<td>Mirroring Settings</td>
<td>Set the mirroring configuration for the switch</td>
</tr>
<tr>
<td></td>
<td>Collect System Dump</td>
<td>Collect system dump from the device</td>
</tr>
<tr>
<td></td>
<td>Software Upgrade</td>
<td>Perform switch software upgrade</td>
</tr>
<tr>
<td></td>
<td>Add to Group</td>
<td>Add switch to logical group</td>
</tr>
<tr>
<td></td>
<td>Remove from Group</td>
<td>Remove switch from logical group</td>
</tr>
<tr>
<td></td>
<td>Suppress Notification</td>
<td>Suppress all event notifications for the switch</td>
</tr>
<tr>
<td></td>
<td>Monitor</td>
<td>Configure and activate switch monitoring</td>
</tr>
<tr>
<td></td>
<td>Go to Devices</td>
<td>Go to devices page and select the device</td>
</tr>
</tbody>
</table>
9.2.11.2 Supported Actions for Externally Managed Switches

<table>
<thead>
<tr>
<th>Element Type</th>
<th>Supported Actions</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Externally Managed Switch</td>
<td>Set Node Description</td>
<td>Sets description for specific node</td>
</tr>
<tr>
<td></td>
<td>Firmware Reset</td>
<td>Perform switch firmware reset</td>
</tr>
<tr>
<td></td>
<td>Firmware Upgrade</td>
<td>Perform switch firmware upgrade</td>
</tr>
<tr>
<td></td>
<td>Add to Group</td>
<td>Add switch to logical group</td>
</tr>
<tr>
<td></td>
<td>Remove from Group</td>
<td>Remove switch from logical group</td>
</tr>
<tr>
<td></td>
<td>Suppress Notification</td>
<td>Suppress all event notifications for the switch</td>
</tr>
<tr>
<td></td>
<td>Monitor</td>
<td>Configure and activate switch monitoring</td>
</tr>
<tr>
<td></td>
<td>Go To Devices</td>
<td>Go to devices page and select the device</td>
</tr>
</tbody>
</table>
9.2.11.3 Supported Actions for Hosts

<table>
<thead>
<tr>
<th>Element Type</th>
<th>Supported Actions</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hosts</td>
<td>Firmware Upgrade</td>
<td>Perform switch firmware upgrade</td>
</tr>
<tr>
<td></td>
<td>Add to Group</td>
<td>Add host to logical group</td>
</tr>
<tr>
<td></td>
<td>Remove from Group</td>
<td>Remove host from logical group</td>
</tr>
<tr>
<td></td>
<td>Suppress Notification</td>
<td>Suppress all event notifications for the host</td>
</tr>
<tr>
<td></td>
<td>Monitor</td>
<td>Configure and activate host monitoring</td>
</tr>
</tbody>
</table>

9.3 Managed Elements

The UFM Managed Elements window allows you to obtain information on the fabric physical elements, such as devices, ports and cables.

⚠️ 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
9.3.1 Devices Window

The Devices window shows data pertaining to the physical devices in a tabular format.

<table>
<thead>
<tr>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Health</td>
<td>Health of the device reflecting the highest alarm severity. Please refer to the Health States table.</td>
</tr>
<tr>
<td>Name</td>
<td>Name of the device</td>
</tr>
<tr>
<td>GUID</td>
<td>System GUID of the device</td>
</tr>
<tr>
<td>Type</td>
<td>Type of the device: switch, node, IB router, and gateway</td>
</tr>
<tr>
<td>IP</td>
<td>IP address of the device</td>
</tr>
<tr>
<td>Vendor</td>
<td>The vendor of the device</td>
</tr>
<tr>
<td>Firmware Version</td>
<td>The firmware version installed on the device</td>
</tr>
</tbody>
</table>

**Health States**

<table>
<thead>
<tr>
<th>Icon</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![ ]</td>
<td>Normal</td>
<td>Information/notification displayed during normal operating state or a normal system event.</td>
</tr>
<tr>
<td>![ ]</td>
<td>Critical</td>
<td>Critical means that the operation of the system or a system component fails.</td>
</tr>
</tbody>
</table>
A right-click on the device name displays a list of actions that can be performed on it.

<table>
<thead>
<tr>
<th>Icon</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>?</td>
<td>Warning</td>
<td>Warning reflects a low priority problem in the fabric with no failure. A warning is asserted when an event exceeds a predefined threshold.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Devices Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Action</strong></td>
</tr>
<tr>
<td>Firmware Upgrade</td>
</tr>
<tr>
<td>Firmware Reset</td>
</tr>
<tr>
<td>Set Node Description</td>
</tr>
<tr>
<td>Collect System Dump</td>
</tr>
<tr>
<td>Add to Group</td>
</tr>
<tr>
<td>Remove from Group</td>
</tr>
<tr>
<td>Suppress Notifications</td>
</tr>
<tr>
<td>Add to Monitor Session</td>
</tr>
<tr>
<td>Show in Network Map</td>
</tr>
</tbody>
</table>
9.3.1.1 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:
9.3.1.2 Mark Device as Healthy

Server /opt/ufm/files/conf/opensm/opensm-health-policy.conf content:
9.3.1.3 Upgrading Software and Firmware for Hosts and Externally Managed Switches

9.3.1.3.1 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)
9.3.1.3.2 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_0080110009
  fw-ConnectX2-rel-2.9.1000-MHQH29B-XTR_A1.bin
  MT_0F90110002
  fw-IS4-rel-7.4.2040-MIS5023Q_A1-A5.bin
```

9.3.1.3.3 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
    fu_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
```

9.3.1.3.4 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 \textit{ibdiagnet} output (section NODES_INFO).

The below is an example of the default value:

\begin{verbatim}
event_plugin_options = (null)
\end{verbatim}

To enable the vendor-specific discovery by opensm, in the gv.cfg configuration file, change the value of event_plugin_option to \textit{(--vendinfo -m 1)}, as shown below:

\begin{verbatim}
event_plugin_options = --vendinfo -m 1
\end{verbatim}

If the value is set to \textit{--vendinfo -m 1}, the data should be supplied by opensm, and in this case the \textit{ibdiagnet} output is ignored.

\begin{itemize}
\item In some firmware versions, the information above is currently not available.
\end{itemize}

### 9.3.1.3.5 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 \textit{ibdiagnet} run (\textit{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 \texttt{[FabricAnalysisys]}, 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 \texttt{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 \textit{ibdiagnet} output.

### 9.3.1.3.6 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.
4. Click Submit to save your changes.

9.3.1.3.7 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.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP</td>
<td>Enter device IP</td>
</tr>
<tr>
<td>Path</td>
<td>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.</td>
</tr>
<tr>
<td>Username</td>
<td>Name of the host username</td>
</tr>
<tr>
<td>Password</td>
<td>Enter the host password</td>
</tr>
</tbody>
</table>

5. Click submit to save your changes.

⚠️ The firmware upgrade takes effect only after the host or externally managed switch is restarted.

9.3.1.3.8 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**

<table>
<thead>
<tr>
<th>Current Firmware Version</th>
<th>Transceiver Type</th>
<th>Image</th>
</tr>
</thead>
<tbody>
<tr>
<td>38.100.122</td>
<td>Hercules2</td>
<td>No Selected Image</td>
</tr>
</tbody>
</table>

[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.
9.3.1.4 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, and device access.

9.3.1.4.1 General Tab

Provides general information on the selected device.
9.3.1.4.2 Ports Tab

This tab provides a list of the ports connected to this device in a tabular format.

<table>
<thead>
<tr>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port Number</td>
<td>The number of ports on device.</td>
</tr>
<tr>
<td>Node</td>
<td>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.</td>
</tr>
<tr>
<td>Data Type</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>Health</td>
<td>Health of the port reflecting the highest alarm severity. Please refer to the Health States table.</td>
</tr>
<tr>
<td>State</td>
<td>Indicates whether the port is connected (active or inactive).</td>
</tr>
<tr>
<td>LID</td>
<td>The local identifier (LID) of the port.</td>
</tr>
<tr>
<td>MTU</td>
<td>Maximum Transmission Unit of the port.</td>
</tr>
</tbody>
</table>
| Speed     | 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     | 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. |

### 9.3.1.4.3 Cables Tab

This tab provides a list of the cables connected to this device in a tabular format.
**Cables Data**

<table>
<thead>
<tr>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Basic Information</strong></td>
<td></td>
</tr>
<tr>
<td>Health</td>
<td>Health of the cable reflecting the highest alarm severity. Please refer to the Health States table.</td>
</tr>
<tr>
<td>Serial Number</td>
<td>Serial number of the cable.</td>
</tr>
<tr>
<td>Identifier</td>
<td>Identifier of the cable.</td>
</tr>
<tr>
<td><strong>Source Port Information</strong></td>
<td></td>
</tr>
<tr>
<td>Source GUID</td>
<td>GUID of the source port the cable is connected to.</td>
</tr>
<tr>
<td>Source Port</td>
<td>The number of the source port the cable is connected to.</td>
</tr>
<tr>
<td><strong>Destination Port Information</strong></td>
<td></td>
</tr>
<tr>
<td>Destination GUID</td>
<td>GUID of the destination port the cable is connected to.</td>
</tr>
<tr>
<td>Destination Port</td>
<td>The number of the destination port the cable is connected to.</td>
</tr>
<tr>
<td><strong>Advanced Information</strong></td>
<td></td>
</tr>
<tr>
<td>Revision</td>
<td>Revision of the cable.</td>
</tr>
<tr>
<td>Link Width</td>
<td>The maximum link width of the cable.</td>
</tr>
<tr>
<td>Part Number</td>
<td>Part number of the cable.</td>
</tr>
<tr>
<td>Technology</td>
<td>The transmitting medium of the cable: copper/optical/etc.</td>
</tr>
<tr>
<td>Length</td>
<td>The cable length in meters.</td>
</tr>
</tbody>
</table>

**9.3.1.4.4 Groups Tab**

This tab provides a list of the groups to which the selected device belongs.
Groups Data

<table>
<thead>
<tr>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Severity</td>
<td>Aggregated severity level of the group (the highest severity level of all group members).</td>
</tr>
<tr>
<td>Name</td>
<td>Name of the group.</td>
</tr>
<tr>
<td>Description</td>
<td>Description of the group.</td>
</tr>
<tr>
<td>Type</td>
<td>Type of the group: General/Rack.</td>
</tr>
</tbody>
</table>

9.3.1.4.5 Alarms Tab

This tab provides a list of all UFM alarms related to the selected device.
### Alarms Data

<table>
<thead>
<tr>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alarms ID</td>
<td>Alarm identifier.</td>
</tr>
<tr>
<td>Source</td>
<td>Source object (device/port) on which the alarm was triggered.</td>
</tr>
<tr>
<td>Severity</td>
<td>The severity of the alarm.</td>
</tr>
<tr>
<td>Description</td>
<td>Description of the alarm.</td>
</tr>
<tr>
<td>Date/Time</td>
<td>The time when the alarm was triggered.</td>
</tr>
<tr>
<td>Reason</td>
<td>Reason for the alarm.</td>
</tr>
<tr>
<td>Count</td>
<td>Number of instances that the alarm occurred on the related source object.</td>
</tr>
</tbody>
</table>

#### 9.3.1.4.6 Events Tab

This tab provides a list of the UFM events that are related to the selected device.
Events Data

<table>
<thead>
<tr>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Severity</td>
<td>Event severity - Info, Warning, Error, Critical or Minor.</td>
</tr>
<tr>
<td>Event Name</td>
<td>The name of the event.</td>
</tr>
<tr>
<td>Source</td>
<td>The source object (device/port) on which the event was triggered.</td>
</tr>
<tr>
<td>Date/Time</td>
<td>The time when the event was triggered.</td>
</tr>
<tr>
<td>Category</td>
<td>The category of the event indicated by icons. Hovering over the icon will display the category name.</td>
</tr>
<tr>
<td>Description</td>
<td>Description of the event. Full description can be displayed by hovering over the text.</td>
</tr>
</tbody>
</table>

9.3.1.4.7 Inventory Tab

This tab provides a list of the device’s modules with information in a tabular format.

⚠️ This tab is available for switches only.
### Inventory Data

<table>
<thead>
<tr>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Severity/Health</td>
<td>Health of the module reflecting the highest alarm severity. Please refer to the Health States table.</td>
</tr>
<tr>
<td>Status</td>
<td>The module status.</td>
</tr>
<tr>
<td>Serial Number</td>
<td>Serial number of the module.</td>
</tr>
<tr>
<td>System</td>
<td>Name of the device.</td>
</tr>
<tr>
<td>Description</td>
<td>Description of the module.</td>
</tr>
<tr>
<td>Type</td>
<td>Type of the module: spine/line/etc.</td>
</tr>
<tr>
<td>Software Version</td>
<td>Firmware version installed on the module.</td>
</tr>
<tr>
<td>Part Number</td>
<td>Part number of the module.</td>
</tr>
<tr>
<td>Hardware Version</td>
<td>Hardware version of the module.</td>
</tr>
<tr>
<td>Power</td>
<td>Power supply of the PSU.</td>
</tr>
</tbody>
</table>

#### 9.3.1.4.8 HCAs Tab

This tab provides a list of the device’s HCAs with information in a tabular format.

⚠️ This tab is available for hosts only.
<table>
<thead>
<tr>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Health</td>
<td>Health of the HCA reflecting the highest alarm severity. Please refer to the Health States table.</td>
</tr>
<tr>
<td>Name</td>
<td>HCA Index</td>
</tr>
<tr>
<td>GUID</td>
<td>HCA GUID</td>
</tr>
<tr>
<td>Type</td>
<td>HCA Type</td>
</tr>
<tr>
<td>Port GUID</td>
<td>HCA ports GUIDs</td>
</tr>
<tr>
<td>PSID</td>
<td>HCA PSID</td>
</tr>
<tr>
<td>FW Version</td>
<td>HCA firmware version</td>
</tr>
</tbody>
</table>

9.3.1.4.9 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).
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)

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.
2. Click Update to save your changes.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>User</td>
<td>Fill in or edit the computer user name.</td>
</tr>
<tr>
<td>Password</td>
<td>Enter the device password.</td>
</tr>
<tr>
<td>Confirmation</td>
<td>Enter the device password a second time to confirm.</td>
</tr>
<tr>
<td>Manual IP</td>
<td>Enter the device IP address (could be IPv4/IPv6).</td>
</tr>
<tr>
<td>Port</td>
<td>Enter the port number.</td>
</tr>
<tr>
<td>Timeout</td>
<td>Enter the connection timeout (in seconds) for the device specific protocol (SSH/HTTP/IPMI).</td>
</tr>
</tbody>
</table>

**Device Access Credentials Parameters**

**9.3.1.4.10 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.
9.3.2 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:

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:
## Cable Information - 7cfe90300f73be0_1

### Operational Info

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vendor Serial Number</td>
<td>MT1515V5S783T</td>
</tr>
<tr>
<td>Vendor Part Number</td>
<td>MCP1500-E031</td>
</tr>
<tr>
<td>Vendor Name</td>
<td>Mellanox</td>
</tr>
<tr>
<td>Attenuation [5g,7g,12g]</td>
<td>4,5,9</td>
</tr>
<tr>
<td>Bias Current [mA]</td>
<td>N/A</td>
</tr>
<tr>
<td>Cable Technology</td>
<td>Copper cable unequalized</td>
</tr>
<tr>
<td>Cable Type</td>
<td>Passive copper cable</td>
</tr>
<tr>
<td>CDR RX</td>
<td>N/A</td>
</tr>
<tr>
<td>CDR TX</td>
<td>N/A</td>
</tr>
<tr>
<td>Compliance</td>
<td>N/A</td>
</tr>
<tr>
<td>Digital Diagnostic Monitoring</td>
<td>No</td>
</tr>
<tr>
<td>RW Version</td>
<td>N/A</td>
</tr>
<tr>
<td>Identifier</td>
<td>QSFP+</td>
</tr>
<tr>
<td>LOS Alarm</td>
<td>N/A</td>
</tr>
<tr>
<td>OUI</td>
<td>Mellanox</td>
</tr>
<tr>
<td>Power Class</td>
<td>1.8 W max</td>
</tr>
<tr>
<td>Rev</td>
<td>A2</td>
</tr>
<tr>
<td>Rx Power Current [dBm]</td>
<td>N/A</td>
</tr>
<tr>
<td>Temperature [°C]</td>
<td>N/A</td>
</tr>
<tr>
<td>Transfer Distance [m]</td>
<td>1</td>
</tr>
<tr>
<td>Tx Power Current [dBm]</td>
<td>N/A</td>
</tr>
<tr>
<td>Voltage [mV]</td>
<td>N/A</td>
</tr>
<tr>
<td>Wavelength [nm]</td>
<td>N/A</td>
</tr>
</tbody>
</table>

## Cable Information - 7cfe90300f73be0_1

### Operational Info

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Auto Negotiation</td>
<td>ON</td>
</tr>
<tr>
<td>FEC</td>
<td>Standard LL-RS-FEC - RS(271,257)</td>
</tr>
<tr>
<td>Loopback Mode</td>
<td>No Loopback</td>
</tr>
<tr>
<td>Physical state</td>
<td>LinkUp</td>
</tr>
<tr>
<td>Speed</td>
<td>B-E0R</td>
</tr>
<tr>
<td>State</td>
<td>Active</td>
</tr>
<tr>
<td>Width</td>
<td>Ox</td>
</tr>
<tr>
<td>Enabled Link Speed</td>
<td>0x0000003F (EDR,FDR,FDR10,GDR,DDR,SDR)</td>
</tr>
<tr>
<td>Supported Cable Speed</td>
<td>0x0000003F (EDR,FDR,FDR10,GDR,DDR,SDR)</td>
</tr>
</tbody>
</table>
9.3.2.1 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.

9.3.2.2 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.
9.3.3 Virtual Ports Window

⚠️ 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.

9.3.4 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.
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.

**To control Partial Switch ASIC Failure event:**
9.3.4.1 Unhealthy Port Connectivity Filter

It is possible 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

9.3.4.2 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:

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.
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.

9.3.5 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 and navigate to peer port.

9.3.6 Groups Window

The Groups window allows users to create new groups of devices and provides information about existing groups.
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)
9.3.7 Inventory Window

Provides a list of all modules in UFM. For more information, see Device's Inventory Tab.

9.3.8 PKeys Window

The PKeys window allows users to create new groups of ports and provides information about existing PKeys.

⚠️ 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.

9.3.8.1 Creating New PKey

➢ To create a 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.

**9.3.8.2 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.

**9.3.8.3 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.
9.3.8.4 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.

For information on partitioning, refer to Appendix - Partitioning.

Note that restarting OpenSM is required for the QoS parameters change to take effect.

9.3.8.5 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.
Click on a device to select the members ports from the Available list:

New PKey

Previous

Finish
9.3.9  HCAs Window

Provides a list of all the HCAs of the hosts in UFM. For more information, see section “HCAs Tab”.

9.4  Events & Alarms

All information 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 before 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.

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_persistence_enabled = true # - will set to true for the feature to work
9.4.1 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.
9.4.2 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.
9.4.2.1 Cable Transceiver Temperatures

The UFM has alarms that notify the user in cases where an active cable overheats/overcools. The UFM uses ibdiagnet to get cable temperature analysis and report exceptions via the Alarms view.

Related events:
- 919 for high cable temperature
- 920 for low cable temperature

9.4.2.1.1 GUI Views

9.4.2.1.1.1 Alarms

9.4.2.1.1.2 Event Policy

9.5 Telemetry - User-Defined Sessions

Error: null
9.6 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**

### 9.6.1 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.

![System Health report](image)

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**

<table>
<thead>
<tr>
<th>Check</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release Number</td>
<td>UFM software version and build.</td>
</tr>
</tbody>
</table>
### UFM Configuration

<table>
<thead>
<tr>
<th>Check</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>License Type</td>
<td>Type of license, permanent or evaluation.</td>
</tr>
<tr>
<td>License Customer Number</td>
<td>The customer number provided by NVIDIA.</td>
</tr>
<tr>
<td>License UID</td>
<td>The UFM serial number provided by NVIDIA.</td>
</tr>
<tr>
<td>License Expiration Date</td>
<td>License expiration date for limited licenses.</td>
</tr>
<tr>
<td>License Functionality</td>
<td>Level of functionality enabled for the end-user, standard or advanced.</td>
</tr>
<tr>
<td>License Devices Limit</td>
<td>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)</td>
</tr>
<tr>
<td>Running Mode</td>
<td>UFM running mode, Standalone or High Availability (HA). When UFM is in HA mode, additional information is displayed for the master and standby servers.</td>
</tr>
</tbody>
</table>

### UFM Processing

<table>
<thead>
<tr>
<th>Check</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OpenSM</td>
<td>Status of the OpenSM service.</td>
</tr>
<tr>
<td>ibpm</td>
<td>Status of the ibpm (Performance Manager) service.</td>
</tr>
<tr>
<td>ModelMain</td>
<td>Status of the main UFM service.</td>
</tr>
<tr>
<td>httpd</td>
<td>Status of the httpd service.</td>
</tr>
<tr>
<td>MySql</td>
<td>Status of the MySql service.</td>
</tr>
</tbody>
</table>

### Memory Monitoring

<table>
<thead>
<tr>
<th>Check</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total memory usage</td>
<td>Percentage of total memory usage.</td>
</tr>
<tr>
<td>UFM memory usage</td>
<td>Percentage of UFM memory usage</td>
</tr>
</tbody>
</table>

### CPU Monitoring

<table>
<thead>
<tr>
<th>Check</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total CPU Capacity</td>
<td>Percentage of CPU capacity available</td>
</tr>
<tr>
<td>CPUs Number</td>
<td>Number of CPUs</td>
</tr>
<tr>
<td>Total CPU utilization</td>
<td>Percentage of total CPU utilization.</td>
</tr>
<tr>
<td>UFM CPU utilization</td>
<td>Percentage of UFM CPU utilization.</td>
</tr>
</tbody>
</table>
### Disk Monitoring

<table>
<thead>
<tr>
<th>Check</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disk &lt;diskname&gt;</td>
<td>Percentage of disk usage.</td>
</tr>
</tbody>
</table>

### Fabric Interface

<table>
<thead>
<tr>
<th>Check</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fabric Interface</td>
<td>Name and state of fabric interface.</td>
</tr>
</tbody>
</table>

## 9.6.2 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.
9.6.2.1 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.

9.6.2.2 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.

9.6.2.3 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.

9.6.3 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 Support.

By default and upon UFM bring up, the UFM creates an initial system dump. 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:

```bash
#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:

- Barematal 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

**9.6.4 Fabric Health Tab**

Through Fabric Health tab, you can access the fabric health reports. There are two kinds of reports:

- Custom Reports - The user can generate a report that runs a series of checks on the fabric on demand.
• Periodic Reports - An automatically generated report that is periodically generated by the UFM.

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.

9.6.4.1 Custom Reports

➤ To run a new report, do the following:

1. Click “Run New Report.”

Results will be displayed automatically:

![Fabric Health Report](image)

The report displays the following:

- A report summary table of the errors and warnings generated by the report.
- 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

<table>
<thead>
<tr>
<th>Check</th>
<th>Description</th>
<th>To run, select:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Duplicate/Zero LID Check</td>
<td>Lists all ports with same LID or zero LID value.</td>
<td>LIDs Check Default: Selected</td>
</tr>
<tr>
<td>Duplicated Node Description</td>
<td>Lists all nodes with same node description. Does not include switches with the same description.</td>
<td>Duplicated Node Description Default: Selected</td>
</tr>
<tr>
<td>Use Node GUID-Description Mapping</td>
<td>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”</td>
<td>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.</td>
</tr>
</tbody>
</table>
| 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 |
| 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. |
<table>
<thead>
<tr>
<th>Check</th>
<th>Description</th>
<th>To run, select:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Effective Ber Check</td>
<td>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).</td>
<td>Effective Ber Check Default: Selected</td>
</tr>
<tr>
<td>Effective Port Grade</td>
<td>Provides a grade per port lane in the fabric, which indicates the current port lane quality.</td>
<td>Physical Port Grade Default: Not Selected</td>
</tr>
<tr>
<td>Firmware Check</td>
<td>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.</td>
<td>Firmware Version Check Default: Selected</td>
</tr>
<tr>
<td>Eye Open Check</td>
<td>(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.</td>
<td>Eye Open Check Default: Selected Minimum and Maximum port bound: By default no bounds are defined.</td>
</tr>
<tr>
<td>Cable Information</td>
<td>Reports cable information as stored in EEPROM on each port: cable vendor, type, length and serial number.</td>
<td>Cable Type Check &amp; Cable Diagnostics Default: NOT selected because this test might take a long time to complete (40 msec per port)</td>
</tr>
<tr>
<td>UFM Alarms</td>
<td>Lists all open alarms in UFM.</td>
<td>UFM Alarms Default: Selected</td>
</tr>
</tbody>
</table>

### 9.6.4.2 Periodic Reports

The periodic reports are generated automatically upon UFM bring up.
To edit the report parameters, click the "Edit Report Parameters".

9.6.5 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.

9.6.5.1 Activating and Deactivating the Daily Report

Daily Report can be activated/deactivated via the `/opt/ufm/conf/gv.cfg` file.

⚠️ 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.

```plaintext
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.

```plaintext
daily_report_enabled = true
```

9.6.5.2 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.

### 9.6.5.3 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.

```ini
[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
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).

9.6.5.4 Report Content

9.6.5.4.1 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.

9.6.5.4.2 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."
9.6.5.4.3 Available Charts

9.6.5.4.3.1 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.

![Graph of events by severity]

Hovering over the bars in the interactive report displays the amount of events per hour.

9.6.5.4.3.2 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.

Hovering over the bars in the interactive report displays the percentage of the traffic/congestion per hour.

9.6.5.4.3.3 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
9.6.5.4.3.4 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.

Hovering over the slices in the interactive report displays the percentage of hosts in this group.

9.6.5.4.3.5 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.
Hovering over the bars in the interactive report displays the percentage of the traffic/congestion for a selected host.

9.6.5.4.3.6 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.

Hovering over the bars in the interactive report displays the amount of events per severity for a selected host.

9.6.5.4.3.7 Hosts with Most Critical Events
Hosts with Most Critical Events displays in a graphical view the hosts with the most critical events.
9.6.5.4.3.8 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.

Hovering over the bars in the interactive report displays the amount of critical events for a selected host.

9.6.5.4.3.9 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.

Hovering over the bars in the interactive report displays the percentage of the congestion for a selected host.
9.6.5.4.3.10 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.

Hovering over the bars in the interactive report displays the amount of link-down events for a selected host.

9.6.5.4.3.11 Switches with Most Critical Events

Switches with Most Critical Events displays in a graphical view the switches with the most critical events.
9.6.5.4.3.12 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.

Hovering over the bars in the interactive report displays the amount of critical events for a selected switch.

9.6.5.4.3.13 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.

Hovering over the bars in the interactive report displays the percentage of the congestion for a selected switch.
9.6.5.4.4 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/recipient will be added to the Daily Reports Recipients list.
These recipients will automatically start receiving the UFM daily reports.

9.6.6 Topology Compare Tab

9.6.6.1 Overview

The Topology Compare tab allows two methods of topology comparison:

- Periodic Comparison - allows users to compare the current fabric topology with a preset master topology.
- Custom Comparison - compares user-defined topology with the current fabric topology.

9.6.6.1.1 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".

Once a topology comparison is initiated by the UFM, the latest (current) topology is compared with the master topology. Upon UFM's bring-up and by default, the UFM sets the latest topology as master.

- To update the master topology with the latest (current) topology or a custom topology saved in an 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.

9.6.6.1.2 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.

![Image]

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.

9.6.6.2 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.**

   ![Image of Topology Compare Report]

2. **Browse for the required topology setup file in the Load Topology File dialog box.**

   ![Image of Load Topology File dialog box]

3. **Click Load.**

   UFM will compare topologies and display the results.

   ![Image of Topology Compare Report with detected differences]

### 9.6.7 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.
<table>
<thead>
<tr>
<th>Test</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Check Lids</td>
<td>Checks for bad lids. Possible lid errors are:</td>
</tr>
<tr>
<td></td>
<td>• zero lid</td>
</tr>
<tr>
<td></td>
<td>• lid duplication</td>
</tr>
<tr>
<td>Check Links</td>
<td>Checks for connectivity issues where all ports connected are not in the same state (active)</td>
</tr>
<tr>
<td>Check Subnet Manager</td>
<td>Checks for errors related to subnet manager. Possible SM errors are:</td>
</tr>
<tr>
<td></td>
<td>• Failed to get SMInfo Mad</td>
</tr>
<tr>
<td></td>
<td>• SM Not Found</td>
</tr>
<tr>
<td></td>
<td>• SM Not Correct (master SM with wrong priority)</td>
</tr>
<tr>
<td></td>
<td>• Many master SMs exists</td>
</tr>
<tr>
<td>Check Duplicate Nodes</td>
<td>Checks for duplications in nodes description</td>
</tr>
<tr>
<td>Check Duplicate GUIDs</td>
<td>Checks for GUIDs duplications</td>
</tr>
<tr>
<td>Check Routing</td>
<td>Checks for failures in getting routing MADs</td>
</tr>
<tr>
<td>Check Link Speed</td>
<td>Checks for errors related to link speed. Possible link speed errors are:</td>
</tr>
<tr>
<td></td>
<td>• Different speed between ports</td>
</tr>
<tr>
<td></td>
<td>• Wrong configuration - 'enable' not part of the 'supported'</td>
</tr>
<tr>
<td></td>
<td>• Unexpected speed</td>
</tr>
<tr>
<td>Check Link Width</td>
<td>Checks for errors related to link width. Possible link width errors are:</td>
</tr>
<tr>
<td></td>
<td>• Different width between ports</td>
</tr>
<tr>
<td></td>
<td>• Wrong configuration - 'enable' not part of the 'supported'</td>
</tr>
<tr>
<td></td>
<td>• Unexpected width</td>
</tr>
<tr>
<td>Check Partition Key</td>
<td>Checks for errors related to PKey. Possible PKey errors are:</td>
</tr>
<tr>
<td></td>
<td>• Failed to get Pkey Tables</td>
</tr>
<tr>
<td></td>
<td>• Mismatching pkeys between ports</td>
</tr>
<tr>
<td>Check Temperature</td>
<td>Checks for failure in getting temperature sensing.</td>
</tr>
<tr>
<td>Test</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Check Cables</td>
<td>Checks for errors related to cables. Possible cable errors are:</td>
</tr>
<tr>
<td></td>
<td>• This device does not support cable info capability</td>
</tr>
<tr>
<td></td>
<td>• Failed to get cable information (provides a reason)</td>
</tr>
<tr>
<td>Check Effective BER</td>
<td>Checks that the Effective BER does not exceed the threshold</td>
</tr>
<tr>
<td>Dragonfly Topology Validation</td>
<td>Validate if the topology is Dragonfly</td>
</tr>
<tr>
<td>SHARP Fabric Topology Validation</td>
<td>Checks for SHARP Configurations in the fabric</td>
</tr>
<tr>
<td>Tree Topology Validation</td>
<td>Checks if the fabric is a tree topology</td>
</tr>
<tr>
<td>Socket Direct Mode Reporting</td>
<td>Presents the inventory of fabric HCAs that are using socket direct</td>
</tr>
</tbody>
</table>

To run a specific test, click the play button. The job will be displayed once completed.

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.

If the data is related to a port the context menu will contain both port and device options.
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.

![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.
3. In the Run screen:
   a. 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.
b. 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.

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

---

**UFM restart is required for these changes to take effect.**

### 9.7 Jobs

**⚠️ 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.
9.8 Settings

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**

9.8.1 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:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description/Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Event</td>
<td>Event description.</td>
</tr>
<tr>
<td>Category</td>
<td>Event category, such as Communication Error and Hardware represented by icons.</td>
</tr>
<tr>
<td>Mail</td>
<td>When selected, the corresponding events will be sent a list of recipients according to Configuring Email-on-Events.</td>
</tr>
<tr>
<td>Web UI</td>
<td>When selected, the corresponding events are displayed in the Events &amp; Alarms window in the Web UI.</td>
</tr>
<tr>
<td>Alarm</td>
<td>Select the Alarm option to trigger an alarm for a specific event. When selected, the alarms will appear in the Events &amp; Alarms window in the Web UI.</td>
</tr>
<tr>
<td>Syslog</td>
<td>When checked along with the Log file option, the selected events will be written to Syslog.</td>
</tr>
<tr>
<td>Log File</td>
<td>Select the Log File option if you would like the selected event to be reported in a log file.</td>
</tr>
<tr>
<td>SNMP</td>
<td>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.</td>
</tr>
<tr>
<td>Threshold</td>
<td>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.</td>
</tr>
<tr>
<td>TTL (Sec)</td>
<td>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.</td>
</tr>
<tr>
<td>Action</td>
<td>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.</td>
</tr>
<tr>
<td>Severity</td>
<td>Select the severity level of the event and its alarm from the drop-down list: Info, Warning, Minor, and Critical.</td>
</tr>
</tbody>
</table>
9.8.1.1 Events Policy Simulation

This feature enables you to simulate one or multiple event policies. To perform a simulation, choose one or multiple events from the events policy table, right-click, and then select the "Simulate" action from the context menu.

To view the simulated events policy, navigate to the Events & Alarms tab.

9.8.1.2 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):
   a. Set the (snmp_listeners) IP addresses and ports
   b. Port is optional - the default port number is 162
   c. 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
   ```

9.8.1.3 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:

Configure SMTP settings under Settings window → Email tab - see Email Tab.

1. Configure the Recipients List under Settings → Events Policy.
2. Click New.

![Events Policy - Recipients](image)

3. In the Recipients List window, enter valid recipient email addresses, comma-separated, and click Submit.

![New Recipients](image)

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.
9.8.2 Device Access

You can configure default access parameters for remote administration via the following protocols:
- **Switch SSH** - allows you to define the SSH parameters to open an SSH session on your switch
- **Server SSH** - allows you to define the SSH parameters to open an SSH session on your server
- **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.

⚠️ 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**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>User</td>
<td>The name of the user allowed remote access.</td>
</tr>
<tr>
<td>Password</td>
<td>Enter the user password.</td>
</tr>
<tr>
<td>Confirmation</td>
<td>Re-enter the password.</td>
</tr>
<tr>
<td>Port</td>
<td>Each communication protocol has a default port for connection. You can modify the port number, if required.</td>
</tr>
<tr>
<td>Timeout</td>
<td>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.</td>
</tr>
</tbody>
</table>
9.8.3 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.

9.8.3.1 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)

9.8.3.2 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.

<table>
<thead>
<tr>
<th>Routing Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LID Matrix Dump File</td>
<td>/opt/ufm/files/conf/opensm/lid_matrix.conf</td>
</tr>
<tr>
<td>LFTS File</td>
<td>/opt/ufm/files/conf/opensm/lft.conf</td>
</tr>
<tr>
<td>Root GUID File</td>
<td>/opt/ufm/files/conf/opensm/root_guid.conf</td>
</tr>
<tr>
<td>Compute Nodes File</td>
<td>N/A</td>
</tr>
<tr>
<td>Node IDs File</td>
<td>N/A</td>
</tr>
<tr>
<td>GUID Routing Order File</td>
<td>N/A</td>
</tr>
<tr>
<td>Active Routing Engine</td>
<td>minhop</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Field/Box</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LID Matrix Dump File</td>
<td>File holding the LID matrix dump configuration</td>
</tr>
<tr>
<td>LFTS File</td>
<td>File holding the LFT routing configuration</td>
</tr>
<tr>
<td>Root GUID File</td>
<td>File holding the root node GUIDs (for fat-tree or Up/Down)</td>
</tr>
<tr>
<td>Compute Nodes File</td>
<td>File holding GUIDs of compute nodes for fat-tree routing algorithm</td>
</tr>
<tr>
<td>GUID Routing Order File</td>
<td>File holding the routing order GUIDs (for MinHop and Up/Down)</td>
</tr>
<tr>
<td>Node IDs File</td>
<td>File holding the node IDs</td>
</tr>
<tr>
<td>Active Routing Engine</td>
<td>The current active routing algorithm used by the managing OpenSM</td>
</tr>
</tbody>
</table>
9.8.4 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.

9.8.4.1 SM Keys Configuration

The SM Keys tab enables you to view the Subnet Manager Keys. You cannot change the configuration in this tab.
### Keys

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>MKey</td>
<td>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.</td>
<td>0x0000000000000000</td>
</tr>
<tr>
<td>SA Key</td>
<td>Shows the SM_Key value to qualify the receive SA queries as 'trusted'.</td>
<td>0x0000000000000000</td>
</tr>
<tr>
<td>Subnet Prefix</td>
<td>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.</td>
<td>0xfe80000000000000</td>
</tr>
<tr>
<td>SM Key</td>
<td>Read-only field that displays the Key of the Subnet Manager (SM).</td>
<td>0x0000000000000000</td>
</tr>
<tr>
<td>MKey Lease Period</td>
<td>A field that allows you to view or edit the lease period used for the M_Key on this subnet in [sec].</td>
<td>0</td>
</tr>
<tr>
<td>LMC</td>
<td>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.</td>
<td>0</td>
</tr>
<tr>
<td>No Partition Enforcement</td>
<td>Disables partition enforcement by switches.</td>
<td>Disabled</td>
</tr>
</tbody>
</table>

### 9.8.4.2 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.”

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Packet Life Time</td>
<td>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^&lt;packet_life_time&gt;. The value 0x14 disables this mechanism.</td>
<td>0x12</td>
</tr>
<tr>
<td>Subnet Timeout</td>
<td>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^&lt;subnet_timeout&gt;</td>
<td>18</td>
</tr>
<tr>
<td>Maximal Operational VL</td>
<td>A field that allows you to view and/or edit the limit of the maximal operational VLS: 0: NO_CHANGE 1: VL0  2: VL0_VL1 3: VL0_VL3 4: VL0_VL7 5: VL0_VL14</td>
<td>3</td>
</tr>
<tr>
<td>Head of Queue Life Time</td>
<td>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^&lt;head of queue lifetime&gt;. The value 0x14 disables this mechanism.</td>
<td>0x12</td>
</tr>
<tr>
<td>Leaf Head of Queue Life Time</td>
<td>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.</td>
<td>0x10</td>
</tr>
<tr>
<td>VL Stall Count</td>
<td>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.</td>
<td>0x07</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
<td>Default</td>
</tr>
<tr>
<td>------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>---------</td>
</tr>
<tr>
<td>Leaf VL Stall Count</td>
<td>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.</td>
<td>0x07</td>
</tr>
<tr>
<td>Force Link Speed</td>
<td>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</td>
<td>15</td>
</tr>
<tr>
<td>Local Physical Error Threshold</td>
<td>A field that allows you to view and/or edit the threshold of local phy errors for sending Trap 129.</td>
<td>0x08</td>
</tr>
<tr>
<td>Overrun Errors Threshold</td>
<td>A field that allows you to view and/or edit the threshold of credit overrun errors for sending Trap 130.</td>
<td>0x08</td>
</tr>
</tbody>
</table>

### 9.8.4.3 SM Lossy Manager Configuration

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.

![Image of the SM Lossy tab](image)

### 9.8.4.4 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.

9.8.4.5 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.”
<table>
<thead>
<tr>
<th>Field/Box</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sweep Interval</td>
<td>A field that allows you to view and/or edit the number of seconds between light sweeps (0 disables it).</td>
<td>10</td>
</tr>
<tr>
<td>Reassign LIDs</td>
<td>If enabled, causes all LIDs to be reassigned.</td>
<td>Disabled</td>
</tr>
<tr>
<td>Sweep on Trap</td>
<td>If enabled, traps 128 and 144 will cause a heavy sweep.</td>
<td>Enabled</td>
</tr>
<tr>
<td>Force Heavy Sweep</td>
<td>If enabled, forces every sweep to be a heavy sweep.</td>
<td>Disabled</td>
</tr>
</tbody>
</table>

**9.8.4.6 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.

<table>
<thead>
<tr>
<th>Field/Box</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>SM Priority</td>
<td>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.</td>
<td>15</td>
</tr>
<tr>
<td>Polling Timeout</td>
<td>A field that shows the timeout in [sec] between two polls of active master SM.</td>
<td>5 (sec)</td>
</tr>
<tr>
<td>Polling Retries</td>
<td>Number of failing polls of remote SM that declares it &quot;not operational.&quot;</td>
<td>4</td>
</tr>
<tr>
<td>Honor GUID to LID File</td>
<td>If enabled, honor the guid2lid file when coming out of standby state, if the file exists and is valid.</td>
<td>false</td>
</tr>
<tr>
<td>Ignore other SMs</td>
<td>If enabled, other SMs on the subnet are ignored.</td>
<td>false</td>
</tr>
</tbody>
</table>
9.8.4.7 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.

<table>
<thead>
<tr>
<th>Field/Box</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max Wire SMPs</td>
<td>A field that shows the maximum number of SMPs sent in parallel.</td>
<td>4</td>
</tr>
<tr>
<td>Transaction Timeout</td>
<td>A field that shows the maximum time in [msec] allowed for a transaction to complete.</td>
<td>200</td>
</tr>
<tr>
<td>Max Message FIFO Timeout</td>
<td>A field that shows the maximum time in [msec] a message can stay in the incoming message queue.</td>
<td>10000</td>
</tr>
<tr>
<td>Single Thread</td>
<td>When enabled, a single thread is used for handling SA queries.</td>
<td>Disabled</td>
</tr>
</tbody>
</table>

9.8.4.8 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.”

<table>
<thead>
<tr>
<th>Field/Box</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Log File</td>
<td>Path of the Log file to be used.</td>
<td><code>/cond/opt/ufm/files/log/opensm.log</code></td>
</tr>
<tr>
<td>Log Max Size</td>
<td>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.</td>
<td>4096</td>
</tr>
<tr>
<td>Dump Files Directory</td>
<td>The directory that holds the SM dump file.</td>
<td><code>/opt/ufm/files/log</code></td>
</tr>
<tr>
<td>Force Log Flush</td>
<td>Force flush to the log file for each log message.</td>
<td>Disabled</td>
</tr>
<tr>
<td>Accumulate Log File</td>
<td>If enabled, the log accumulates over multiple SM sessions.</td>
<td>Enabled</td>
</tr>
<tr>
<td>Log Levels</td>
<td>Available log levels: Error, Info, Verbose, Debug, Funcs, Frames, Routing, and Sys.</td>
<td>Error and Info</td>
</tr>
</tbody>
</table>

### 9.8.4.9 SM Miscellaneous Settings

The Misc tab enables you to view additional Subnet Manager Configuration parameters. You cannot change the configuration in this tab.
<table>
<thead>
<tr>
<th>Field/Box</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Node Names Map File</td>
<td>A field that allows you to view and/or set the node name map for mapping nodes to more descriptive node descriptions.</td>
<td>None</td>
</tr>
<tr>
<td>SA Database File</td>
<td>SA database file name</td>
<td>None</td>
</tr>
<tr>
<td>No Clients Reregistration</td>
<td>If enabled, disables client re-registration.</td>
<td>Disabled</td>
</tr>
<tr>
<td>Disable Multicast</td>
<td>If enabled, the SM disables multicast support and no multicast routing is performed.</td>
<td>Disabled</td>
</tr>
<tr>
<td>Exit on Fatal Event</td>
<td>If enabled, the SM exits on fatal initialization issues.</td>
<td>Enabled</td>
</tr>
</tbody>
</table>

**9.8.4.10 SM QoS Configuration**

The QoS tab allows you to enable or disable QoS functionality. QoS is disabled by default.
9.8.4.11 SM Congestion Control Configuration

The Congestion Control tab allows you to enable, disable, or ignore congestion control.
9.8.4.12 SM Adaptive Routing Configuration

The Adaptive Routing tab allows you to configure adaptive routing parameters.
9.8.5 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.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ignore</td>
<td>Ignore the non-optimal links</td>
</tr>
<tr>
<td>Reset</td>
<td>Reset all non-optimal links ports</td>
</tr>
<tr>
<td>Disable</td>
<td>Disable all non-optimal links ports</td>
</tr>
</tbody>
</table>

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.

### 9.8.6 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).

9.8.7 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:
<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
</table>
| 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                                                                                           |
| Use SSL          | Default value is false - not using SSL                                                                                                                                                                       |
| Username         | SMTP account username                                                                                                                                                                                       |
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:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Password</td>
<td>SMTP account password</td>
</tr>
</tbody>
</table>

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.

9.8.8 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.
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Protocol</td>
<td>The protocol to use to move the dump file to the external storage (scp/sftp)</td>
</tr>
<tr>
<td>Server</td>
<td>Hostname or IP address of the server</td>
</tr>
<tr>
<td>Path</td>
<td>The path where dump files are saved</td>
</tr>
<tr>
<td>Username</td>
<td>Username for the server</td>
</tr>
<tr>
<td>Password</td>
<td>Respective password</td>
</tr>
</tbody>
</table>

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.

### 9.8.9 Data Streaming

This section allows users to configure System Logs settings via web UI.
### Data Streaming Configurations

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Status</td>
<td>Enable/disable exporting UFM logs to system logs</td>
</tr>
<tr>
<td>Mode</td>
<td>Export logs to local or remote system logs</td>
</tr>
<tr>
<td>Destination</td>
<td>Remote server IP/hostname and port</td>
</tr>
<tr>
<td>System Logs Level</td>
<td>Log level to export</td>
</tr>
<tr>
<td>Streaming Data</td>
<td>Logs to export to system logs.</td>
</tr>
</tbody>
</table>

**Events logs are selected one by one from Events Policy settings when the system logs feature is enabled.**

### 9.8.10 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

9.8.11 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.

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):

<table>
<thead>
<tr>
<th>Name</th>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Revoke</td>
<td><img src="trash_can.png" alt="Trash Can" /></td>
<td>Revoke a specific token. The revoked token will no longer be valid.</td>
</tr>
<tr>
<td>Delete</td>
<td><img src="trash_can.png" alt="Trash Can" /></td>
<td>Delete a specific token.</td>
</tr>
<tr>
<td>Copy</td>
<td><img src="clipboard.png" alt="Clipboard" /></td>
<td>Copy specific token into the clipboard.</td>
</tr>
</tbody>
</table>

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.
9.8.12 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.

- **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.

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.
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.

**9.8.13 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.

9.8.13.1 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.
9.8.13.2 Updating Custom Roles
1. Select the role that requires updating.

2. Modify the allowed list from the role information section.

9.8.13.3 Deleting Custom Roles
1. Right-click on the role that needs deletion.
2. Choose the "Delete" option from the context menu.

Deleting and updating predefined roles is not permitted.

9.8.13.4 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.
9.8.14 User Preferences

This page allows user to change UI preferences in general.

When user enables dark mode, the UFM is presented in dark theme.
10 Multi-Subnet UFM

10.1 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.

10.2 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.

10.3 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.

a. 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.
b. By editing the Provider view, you can change Provider's credentials.

c. 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.
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.

   a. Consumer System Health tab contains sections applicable to Consumer UFM specifically (e.g., logs from Consumer UFM).

   b. 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.**

![UFM Health tab](image)

**System Health**

<table>
<thead>
<tr>
<th>Site</th>
<th>Provider</th>
<th>Test</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>provider_100</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>provider_200</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**UFM Health Report**

- No reports available. Please click on Run New Report to generate one.
10. Fabric Health contains sub report tables for each provider.

11. Daily Reports:
   a. Consumer Daily reports display consumer reports.
b. 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:
a. Consumer logs:

Consumer System Health

<table>
<thead>
<tr>
<th>Log View</th>
</tr>
</thead>
</table>

In the "System Dump" tab:

b. Providers logs display providers log separately, displaying logs for all providers is not supported.

14. In the "System Dump" tab:

a. "Consumer System Dump" collects system dump for consumer

Providers System Health

<table>
<thead>
<tr>
<th>Log View</th>
</tr>
</thead>
</table>

The last system dump was created successfully. Click here to download it.
b. "Providers System Dump" collect system dumps for one or all providers and merge them into one folder

15. Under "Settings", subsections for Consumer and Provider are available.
   a. "Consumer Settings" contain sections applicable to Consumer UFM specifically (e.g., creation of access tokens for UFM consumer authentication);
   b. "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).
11 UFM Plugins

- rest-rdma Plugin
- NDT Plugin
- UFM Telemetry Fluentd Streaming (TFS) Plugin
- UFM Events Fluent Streaming (EFS) Plugin
- UFM Bright Cluster Integration Plugin
- UFM Cyber-AI Plugin
- Autonomous Link Maintenance (ALM) Plugin
- DTS Plugin
- GRPC-Streamer Plugin
- Sysinfo Plugin
- SNMP Plugin
- Packet Mirroring Collector (PMC) Plugin
- PDR Deterministic Plugin
- GNMI-Telemetry Plugin
- UFM Telemetry Manager (UTM) Plugin
- UFM Consumer Plugin
- Fast-API Plugin

11.1 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.

11.1.1 Deployment Server

11.1.1.1 Deploy Plugin on UFM Appliance

1. Log into your UFM as admin.
2. Enter config mode. Run:

```
enable
config terminal
```

⚠️ 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.


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]`.

### 11.1.1.2 Deploy Plugin on Bare Metal Server

1. Verify that UFM is installed and running.
2. Pull image from docker hub:
   ```bash
docker pull mellanox/ufm-plugin-rest-rdma:[version]
   ```
3. To load image run:
   ```bash
   /opt/ufm/scripts/manage_ufm_plugins.py add -p rest-rdma
   ```

### 11.1.1.3 Deployment Client

Run the following command to pull the image from the docker hub:

```bash
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:

```bash
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:

```bash
docker ps
```

### 11.1.2 How to Run

#### 11.1.2.1 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:

```bash
cd /opt/ufm/src/ufm-plugin-ufm-rest
./ufm_rdma.py -r server
```

#### 11.1.2.2 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
   a. Enter the docker container using `docker exec -it ufm-plugin-rest-rdma bash`
   b. Then, run `cd /opt/ufm/src/ufm-plugin-rest-rdma`
   c. 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:

   ```
   cp <containerId>:/opt/ufm/ufm-plugin-ufm-rest/[script name] /host/path/target
   ```
   
   Example:
   ```
   cp <containerId>:/opt/ufm/ufm-plugin-ufm-rest/ufm-rest-rdma_client.sh /host/path/target
   ```

   a. To see the available options, run:
   ```
   ./ufm-rest-rdma_client.sh -h
   ```

3. Option 3: From hosting server, use the `docker exec` command.

   ```
   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
   ```

11.1.3 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
```

11.1.3.1 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:
Example:

```bash
cp <containerId>:/opt/ufm/ufm-plugin-ufm-rest/[script name] /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]
```

### 11.1.3.2 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
```

**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.

### 11.1.3.3 Rest Request with Username/Password Authentication

To get the UFM version from inside the docker:
To get the UFM version from hosting server using script:

```
./ufm_rdma.py -r client -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": ""}}'
```

11.1.3.4 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
```

11.1.3.5 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
```

11.2 NDT Plugin

11.2.1 Overview

NDT plugin is a self-contained Docker container with REST API support managed by UFM. The NDT plugin introduces the following capabilities:

a. 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.

b. 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.

11.2.2 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.

11.2.3 Authentication

Following authentication types are supported:
• basic (/ufmRest)
• client (/ufmRestV2)
• token (/ufmRestV3)

11.2.4 REST API

The following REST APIs are supported:

11.2.4.1 Topodiff
• GET /help
• GET /version
• POST /upload_metadata
• GET /list
• POST /compare
• POST /cancel
• GET /reports
GET /reports/<report_id>
POST /delete

11.2.4.2 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.

11.2.5 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

11.2.5.1 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:

<table>
<thead>
<tr>
<th>Start Device</th>
<th>Start Port</th>
<th>End Device</th>
<th>End Port</th>
<th>Link Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSM07-0101-0702-01IB0</td>
<td>Port 21</td>
<td>DSM07-0101-0702-01IB1</td>
<td>Blade 2_Port 1/1</td>
<td>Data</td>
</tr>
<tr>
<td>DSM07-0101-0702-01IB0</td>
<td>Port 22</td>
<td>DSM07-0101-0702-01IB1</td>
<td>Blade 2_Port 1/1</td>
<td>Data</td>
</tr>
</tbody>
</table>
### 11.2.5.2 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:

<table>
<thead>
<tr>
<th>Start Device</th>
<th>Start Port</th>
<th>End Device</th>
<th>End Port</th>
<th>Link Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSM07-0101-0702-01IB0</td>
<td>Port 23</td>
<td>DSM07-0101-0702-02IB1</td>
<td>Blade 3_Port 1/1</td>
<td>Data</td>
</tr>
<tr>
<td>DSM09-0101-0617-001IB2</td>
<td>Port 33</td>
<td>DSM09-0101-0721-001IB4</td>
<td>Port 1</td>
<td>Data</td>
</tr>
<tr>
<td>DSM09-0101-0617-001IB2</td>
<td>Port 34</td>
<td>DSM09-0101-0721-001IB4</td>
<td>Port 2</td>
<td>Data</td>
</tr>
<tr>
<td>DSM09-0101-0617-001IB2</td>
<td>Port 35</td>
<td>DSM09-0101-0721-001IB4</td>
<td>Port 3</td>
<td>Data</td>
</tr>
</tbody>
</table>

### 11.2.5.3 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.

11.2.6 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.
11.2.6.1 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.

11.2.6.2 Subnet Merger Flow

**Initial Fabric Topology**

![Initial Fabric Topology Diagram]

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:
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
- 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,3,26,0xf452140300280040,1,Any,Active
```

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,EndDevice,EndPort,EndDeviceLocation,U height_1,LinkType,Speed_2,Cable Length_3,_4,_5,_6,_7,State,Domain
,MFSr:r-ufm-sw111:MQM8700/01,Port 1,,,,,,,,,,,,,,,Disabled,Boundary
,MFSr:r-ufm-sw111:MQM8700/01,Port 30, r-ufm55 mlx5_1,Port 1,,,,,,,,,,,,,,,Active,In-Scope
,MFSr:r-ufm-sw111:MQM8700/01,Port 29, r-ufm55 mlx5_0,Port 1,,,,,,,,,,,,,,,Active,In-Scope
,MFSr:r-ufm-sw111:MQM8700/01,Port 26, r-ufm64 mlx5_0,Port 1,,,,,,,,,,,,,,,Active,In-Scope
,MFSr:r-ufm-sw111:MQM8700/01,Port 3,,,,,,,,,,,,,,,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
```
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,0,0xf452140300280081,1,Any,Active
0xb83fd2030080302e,1,0x98039b0300867bba,1,Any,Active
0xb83fd2030080302e,3,0x98039b0300867bba,1,Any,Active
0xb83fd2030080302e,9,0x98039b0300867bba,1,Any,Active
```

9. Repeat previous steps if need to perform additional setup extension.

11.2.6.3 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
2. **Access “Subnet Merger” to initiate the bring-up wizard.**

3. The wizard will guide you through the process, containing the following steps:
   a. **Upload the initial NDT tab and validate it.**
b. Once you are satisfied with the results of the validation in the previous tab, you can proceed to deploy the file.
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.

11.2.6.4 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.

11.3 UFM Telemetry Fluentd Streaming (TFS) Plugin

11.3.1 Overview
The TFS plugin is designed to extract UFM Telemetry counters from specified telemetry HTTP endpoints and stream them to the designated Fluentd collector destination.

The diagram below illustrates the plugin’s stream flow:

11.3.2 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).

11.3.3 Authentication
The following authentication types are supported:
   - basic (/ufmRest)
   - client (/ufmRestV2)
   - token (/ufmRestV3)

11.3.4 Rest API
The following REST APIs are supported:
11.4 UFM Events Fluent Streaming (EFS) Plugin

11.4.1 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.

11.4.2 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).

11.4.3 Authentication

The following authentication types are supported:

- basic (/ufmRest)
- client (/ufmRestV2)
- token (/ufmRestV3)

11.4.4 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.
11.5 UFM Bright Cluster Integration Plugin

11.5.1 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.

11.5.2 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.

11.5.3 Authentication
The following authentication types are supported:
- basic (/ufmRest)
- client (/ufmRestV2)
- token (/ufmRestV3)

11.5.4 Bright Cluster Integration UI
1. After the successful deployment of the plugin, a new tab is shown under the UFM settings section for bright configurations management:
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

#### 11.5.5 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](#).

#### 11.6 UFM Cyber-AI Plugin

##### 11.6.1 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.

##### 11.6.2 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.

![UFM GUI with Plugins Management Table]

After running the plugin successfully. You should be able to see the Cyber-AI items under the main
11.7 Autonomous Link Maintenance (ALM) Plugin

11.7.1 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.
11.7.2 Schematic Flow

![Diagram showing the flow of Autonomous Link Maintenance (ALM) plugin]

11.7.3 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:
   
   ```bash
   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.
11.7.4 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 7 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_errors_lane2, 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_events, plr_rcv_codes, plr_rcv_code_err, plr_rcv_uncorrectable_code, plr_xmit_codes, plr_xmit_retry_codes, 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, PortRcvRemotePhysicalErrorsExtended, PortRcvErrorsExtended, PortXmitDiscardsExtended, PortRcvSwitchRelayErrorsExtended, PortRcvConstraintErrorsExtended, VL15DroppedExtended, PortXmitWaitExtended, PortXmitDataExtended, PortRcvDataExtended, PortXmitPktsExtended, PortRcvPktsExtended, PortUniCastXmitPktsExtended, PortUniCastRcvPktsExtended, PortMultiCastXmitPktsExtended, PortMultiCastRcvPktsExtended

The supported InfiniBand hardware technologies are HDR, Beta on NDR.
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

11.7.5 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.

<table>
<thead>
<tr>
<th>Name</th>
<th>Section name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mode</td>
<td>Prediction</td>
<td>The mode can be either “active” or “shadow.” In active mode, the ALM will enforce isolation/deisolation rules on all ports which predict to fail 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, and predict to fail.</td>
</tr>
<tr>
<td>Name</td>
<td>Section name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>--------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>except_list</td>
<td>Prediction</td>
<td>Includes the ports that receive the opposite treatment compared to the mode. The expect list saved in location <code>/opt/ufm/files/conf/plugin/alm/predict.csv</code> Format: <code>port_guid,port_number</code> 0x1070fd03001769b4,1 0x1070fd03001769b4,3</td>
</tr>
<tr>
<td>mode</td>
<td>NOC</td>
<td>The mode can be either &quot;active&quot; or &quot;shadow.&quot; In active mode, the ALM will enforce isolation/deisolation rules on all ports that considered as out of nominal condition except those listed in the &quot;expect&quot; list. In shadow mode, the ALM will enforce isolation/deisolation rules on the ports listed in the &quot;except&quot; list.</td>
</tr>
<tr>
<td>except_list</td>
<td>NOC</td>
<td>Includes the ports that receive the opposite treatment compared to the mode. The expect list saved in location <code>/opt/ufm/files/conf/plugin/alm/noc.csv</code> Format: <code>port_guid,port_number</code> 0x1070fd03001769b4,1 0x1070fd03001769b4,3</td>
</tr>
<tr>
<td>max_per_hour</td>
<td>Isolation</td>
<td>The maximum number of ports that can be isolated in an hour</td>
</tr>
<tr>
<td>max_per_week</td>
<td>Isolation</td>
<td>Maximum number of ports that can be isolated in a week</td>
</tr>
<tr>
<td>max_per_month</td>
<td>Isolation</td>
<td>Maximum number of the ports that can be isolated in a month</td>
</tr>
<tr>
<td>min_links_per_switch_pair</td>
<td>Isolation</td>
<td>Minimum links between two switches to perform isolation</td>
</tr>
<tr>
<td>min_active_ports_per_switch</td>
<td>Isolation</td>
<td>Minimum number of active ports per switch before perform isolation</td>
</tr>
<tr>
<td>Deisolation_time</td>
<td>DeIsolation</td>
<td>The waiting time before deisolate the isolated port</td>
</tr>
<tr>
<td>max_per_hour</td>
<td>DeIsolation</td>
<td>The maximum number of deisolated port per hour</td>
</tr>
<tr>
<td>absolute_threshold_of_isolated_ports</td>
<td>Isolation</td>
<td>The maximum number of ports than can be isolated in one sample</td>
</tr>
</tbody>
</table>

### 11.7.6 ALM UI

After the successful deployment of the plugin, a new item is shown in the UFM side menu for the ALM plugin:
11.7.6.1 Current State

This page displays a table presenting the current cluster status, outlining the following counts:

1. Number of ports
2. Number of isolated ports
3. Number of ports in active/shadow prediction mode
4. Number of ports in active/shadow NOC mode
5. Number of ports out of NOC
11.7.6.2 Events Summary

This page displays a table presenting a port count summary, outlining the following counts:

1. Number of isolated ports in the past hour, week, and month for ‘host to switch’ and ‘switch to switch’.
2. Number of de-isolated ports in the past hour, week, and month for ‘host to switch’ and ‘switch to switch’.
3. Number of isolation actions not taken from prediction by ALM in the past hour, week, and month for ‘host to switch’ and ‘switch to switch’.
4. Number of isolation actions not taken from NOC by ALM in the past hour, week, and month for ‘host to switch’ and ‘switch to switch’.
11.7.6.3 Port Level Status
This page displays a table presenting the cluster ports.

11.7.6.4 Model Performance
Different metrics can measure the performance and accuracy of the ALM model, these metrics will be exposed to the user through the table as follows.

11.7.6.5 Configuration
This page displays ALM plugin configuration update method.
The ALM configuration is divided into four sections:
- General Configurations
### Prediction Mode

- **Default Mode**
  - Active
  - Shadow

- **Exception List**
  - Select Ports

### NOC Mode

- **Default Mode**
  - Active
  - Shadow

- **Exception List**
  - Select Ports

### ML Model Configurations
11.7.7 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.

<table>
<thead>
<tr>
<th>ALM Job Name</th>
<th>Description</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port_hist</td>
<td>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.</td>
<td>7 mins</td>
</tr>
<tr>
<td>Low_freq_predict</td>
<td>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.</td>
<td>7 mins</td>
</tr>
</tbody>
</table>

11.8 DTS Plugin

11.8.1 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.

11.8.2 Deployment

11.8.2.1 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:
11.8.2.2 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.
11.8.3 DTS UI

11.8.3.1 Info

11.8.3.2 Health
11.8.3.3 Telemetry

11.8.3.4 Data Sources

11.9 GRPC-Streamer Plugin

11.9.1 Authentication

The following authentication types are supported:

- Basic (/ufmRest)
- Token (/ufmRestV3)
11.9.2 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:

```plaintext
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:

```plaintext
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:

```plaintext
client session --server_ip=server_ip --id=client_id --auth=username,password --token=token
```

11.9.3 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:

```plaintext
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:

```protobuf
def message SessionRespond{
  string respond;
}
```

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).

11.9.4 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:

```protobuf
def message SubscriberParams{
  message APIParams {
    string ufm_api_name = 1;
    int32 interval = 2;
    optional bool only_delta = 3;
  }
  string job_id = 4; //unique identifier for this job
  repeated APIParams apiParams = 5;
}
```

- 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:

```protobuf
def message SessionRespond{
  string respond;
}
```

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.

11.9.5 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:

```protobuf
message ListSubscriberParams{
  repeated SubscriberParams subscribers = 1;
}
```
- Console command: server subscribes --server_ip=server_ip

11.9.6 Delete a Known Subscriber
- Description: Deletes an existing subscriber and removes the session.
- Call: DeleteSubscriber
- Request Content Type: Message gRPCStreamerID
- Request Data:

```protobuf
message gRPCStreamerID{
  string job_id = 1;
}
```
- Response:protobuf.Empty

11.9.7 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:

```protobuf
message gRPCStreamerID{
  string job_id = 1;
}
```
- Response content type:

```protobuf
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:
  
  ```bash
  client once_id --server_ip=server_ip --id=client_id
  ```

11.9.8 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 iterator, 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:
  
  ```bash
  client stream_id --server_ip=server_ip --id=client_id
  ```

11.9.9 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:

```proto
message SubscriberParams {
  message APIParams {
    string ufms_api_name = 1;
    int32 interval = 2;
    optional bool only_delta = 3;
    string job_id = 4; //unique identifier for this job
    repeated APIParams apiParams = 2;
  }
  string job_id = 1; //unique identifier for this job
  repeated gRPCStreamerParams results = 2;
}
```

- Response content type:

```proto
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:

```bash
client once --server_ip=server_ip --id=client_id --auth=username,password --token=token --apis=events;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.

### 11.9.10 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 iterator, where each item is message gRPCStreamerParams. at the end, delete the session.
- Call: RunPeriodically
- Request Content Type: Message SubscriberParams
- Request Data:

```proto
message SubscriberParams {
  message APIParams {
    string ufms_api_name = 1;
    int32 interval = 2;
    optional bool only_delta = 3;
    string job_id = 4; //unique identifier for this job
    repeated APIParams apiParams = 2;
  }
  string job_id = 1; //unique identifier for this job
  repeated gRPCStreamerParams results = 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
11.9.11 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

11.9.12 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:

```protobuf
type gRPCStreamerID{
    string job_id = 1;
}
```
- Response: google.protobuf.Empty

11.9.13 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

```protobuf
type 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
google.protobuf.Bytes 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:
11.9.14 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:

```protobuf
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;
}
```

11.9.14.1 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:

```protobuf
message SessionRespond{
  string respond;
}
```

11.10 Sysinfo Plugin

11.10.1 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.

11.10.2 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)

11.10.3 REST API

The following REST APIs are supported:
- GET /help
- GET /version
- POST /query
- POST /update
- POST /cancel
- POST /delete

11.10.4 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.

11.11 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.

11.11.1 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.

11.11.2 Authentication

The following authentication types are supported:

- basic (ufmRest)
- client (ufmRestV2)
- token (ufmRestV3)

11.11.3 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.

11.11.4 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.

<table>
<thead>
<tr>
<th>OID</th>
<th>Name</th>
<th>Description</th>
<th>Status</th>
<th>Severity</th>
</tr>
</thead>
<tbody>
<tr>
<td>MELLANOX-EFM-MIB::testTrap</td>
<td>send-test</td>
<td>A test trap ordered by the system administrator</td>
<td>Enabled</td>
<td>Warning</td>
</tr>
<tr>
<td>MELLANOX-EFM-MIB::asicChipDown</td>
<td>asic-chip-down</td>
<td>ASIC (Chip) Down</td>
<td>Enabled</td>
<td>Critical</td>
</tr>
<tr>
<td>MELLANOX-EFM-MIB::cpuUtilHigh</td>
<td>cpu-util-high</td>
<td>CPU utilization has risen too high</td>
<td>Enabled</td>
<td>Warning</td>
</tr>
<tr>
<td>OID</td>
<td>Name</td>
<td>Description</td>
<td>Status</td>
<td>Severity</td>
</tr>
<tr>
<td>-----------------</td>
<td>-----------------------------</td>
<td>-------------------------------------------------------</td>
<td>---------</td>
<td>----------</td>
</tr>
<tr>
<td>MELLANOX-EFM-MIB::diskSpaceLow</td>
<td>disk-space-low</td>
<td>Filesystem free space has fallen too low</td>
<td>Enabled</td>
<td>Warning</td>
</tr>
<tr>
<td>MELLANOX-EFM-MIB::expectedShutdown</td>
<td>expected-shutdown</td>
<td>Expected system shutdown</td>
<td>Enabled</td>
<td>Info</td>
</tr>
<tr>
<td>MELLANOX-EFM-MIB::systemHealthStatus</td>
<td>health-module-status</td>
<td>Health module Status</td>
<td>Enabled</td>
<td>Critical</td>
</tr>
<tr>
<td>MELLANOX-EFM-MIB::insufficientFans</td>
<td>insufficient-fans</td>
<td>Insufficient amount of fans in system</td>
<td>Enabled</td>
<td>Warning</td>
</tr>
<tr>
<td>MELLANOX-EFM-MIB::insufficientFansRecover</td>
<td>insufficient-fans-recover</td>
<td>Insufficient amount of fans in system recovered</td>
<td>Enabled</td>
<td>Info</td>
</tr>
<tr>
<td>MELLANOX-EFM-MIB::insufficientPower</td>
<td>insufficient-power</td>
<td>Insufficient power supply</td>
<td>Enabled</td>
<td>Warning</td>
</tr>
<tr>
<td>RFC1213::linkdown</td>
<td>interface-down</td>
<td>An interface's link state has changed to down</td>
<td>Enabled</td>
<td>Minor</td>
</tr>
<tr>
<td>RFC1213::linkup</td>
<td>interface-up</td>
<td>An interface's link state has changed to up</td>
<td>Enabled</td>
<td>Info</td>
</tr>
<tr>
<td>MELLANOX-EFM-MIB::unexpectedShutdown</td>
<td>unexpected-shutdown</td>
<td>Unexpected system shutdown</td>
<td>Enabled</td>
<td>Minor</td>
</tr>
<tr>
<td>SNMPv2-MIB::coldStart</td>
<td>cold-start</td>
<td>SNMP entity reinitialized</td>
<td>Enabled</td>
<td>Info</td>
</tr>
</tbody>
</table>

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:
11.11.5 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](#).

11.12 Packet Mirroring Collector (PMC) Plugin

11.12.1 Overview

The Packet Mirroring Collector/Controller plugin facilitates the configuration capture and display of mirroring on a variety of triggers, enabling users to conduct real-time monitoring of network events. Supported triggers are pFRN, Congestion, Fast Recovery, CQE and PHY Error Links.

11.12.2 Deployment

11.12.2.1 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.

11.12.3 PMC UI

11.12.3.1 Network Event Collector Display
11.12.3.2 Network Event Collector Configuration

11.12.3.2.1 pFRN:
- pFRN Notifications - Enables/Disables mirroring on pFRN trigger for entire network or list of GUIDs

11.12.3.2.2 Fast Recovery:
- Fast Recovery Notifications - Enables/Disables mirroring on Fast Recovery trigger for entire network or list of GUIDs
- Notifications Level - Specifies threshold for Fast Recovery mirroring. (Thresholds are configured in SM configuration)

11.12.3.2.3 PHY Error Links:
- PHY Error Links Notifications - Enables/Disables mirroring on PHY Link Error trigger for entire network or list of GUIDs
- Specifies threshold for PHY Link Error mirroring. (Thresholds are configured in SM configuration)
11.12.3.2.4 CQE:
- CQE Notifications - Enables/Disables mirroring on CQE Notifications trigger for entire network or list of GUIDs

11.12.3.2.5 Congestion:
- Congestion Notifications - Enables/Disables mirroring on Congestion Notifications trigger for entire network or list of GUIDs
  - Mirrored packets (%) - Specifies the percent of congested packets to be mirrored.
  - High threshold - High threshold percentage for InfiniBand switch egress port queue size. Values are in the [1,1023] range.
  - Low threshold - Low threshold percentage for InfiniBand switch egress port queue size. Values are in the [1,1023] range.

NOTE: When a packet enters an InfiniBand switch, its data is stored at an ingress port buffer. A pointer to the packet's data is inserted into the egress port's queue, from which the packet will be exiting the switch. At that point, the threshold given by this command line argument is compared to the egress queue data size. If the queue data size exceeds the threshold, a congestion event is reported. The threshold is given in percent of the ingress port size.

An egress port queue can point data coming from multiple ingress port buffers, therefore the threshold can be bigger than 100%.

11.13 PDR Deterministic Plugin

11.13.1 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.

### 11.13.2 Schematic Flow

![Schematic Flow Diagram](image)

### 11.13.3 Deployment

To deploy the plugin, follow these steps:

1. Download the ufm-plugin-pdr_deterministic-image from the Docker Hub.
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:
   a. Log in to the UFM server terminal.
   b. Run:

   ```
   docker load -i <path_to_image>
   ```
   
   c. 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.

11.13.4 Isolation Decisions

11.13.4.1 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 \( S_{12m} = (12 + Pm + 1)/Pm \) and \( S_{125m} = (125 + Pm + 1)/Pm \). \( 12m_{\text{thd}} = \text{LinkBW}_{\text{Gbps}} \times 10^9 \times 12 \times 60 \times 10^{-14} \) (2.88 for NDR) and \( 125m_{\text{thd}} = \text{LinkBW}_{\text{Gbps}} \times 10^9 \times 125 \times 60 \times 10^{-15} \) (3 for NDR).

Check the following conditions for every port in the given set:

1. If the \( \Delta (\text{LinkDownedCounterExt}) \) port is > 0 and the \( \Delta (\text{LinkDownedCounterExt}) \) remote port is > 0, add it to the list of \( \text{bad}_\text{ports} \). This condition should be ignored if the \(--\text{no\_down\_count}\) flag is provided.
2. If the \( \text{symbol}\_\text{errors}[\text{now}\_\text{idx}] \text{ - symbol}\_\text{errors}[\text{now}\_\text{idx} - S_{12m}] \) is > \( 12m_{\text{thd}} \), add the link to the list of \( \text{bad}_\text{ports} \), and continue with next link.
3. If the \( \text{symbol}\_\text{errors}[\text{now}\_\text{idx}] \text{ - symbol}\_\text{errors}[\text{now}\_\text{idx} - S_{125m}] \) is > \( 125m_{\text{thd}} \), add the link to the list of \( \text{bad}_\text{ports} \), continue with next link.

Packet 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.

11.13.4.2 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.

11.13.4.3 Configuration

The following parameters are configurable via the plugin’s configuration file. *(pdr_deterministic.conf)*

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>INTERVAL</td>
<td>Interval for requesting telemetry counters, in seconds.</td>
<td>300</td>
</tr>
<tr>
<td>MAX_NUM_ISOLATE</td>
<td>Maximum ports to be isolated. max(MAX_NUM_ISOLATE, 0.5% * fabric_size)</td>
<td>10</td>
</tr>
<tr>
<td>TMAX</td>
<td>Maximum temperature threshold</td>
<td>70 (Celsius)</td>
</tr>
<tr>
<td>D_TMAX</td>
<td>Maximum allowed Temperature Delta</td>
<td>10</td>
</tr>
<tr>
<td>MAX_PDR</td>
<td>Maximum allowed packet drop rate</td>
<td>1e-12</td>
</tr>
<tr>
<td>CONFIGURED_BER_CHECK</td>
<td>If set to true, the plugin will isolate based on BER calculations</td>
<td>True</td>
</tr>
<tr>
<td>CONFIGURED_TEMP_CHECK</td>
<td>If set to true, the plugin will isolate based on temperature measurements</td>
<td>True</td>
</tr>
<tr>
<td>LINK_DOWN_ISOLATION</td>
<td>If set to true, the plugin will isolate based on LinkDownedCounterExt measurements</td>
<td>False</td>
</tr>
<tr>
<td>SWITCH_TO_HOST_ISOLATION</td>
<td>If set to true, the plugin will isolate ports connected via access link</td>
<td>False</td>
</tr>
<tr>
<td>DRY_RUN</td>
<td>Isolation decisions will be only logged and will not take effect</td>
<td>False</td>
</tr>
<tr>
<td>DEISOLATE_CONSIDER_TIME</td>
<td>Consideration time for port de-isolation (in minutes)</td>
<td>5</td>
</tr>
<tr>
<td>DO_DEISOLATION</td>
<td>If set to false, the plugin will not perform de-isolation</td>
<td>True</td>
</tr>
<tr>
<td>DYNAMIC_WAIT_TIME</td>
<td>Seconds to wait for the dynamic telemetry session to respond</td>
<td>30</td>
</tr>
</tbody>
</table>

11.13.4.4 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:
\[ \text{seconds} = \frac{\text{max\_BER\_target}^{-1}}{\text{min\_port\_rate}} \]

Example:

<table>
<thead>
<tr>
<th>Rate</th>
<th>BER Target</th>
<th>Minimum Bits</th>
<th>Minimum Time in Seconds</th>
<th>In Minutes</th>
</tr>
</thead>
<tbody>
<tr>
<td>HDR</td>
<td>2.00E+11</td>
<td>1.00E-12</td>
<td>1.00E+12</td>
<td>5</td>
</tr>
<tr>
<td>HDR</td>
<td>2.00E+11</td>
<td>1.00E-13</td>
<td>1.00E+13</td>
<td>50</td>
</tr>
<tr>
<td>HDR</td>
<td>2.00E+11</td>
<td>1.00E-14</td>
<td>1.00E+14</td>
<td>500</td>
</tr>
<tr>
<td>HDR</td>
<td>2.00E+11</td>
<td>1.00E-16</td>
<td>1.00E+16</td>
<td>50000</td>
</tr>
</tbody>
</table>

BER counters are calculated with the following formula:

\[
BER = \frac{\text{error bits}_i - \text{error bits}_{i-1}}{\text{total bits}_i - \text{total bits}_{i-1}} = \frac{\text{error bits}_i - \text{error bits}_{i-1}}{\text{Link data rate} \times (\text{time}_i - \text{time}_{i-1})}
\]

11.13.4.5 Ports Exclusion List

You can designate specific ports to be excluded from PDR analysis, isolation, or de-isolation for an indefinite or limited period. Already excluded ports can also be removed from this list.

Ports are added to or removed from the exclusion list via the PDR plugin’s REST API.

To add ports to the exclusion list (to be excluded from analysis), run:

```bash
curl -k -i -u <user:password> -X PUT 'https://<host_ip>/ufmRest/plugin/pdr_deterministic/excluded' -d \\
'\[[<formatted_ports_list>]\]' -H "Content-Type: application/json"
```

Optionally, you can specify a TTL (time to live in the exclusion list) following the port after the comma. If zero or not specified, the port is excluded. For example:

```
-d '\[["9c0591030085ac80_45"],("9c0591030085ac80_46",300)\]'
```

To remove ports from the exclusion list:

```bash
curl -k -i -u <user:password> -X DELETE 'https://<host_ip>/ufmRest/plugin/pdr_deterministic/excluded' -d \\
'\[<comma_separated_port_names>]\]' -H "Content-Type: application/json"
```

Example:

```
-d '\["9c0591030085ac80_45","9c0591030085ac80_46"\]'
```

To retrieve ports and their remaining exclusion times from the exclusion list:

```bash
curl -k -i -u <user:password> -X GET 'https://<host_ip>/ufmRest/plugin/pdr_deterministic/excluded'
```
11.14 GNMI-Telemetry Plugin

The GNMI Telemetry Plugin is 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.

11.14.1 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, changing the UFM inventory location to a different machine in the config file is possible. To do so, token access to that machine is necessary.

11.14.2 Secure Server

The server can be secured by using certificates. To secure the server, the "secure_mode_enabled" flag need to be set to "true" in the configuration, on default, the server security is configured to be true.

Upon initialization, the gNMI server retrieves the UFM certificates from the /opt/ufm/conf/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 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. In addition, the server requires that the client certifications naming convention is aligned with the DNS name (SAN) as the UFM.
11.14.3 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
```

11.14.4 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_counters/hist0
```

The request retrieves data from node_guid **0x5255456**, specifically in port number 2, with the request counter set to hist0.

Example 2:

```
gnmic -a localhost:9339 --insecure get --path nvidia/ib/guid[*]/port[port_number=*]/amber/port_counters/hist0
```

The request retrieves the data from all the ports and the node_guids, with the request counter set to hist0.

Example 3:

```
gnmic -a localhost:9339 --insecure get --path nvidia/ib/guid[guid=0x5255456]/port[port_number=2]/amber/*
```

The request retrieves the data from node_guid **0x5255456**, port 2, with the request counters set to "all".

11.14.5 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:

```
$ gnic -a localhost:9339 --insecure sub --path nvidia/ib/guid[guid=0x5255456]/port[port_number=2]/amber/port_counters/hist0 -i 30s
```

This request retrieves data from the node_guid `0x5255456`, port 2, where the request counter is hist0, and the interval is configured for 30 seconds. If the user wishes to test the stream, the stream mode can be configured to "once," and following a single response, the stream will be stopped.

Example:

```
$ gnic -a localhost:9339 --insecure sub --path nvidia/ib/guid[guid=0x5255456]/port[port_number=2]/amber/port_counters/hist0 -i 30s --mode once
```

This request retrieves the data from node_guid `0x5255456`, port 2, where the request counter is hist0. The stream shuts down after one response, similar to a GET request.

### 11.14.6 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. On-change requests contain inventory and event paths as well.

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:

```
$ gnic -a localhost:9339 --insecure sub --path nvidia/ib/guid[guid=0x5255456]/port[port_number=2]/amber/port_counters/hist0 --stream-mode on-change --heartbeat-interval 1m
```

This request retrieves data from node_guid `0x5255456`, port 2, with the request counters set to hist0. It periodically checks for changes every minute, and when changes are detected, it promptly sends the updated values.

Example:

```
$ gnic -a localhost:9339 --insecure sub --path nvidia/ib/guid[guid=0x5255456]/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.
The below is an example of the response to a particular GUID, which represents an on-change request for a few counters. However, only specific counters have been updated, those who have not updated have a value of 0. Because the flag `include_old_data_on_change` default is true

```
1786532307824,0x0002c903007e5220,1,0,0,0,41447490564,617155163,24184739,17,0,0,0,0
```

The same example with the flag set to true will give this:

```
1786532307824,0x0002c903007e5220,1,,,,,41447490564,617155163,24184739,17,,,,
```

Only the values that have changed return while the others are empty values. To get this format of data, one need to change the `include_old_data_on_change` in the config file to false.

### 11.14.7 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:**

```
gnmc -a localhost:9339 --insecure sub --path nvidia/ib/amber/reply/sample
```

The second message on the headers will be set to hash values.

### 11.14.8 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.

Moreover, inventory includes the size of the telemetry, which is not always the same as the active ports. 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`

---

398
Example:
```
gmic -a localhost:9339 --insecure get -p 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,
        "TelemetrySize": 4,
        "timestamp": 1698824211535069000
      }
    }
  }]
}
```

### 11.14.9 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:
```
gmic -a localhost:9339 --insecure get -p 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": "id",
          "object_name": "object_name",
          "write_to_syslog": "write_to_syslog",
          "description": "description",
          "type": "type",
          "event_type": "event_type",
          "severity": "severity",
          "timestamp": "timestamp",
          "counter": "counter",
          "category": "category",
          "object_path": "object_path",
          "object_name": "name"
        }],
        "Values": [{
          "id": 7718,
          "object_name": "Grid",
          "write_to_syslog": false,
          "description": "Disk space usage in /opt/ufm/files/log is above the threshold of 90.0%.",
          "type": "Grid",
          "event_type": "Critical",
          "severity": "Critical",
          "timestamp": "2023-11-01 07:25:54",
          "counter": "N/A",
          "category": "Maintenance",
          "object_path": "Grid",
          "object_name": "Disk utilization threshold reached"
        },
        "id": 7717,
        "object_name": "Grid",
        "write_to_syslog": false,
        "description": "Disk space usage in /opt/ufm/files/log is above the threshold of 90.0%.",
        "type": "Grid",
        "event_type": "Critical",
        "severity": "Critical",
        "timestamp": "2023-11-01 07:24:54",
        "counter": "N/A",
        "category": "Maintenance",
        "object_path": "Grid",
        "object_name": "Disk utilization threshold reached"
      }
    }
  }]
}
```

### 11.15 UFM Telemetry Manager (UTM) Plugin

#### 11.15.1 Overview

Managed telemetry is a mode of high availability and improved performance of UFM Telemetry processes.
Governed by UFM Telemetry Manager (UTM) several UFM Telemetry Instances (TIs) run on one or more machines, each collecting a subset of the cluster fabric.

UTM manages the following aspects:
- monitoring of TI states: down, initializing, running, paused
- TI management commands: add, remove, pause, start, restart
- partitioning of fabric based on TIs health and fabric changes
- assigning fabric segments to TIs
- telemetry coverage check of a cluster

The UFM Telemetry Manager (UTM) Plugin facilitates managed telemetry in high availability mode, enhancing the performance of UFM Telemetry operations.

Under the governance of UFM Telemetry Manager (UTM), multiple UFM Telemetry Instances (TIs) are executed on one or more machines, with each TI responsible for collecting a specific portion of the cluster fabric.

Key functionalities managed by UTM include:
- Monitoring TI statuses: down, initializing, running, paused
- Execution of TI management commands: add, remove, pause, start, restart
- Fabric partitioning based on TI health and fabric changes
- Assigning fabric segments to TIs
- Verification of telemetry coverage across the cluster

11.15.2 Deployment

As a first step, get the UTM image:
Get UTM image

docker pull mellanox/utm-plugin-utm

The UTM plugin is designed to operate either as a UFM plugin or independently or in standalone mode.

In both setups, it is advisable to utilize UTM deployment scripts. These scripts streamline the process by enabling the deployment or cleanup of the entire setup with just a single command. This includes UTM, host TIs, and preparation of the Switch Telemetry image.

11.15.2.1 UTM Deployment Scripts

Get deployment scripts and examples by mounting the local folder `UTM_DEPLOYMENT_SCRIPTS` (`/tmp/utm_deployment_scripts` in this example) and running `get_deployment_scripts.sh`:

```
$ export UTM_DEPLOYMENT_SCRIPTS=/tmp/utm_deployment_scripts
$ docker run -v "$UTM_DEPLOYMENT_SCRIPTS:/deployment_scripts" --rm --name utm-deployment-scripts -ti mellanox/utm-plugin-utm:latest /bin/sh /get_deployment_scripts.sh
```

The content of the script folder consists of:

- **Examples** - Contains run/stop scripts for both standalone and UFM plugin modes. Each example script is an example of actual deployment script usage.
- **hostlist.txt** - Specifies the hosts, ports, and HCAs for TIs to be deployed
- **Scripts** - Contains actual deployment scripts. Entry-point script `deploy Managed telemetry.sh` triggers the rest two scripts, depending on input arguments.

```
$ cd $UTM_DEPLOYMENT_SCRIPTS
$ tree
   .
   ├── examples
   │   ├── run_standalone.sh
   │   └── run_with_plugin.sh
   │
   │   ├── stop_standalone.sh
   │   └── stop_with_plugin.sh
   │
   | hostlist.txt
   | README.md
   | scripts
   |   └── deploy bringup.sh
   |   └── deploy Managed telemetry.sh
   |   └── deploy ufm telemetry.sh
```

⚠️ All example/deployment scripts should run from the `UTM_DEPLOYMENT_SCRIPTS` folder.
11.15.2.1.1 Hostlist File

Please note the following:

- The hostlist.txt file should be set before running any script.
- The hostname and port will be used for communication and HCA for telemetry collection.
- To ensure optimal functionality, UTM only supports a single fabric for managed TIs, even if different HCAs on the same machine are connected to different fabrics.
- Both local and remote hosts are supported for TI deployments.

```
$ cat hostlist.txt

# List lines in the following format:
# host:port:hca
#
# where:
#  - host is IP or hostname. Use localhost or 127.0.0.1 for local deployment
#  - port to run telemetry on.
#  - hca is the target host device from which telemetry collects. Run `ssh $host ibstat`
#    to find the active device on the target host.
=localhost:8123:mlx5_0
=localhost:8124:mlx5_0
```

11.15.2.1.2 Main Deployment Script

For a more customizable setup beyond what the example scripts offer, users have the option to manually run `./scripts/deploy_managed_telemetry.sh`. This primary deployment script can deploy multiple TIs and optionally UTM as well.

Use `deploy_managed_telemetry.sh --help` to get help.

```
./deploy_managed_telemetry.sh --help
./deploy_managed_telemetry.sh options: mandatory:
    --hostlist-file= Path to a file that lists hostname:port:hca lines
mandatory run options (use only one at the same time):
    -r, --run Deploy and run managed telemetry setup
    -s, --stop Stop all processes and cleanup
mandatory telemetry deployment options (use only one at the same time):
    -t=, --ufmt-image= UFM telemetry docker image or tar.gz/image
     or:
    --bringup-package= Bringup tar.gz package
     according hostlist file
    --utm-image= UTM docker image or tar.gz/image. Runs UTM only if it is set. Configures UTM
    --utm-as-plugin= If UTM runs as a plugin, set this flag
    --data-root= Root directory for run data. | Default: '/tmp/managed_telemetry/'
    --switch-telem-image= Switch telemetry image [tar.gz-file or docker image]. UTM will be able to deploy it
    --common-data-dir= Common data folder for TIs
   -h, --help Print this message
```
11.15.2.2 UFM Plugin Mode

1. Upload the UTM Docker image to the Docker registry on the machine running UFM Enterprise.
2. Navigate to the UFM web UI and click on “Settings” in the left panel.
3. Go to the “Plugin Management” tab.
4. Right-click on the UTM plugin row and select “Add.”

5. Go to the option on the left called "Telemetry Status" to see the UTM UI page.
6. Prepare TI setup using utm_deployment_scripts example scripts:
   a. Change directory:
      ```
      cd $UTM_DEPLOYMENT_SCRIPTS
      ```
   b. Open and configure hostlist.txt
   c. Deploy and run TIs according to hostlist.txt and set these TIs to be monitored by UTM:
      ```
      sudo ./examples/run_with_plugin.sh
      ```
   d. To stop and cleanup TIs setup and unset TIs to be monitored by UTM:
      ```
      sudo ./examples/stop_with_plugin.sh
      ```
      
      This script does not stop UTM plugin!
To stop the UTM plugin, go to “Plugin Management”, right-click on the UTM plugin line and click on disable.

11.15.2.2.1 Default UFM Telemetry Monitoring

UFM Telemetry has high and low-frequency (Primary and Secondary, respectively) TIs that are running by default.

To enable meaningful monitoring:

1. Set `plugin_env_CLX_EXPORT_API_SHOW_STATISTICS=1` in the config files:

   ```
   deployment help
   ```

   ```
   /opt/ufm/files/conf/telemetry_defaults/launch_ibdiagnet_config.ini
   /opt/ufm/files/conf/secondary_telemetry_defaults/launch_ibdiagnet_config.ini
   ```

2. Restart telemetry instances with the new config. If UFM Enterprise runs as a docker container, this command should be executed inside the container.

   ```
   deployment help
   ```

   ```
   /etc/init.d/ufmd ufm_telemetry_restart
   ```

3. Give TIs some time to update performance metrics. The time depends on the update interval of default TIs.

11.15.2.3 Standalone Mode

In standalone mode, UTM periodically tracks fabric changes by itself and does not require UFM Enterprise.

Deploy via example scripts:

1. Change directory

   ```
   cd $UTM_DEPLOYMENT_SCRIPTS
   ```

2. Open and configure `hostlist.txt`

3. Deploy and run TIs according to `hostlist.txt` and run UTM:

   ```
   sudo ./examples/run_standalone.sh
   ```

4. To stop and cleanup TIs setup and UTM, run:

   ```
   sudo ./examples/stop_standalone.sh
   ```

11.15.2.4 Manual Deployment

This section provides detailed instructions for manually deploying UTM and managed TIs to ensure coverage of all potential corner cases where the convenience script may not be effective.
11.15.2.4.1 UTM Deployment

UTM can be started with two docker run commands.

1. Set `utm_config`, `utm_data`, `utm_log`, and `utm_image` variables.
2. Initialize UTM config:

   ```bash
   Initialize UTM
   
   docker run -v $utm_config:/config 
   -v $utm_data:/data 
   --name utm-init 
   --device=/dev/infiniband/ 
   $utm_image/init.sh
   ```

3. Run UTM:

   ```bash
   Run UTM
   
   docker run -d --net=host 
   --security-opt seccomp=unconfined --cap-add=SYS_ADMIN 
   --device=/dev/infiniband/ 
   -v $utm_config:/config 
   -v $utm_data:/data 
   -v $utm_log:/log 
   --name utm $utm_image
   ```

11.15.2.4.2 Managed/Standalone TIs Manual deployment

TI can be represented either as a UFM Telemetry docker container or as a UFM Telemetry Bring-up package.

To run the docker container in managed mode, `launch_ibdiagnet_config.ini` should have the following flags enabled:

```bash
TI docker config

plugin_env_CLX_EXPORT_API_SHOW_STATISTICS=1
plugin_env_UFM_TELEMETRY_MANAGED_MODE=1
```

To run UFM Telemetry with Distributed Telemetry, enable its receiver and specify `HCA` to work on:

```bash
TI docker config

plugin_env_CLX_EXPORT_API_RUN_DT_RECEIVER=1
plugin_env_CLX_EXPORT_API_DT_RECEIVER_HCA=$HCA
```

To run bringup in managed mode, create `enableManaged.ini` file with the same flags and use `custom_config` option of `collection_start`:
11.15.2.5 UTM Configuration File

The UTM configuration file `utm_config.ini` is placed under the configuration folder (which is referred to as `UTM_CONFIG` later on this document).

In the case of UFM plugin mode, `UTM_CONFIG = /opt/ufm/files/conf/plugins/utm/`.

In the case of standalone mode, the default value is `UTM_CONFIG = /tmp/managed_telemetry/utm/config` and can be changed via `--data-root` argument of deployment script.

When changes are made to the configuration file, UTM initiates a restart of its main process to implement the updated configuration.

Users may wish to adjust timeout and update rate configurations based on their specific setups. However, it is important to note that the remaining configurations are tailored to enable UTM to function as a UFM plugin and should not be modified.

11.15.2.5.1 Distributed Telemetry

To enable distributed telemetry set `dt_enable=1` in the corresponding section.

⚠️ Distributed Telemetry requires Switch Telemetry docker image tagged as `switch-telemetry:{version}` and placed under `$UTM_CONFIG/telem_files/` as `switch-telemetry_{version}.tar.gz`

UTM scans this file at its start.

Example deployment scripts handle it for both UFM plugin and standalone modes.

For more details refer to NVIDIA UFM Telemetry Documentation → Distributed Telemetry - Switch Telemetry Agent

11.15.3 GUI

To access the GUI within the UFM web UI, navigate to the Telemetry status section in the left panel.

The UI is accessible whether it is running as a part of UFM Enterprise or standalone via the endpoint: `http://127.0.0.1:8888/files/index.html`.

The GUI comprises several zones:
The top pane displays general information and provides options to add a server name/IP and port for monitoring. Users can set the GUI refresh interval in the top right corner.

The middle panes showcase TI groups, with the default group being basic. Unmanaged (standalone) TIs can be monitored and are placed in the "Unmanaged" group.

Each group pane presents monitoring information for each TI.

The bottom pane exhibits system events. Utilize the bottom right menu to navigate through the events history.

11.15.3.1 TI Management

In managed mode, UTM has the capability to dispatch commands to TIs. By right-clicking on the TI line, users can:

- Pause a currently running TI. This action redistributes fabric sharding among the active TIs.
- Resume a paused TI.
- Exclude a TI from monitoring. Although the TI remains on the machine, it enters a paused state and is removed from its group. It’s important to note that empty TI groups are automatically removed.

11.15.3.2 Telemetry status fields

The table below lists each column of a Telemetry Group panel:

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>URL</td>
<td>TI URL in format http://{hostname}:{port} or http://{IP}:{port}</td>
</tr>
<tr>
<td>Mode</td>
<td>standalone or managed / platform</td>
</tr>
<tr>
<td>DT receiver</td>
<td>With or without a Distributed Telemetry receiver. If 0, cannot receive DT data from a switch TI</td>
</tr>
<tr>
<td>Status</td>
<td>Down, Running, Initializing, Paused, or Restarting</td>
</tr>
<tr>
<td>Field Name</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Uptime</td>
<td>TI uptime in human-readable format</td>
</tr>
<tr>
<td>Collected host/switch ports</td>
<td>Ports collected from the host/switch. By default data that did not change from the last sample is not being re-exported. Such data is shown in the host part ad +num_old_ports. In the screenshot above, first TI of the “unmanaged” group sampled 0 new data samples and found 35 old ones. Nothing is being sampled from Distributed Telemetry, because this TI runs without DT receiver. The resulting format is: 0+35/-</td>
</tr>
<tr>
<td>Configured host/switch ports</td>
<td>Ports configured to be sampled from a host and corresponding switches in total. For more details refer to Distributed Telemetry documentation.</td>
</tr>
<tr>
<td>Enabled/discovered ports</td>
<td>Enabled and discovered ports of the Fabric.</td>
</tr>
<tr>
<td>Iteration time</td>
<td>Total iteration time of UFM Telemetry data collection</td>
</tr>
<tr>
<td>Export time</td>
<td>Export time in the last iteration of UFM Telemetry data collection. Included to Iteration time</td>
</tr>
<tr>
<td>Port counters time</td>
<td>Time spent only on port counters telemetry collection. Included to Iteration time</td>
</tr>
<tr>
<td>Ports/sec</td>
<td>Speed of new port counters data collection during the last iteration of UFM Telemetry.</td>
</tr>
</tbody>
</table>

11.15.4 REST API

All the GUI features including TI management and monitoring can be accessed via REST API.

To access the REST API command list in UFM plugin mode via UFM proxy:

Standalone UTM help

```bash
curl -s -k https://{UFM_HOST_IP}/ufmRest/plugin/utm/help -u {user}:{pass}
```

By default, UTM runs on port 8888. To access the command list in standalone mode directly list use:

Standalone UTM help

```bash
curl -s http://127.0.0.1:8888/help
```

- Get the status of the monitored TIs in JSON format:
11.16 UFM Consumer Plugin

11.16.1 Overview

The UFM consumer plugin is a self-contained Docker container with REST API support, under UFM management. It serves as a Multi-Subnet consumer within UFM, offering all the functionalities available for Multi-Subnet UFM.

The Multi-Subnet UFM feature allows the management of large fabrics spanning multiple sites within a single product, specifically Multi-Subnet UFM.

This feature comprises two layers: UFM Multi-Subnet Provider and UFM Multi-Subnet Consumer (UFM consumer plugin).
The UFM Provider acts as a Multi-Subnet Provider, exposing all local InfiniBand fabric information to the UFM consumer plugin. On the other hand, the UFM consumer plugin functions as a Multi-Subnet Consumer, gathering and consolidating data from configured UFM Providers. This enables users to manage multiple sites conveniently in one location. While the UFM Consumer offers similar functionality to regular UFM, there are behavioral differences concerning aggregation.

Upon deployment of the UFM consumer plugin, an additional tab is incorporated into the UFM interface.

![UFM Enterprise Interface](image)

After clicking on UFM consumer plugin UI, a new browser tab with UFM Multi-Subnet consumer functionality options is presented. Upon the initial launch of the UFM consumer plugin UI and in case there are no configured providers, a screen for adding providers (Provider Management) will be displayed.
The functionality of the UFM consumer plugin is similar to that of Multi-Subnet UFM. For further details, refer to Multi-Subnet UFM.

11.17 Fast-API Plugin

11.17.1 Overview

The Fast-API plugin is a new component that runs in parallel to the UFM model and provides a more scalable and efficient way to manage the inventory and topology of the fabric. It uses a Redis database to store the data it receives from the SM client consumer and the ibdilagnet collector. It also exposes REST APIs for querying the ports, systems, links, switches, and sharp reservations. The Fast-API plugin also supports persistent inventory, which means it keeps track of all the devices, ports, and links the UFM ever encountered, even if they are disabled or removed. The Fast-API plugin is designed to be backward compatible with the UFM model, except for some minor differences in the API parameters and the handling of generic node names.

11.17.2 Deployment

- As a first step, get the Fast API image:

  ```bash
docker pull mellanox/ufm-plugin-fast_api
  ```
• Load the downloaded image onto the UFM server. To do this, use the UFM GUI and navigate to the Settings -> Plugins Management tab.
• After running the plugin successfully, you should be able to see the Fast-API items under the main UFM navigation menu.

11.17.3 Sequence Diagram

11.17.4 Supported APIs

The following APIs are supported:

Ports API - Retrieve all the ports in the cluster or filter by the system name. The following attributes are retrieved:

• GET https://<UFM_IP>/ufmRestV2/plugin/fast_api/resources/ports
• GET /resources/ports/ - deviceId referred as sys_guid for switches and hostname for hosts
- **name**: The name of the port.
- **lid**: The local ID of the port.
- **logical_state**: The logical state of the port.
- **physical_state**: The physical state of the port.
- **number**: The number of the port.
- **decimal_guid**: The decimal GUID of the port.
- **mtu**: The maximum transmission unit of the port.
- **active_speed**: The active speed of the port.
- **active_width**: The active width of the port.
- **peer_address**: The peer address of the port.
- **peer_port**: The peer port of the port.
- **tier**: The tier of the port.
- **label**: The label of the port.

This API retrieves all the ports in the cluster, including cable information. The following attributes are supported for cable information:

- **GET** [https://<UFM_IP>/ufmRestV2/plugin/fast_api/resources/ports/?cable_info=true](https://<UFM_IP>/ufmRestV2/plugin/fast_api/resources/ports/?cable_info=true)
  - **part_number**: The part number of the port.
  - **serial_number**: The serial number of the port.
  - **revision**: The revision of the port.
  - **identifier**: The identifier of the port.
  - **length**: The length of the port.
  - **technology**: The technology of the port.
  - **fw_version**: The firmware version of the port.

Links API - Retrieve all the links in the cluster. The following attributes are retrieved:

- **GET** [https://<UFM_IP>/ufmRestV2/plugin/fast_api/resources/links](https://<UFM_IP>/ufmRestV2/plugin/fast_api/resources/links)
  - **source_guid**: The GUID of the source port.
  - **source_port**: The source port.
  - **destination_guid**: The GUID of the destination port.
  - **destination_port**: The destination port.
  - **source_port_dname**: The display name of the source port.
  - **source_port_name**: The name of the source port.
  - **destination_port_dname**: The display name of the destination port.
  - **destination_port_name**: The name of the destination port.
  - **width**: The width of the link.
  - **severity**: The severity of the link.
  - **source_port_node_description**: The description of the node connected to the source port.
  - **destination_port_node_description**: The description of the node connected to the destination port.
  - **name**: The name of the link.
  - **active**: The status of the link.
This API retrieves information about the systems in the cluster. It can retrieve all the systems or filter by the system name. The following attributes are retrieved:

- **GET** https://<UFM_IP>/ufmRestV2/plugin/fast_api/resources/systems
- **GET** https://<UFM_IP>/ufmRestV2/plugin/fast_api/resources/systems/<system_name>
- **GET** https://<UFM_IP>/ufmRestV2/plugin/fast_api/resources/switches

  - **name**: The display name of the system.
  - **system_name**: The name of the system.
  - **guid**: The GUID of the system.
  - **sys_guid**: The system GUID.
  - **ports**: A list of ports associated with the system.
  - **hostname**: The hostname of the system.
  - **vendor**: The vendor of the system.
  - **temperature**: The temperature of the system.
  - **fw_version**: The firmware version of the system.
  - **technology**: The technology used by the system.
  - **level**: The level of the system.
  - **uptime**: The uptime of the system.
  - **sw_version**: The software version of the system.
  - **system_type**: The type of the system.
  - **description**: The description of the system.
  - **psid**: The PSID of the system.
  - **active**: The status of the system.
  - **state**: The state of the system.
  - **is_managed**: Indicates whether the system is managed.
12 Troubleshooting

12.1 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.

12.2 Performing Failover on Non-Master Node

The `ufm_ha_cluster` failover action fails with the following error: "Cannot perform failover on non-master node". To fix, follow the below action:

- **Step 1:** Verify that `/etc/hosts` file on both the master and standby UFM hosts contains the correct host names and IP addresses mapping.
- **Step 2:** If necessary, fix the mapping and retry the failover command.
12.3 Restoring UFM Data Upon In-Service Upgrade Failure

These instructions apply in high availability scenario only.

In the event of an in-service upgrade failure, the previous version of UFM's data will be safeguarded as a backup in the "/opt/ufm/BACKUP" directory, formatted as "ufm_upgrade_backup_<prev_version>-<new_version_<date>.zip."

To restore the data on the unupgraded node, follow these steps:

1. Copy the backup file from the upgraded node to the unupgraded node using the following command:

   scp /opt/ufm/BACKUP/ufm_upgrade_backup_<prev_version>-<new_version_<date>.zip root@<unupgraded_node_ip>:/opt/ufm/BACKUP/

2. Perform a failover of UFM to the master node, which is mandatory for data mount migration (including '/opt/ufm/files') to the master node: On the Master node, execute:

   ufm_ha_cluster takeover

3. Stop UFM on the unupgraded node:

   ufm_ha_cluster stop

4. Restore UFM configuration files from the backup:

   /opt/ufm/scripts/ufm_restore.sh -f /opt/ufm/BACKUP/ufm_upgrade_backup_<prev_version>-<new_version_<date>.zip

5. Start UFM on the unupgraded node (Note: Only the upgraded node can function until the upgrade issue is resolved, and failovers will not work).

Now, the issue that caused the upgrade failure can be addressed. If the problem is resolved, you can attempt the in-service upgrade again by failing UFM over to the upgraded node.

Alternatively, if needed, you can revert the changes made by reinstalling the old UFM version on the upgraded node.
13 Appendixes

- SM Configurations
- Appendix - Diagnostic Utilities
- Appendix - Supported Port Counters and Events
- Appendix - Used Ports
- Appendix - Configuration Files Auditing
- Appendix - IB Router
- Appendix - NVIDIA SHARP Integration
- Appendix - UFM SLURM Integration
- Appendix - Switch Grouping
- Appendix - Device Management Feature Support

13.1 SM Configurations

- Appendix - SM Default Files
- Appendix - UFM Subnet Manager Default Properties
- Appendix - Partitioning
- Appendix - Enhanced Quality of Service
- Appendix - OpenSM Configuration Files for Congestion Control
- Appendix - Routing Chains
- Appendix - Adaptive Routing
- Appendix - Security Features
- Appendix - SM Activity Report

13.1.1 Appendix - SM Default Files

The SM default files are located under the following paths:

- Default SM configuration file - /opt/ufm/files/conf/opensm/opensm.conf
- Default node name map file - /opt/ufm/files/conf/opensm/lib-node-name-map
- Default partition configuration file - /opt/ufm/files/conf/opensm/partitions.conf
- Default QoS policy configuration file - /opt/ufm/files/conf/opensm/qos-policy.conf
- Default prefix routes file - /opt/ufm/files/conf/opensm/prefix-routes.conf

13.1.2 Appendix - UFM Subnet Manager Default Properties

The following table provides a comprehensive list of UFM SM default properties.

<table>
<thead>
<tr>
<th>Category</th>
<th>Property</th>
<th>Config File Attribute</th>
<th>Default</th>
<th>Mode/Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generic</td>
<td>Subnet Prefix</td>
<td>subnet_prefix</td>
<td>0xfe80000000000000</td>
<td>RW</td>
<td>Subnet prefix used on the subnet 0xfe80000000000000</td>
</tr>
<tr>
<td>Category</td>
<td>Property</td>
<td>Config File Attribute</td>
<td>Default</td>
<td>Mode/Field</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>----------</td>
<td>-----------------------</td>
<td>---------</td>
<td>------------</td>
<td>-------------</td>
</tr>
<tr>
<td>LMC</td>
<td>lmc</td>
<td>0</td>
<td>RW</td>
<td></td>
<td>The LMC value used on the subnet: 0-7 Changes to the LMC parameter require a UFM restart.</td>
</tr>
<tr>
<td>SM LID</td>
<td>master_sm_lid</td>
<td>0</td>
<td>RW</td>
<td></td>
<td>Force LID for local SM when in MASTER state Selected LID must match configured LMC 0 disables the feature</td>
</tr>
<tr>
<td>Keys</td>
<td>M_Key</td>
<td>m_key</td>
<td>0x0000000000000000</td>
<td>RW</td>
<td>M_Key value sent to all ports -used to qualify the set(PortInfo)</td>
</tr>
<tr>
<td></td>
<td>m_key_lease_period</td>
<td>0</td>
<td>RW</td>
<td></td>
<td>The lease period used for the M_Key on the subnet in [sec]</td>
</tr>
<tr>
<td></td>
<td>sm_key</td>
<td>0x0000000000000000</td>
<td>RO</td>
<td></td>
<td>SM_Key value of the SM used for SM authentication</td>
</tr>
<tr>
<td></td>
<td>sa_key</td>
<td>0x0000000000000000</td>
<td>RO</td>
<td></td>
<td>SM_Key value to qualify rcv SA queries as ‘trusted’</td>
</tr>
<tr>
<td></td>
<td>part_enforce</td>
<td>Out, In, Both (default-outbound and inbound enforcement enabled)</td>
<td>RO</td>
<td></td>
<td>Partition enforcement type (for switches)</td>
</tr>
<tr>
<td></td>
<td>m_key_lookup</td>
<td>FALSE</td>
<td>RW</td>
<td></td>
<td>If FALSE, SM will not try to determine the m_key of unknown ports.</td>
</tr>
<tr>
<td></td>
<td>m_key_per_port</td>
<td>FALSE</td>
<td>RW</td>
<td></td>
<td>When m_key_per_port is enabled, OpenSM will generate an M_Key for each port</td>
</tr>
<tr>
<td>Limits</td>
<td>packet_life_time</td>
<td>0x12</td>
<td>RW</td>
<td></td>
<td>The maximum lifetime of a packet in a switch. The actual time is 4.096usec * 2^&lt;packet_life_time&gt; The value 0x14 disables the mechanism</td>
</tr>
<tr>
<td></td>
<td>vl_stall_count</td>
<td>0x07</td>
<td>RO</td>
<td></td>
<td>The number of sequential packets dropped that cause the port to enter the VL Stalled state. The result of setting the count to zero is undefined.</td>
</tr>
<tr>
<td>Category</td>
<td>Property</td>
<td>Config File Attribute</td>
<td>Default</td>
<td>Mode</td>
<td>Field</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>-------------------------------</td>
<td>-----------------------------</td>
<td>---------</td>
<td>------</td>
<td>-------</td>
</tr>
<tr>
<td>Leaf VL Stall Count</td>
<td>叶VL停滞计数</td>
<td>leaf_vl_stall_count</td>
<td>0x07</td>
<td>RO</td>
<td></td>
</tr>
<tr>
<td>Head Of Queue Lifetime</td>
<td>队列头 lifetime</td>
<td>head_of_queue_lifetime</td>
<td>0x12</td>
<td>RW</td>
<td></td>
</tr>
<tr>
<td>Leaf Head Of Queue Lifetime</td>
<td>队列头 lifetime</td>
<td>leaf_head_of_queue_lifetime</td>
<td>0x10</td>
<td>RW</td>
<td></td>
</tr>
<tr>
<td>Maximal Operational VL</td>
<td></td>
<td>max_op_vls</td>
<td>2</td>
<td>RW</td>
<td></td>
</tr>
<tr>
<td>Force Link Speed</td>
<td></td>
<td>force_link_speed</td>
<td>15 (Do NOT change)</td>
<td>RO</td>
<td></td>
</tr>
<tr>
<td>Limits</td>
<td>Subnet Timeout</td>
<td>subnet_timeout</td>
<td>18 (1second)</td>
<td>RW</td>
<td></td>
</tr>
<tr>
<td>Local PHY Error Threshold</td>
<td></td>
<td>local_phy_errors_threshold</td>
<td>0x08</td>
<td>RW</td>
<td></td>
</tr>
<tr>
<td>Overrun Errors Threshold</td>
<td></td>
<td>overrun_errors_threshold</td>
<td>0x08</td>
<td>RW</td>
<td></td>
</tr>
<tr>
<td>Sweep</td>
<td>Sweep Interval</td>
<td>sweep_interval</td>
<td>10</td>
<td>RW</td>
<td></td>
</tr>
<tr>
<td>Reassign Lids</td>
<td></td>
<td>reassign_lids</td>
<td>FALSE (disabled)</td>
<td>RW</td>
<td></td>
</tr>
<tr>
<td>Category</td>
<td>Property</td>
<td>Config File Attribute</td>
<td>Default</td>
<td>Mode/Field</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-------------------------------------------</td>
<td>---------------------------------</td>
<td>-----------</td>
<td>------------</td>
<td>----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Force Heavy Sweep</td>
<td></td>
<td>force_heavy_sweep_window</td>
<td>-1</td>
<td>RW</td>
<td>Forces heavy sweep after number of light sweeps (-1 disables this option and 0 will cause every sweep to be heavy)</td>
</tr>
<tr>
<td>Sweep On trap</td>
<td>sweep_on_trap</td>
<td>TRUE (enabled)</td>
<td>RW</td>
<td></td>
<td>If TRUE every trap 128 and 144 will cause a heavy sweep</td>
</tr>
<tr>
<td>Alternative Route Calculati on</td>
<td>max_alt_dr_path_retries</td>
<td>4</td>
<td>RW</td>
<td></td>
<td>Maximum number of attempts to find an alternative direct route towards unresponsive ports</td>
</tr>
<tr>
<td>Fabric Rediscovery</td>
<td>max_seq_redisc</td>
<td>2</td>
<td>RW</td>
<td></td>
<td>Max Failed Sequential Discovery Loops</td>
</tr>
<tr>
<td>Offsweep Rebalancing Enable</td>
<td>offsweep_balancing_enabled</td>
<td>FALSE</td>
<td>RW</td>
<td></td>
<td>Enable/Disable idle time routing rebalancing</td>
</tr>
<tr>
<td>Offsweep Rebalancing Window</td>
<td>offsweep_balancing_window</td>
<td>180</td>
<td>RW</td>
<td></td>
<td>Set the time window in seconds after sweep to start rebalancing</td>
</tr>
<tr>
<td>Handover</td>
<td>SM Priority</td>
<td>sm_priority</td>
<td>15</td>
<td>RO</td>
<td>SM (enabled). The priority used for deciding which is the master. Range is 0 (lowest priority) to 15 (highest)</td>
</tr>
<tr>
<td></td>
<td>Ignore Other SMs</td>
<td>ignore_other_sm</td>
<td>FALSE (disabled)</td>
<td>RO</td>
<td>If TRUE other SMs on the subnet should be ignored</td>
</tr>
<tr>
<td>Polling Timeout</td>
<td>sminfo_polling_timeout</td>
<td>10</td>
<td>RO</td>
<td></td>
<td>Timeout in seconds between two active master SM polls</td>
</tr>
<tr>
<td>Polling Retries</td>
<td>polling_rety_number</td>
<td>4</td>
<td>RO</td>
<td></td>
<td>Number of failing remote SM polls that declares it non-operational</td>
</tr>
<tr>
<td>Honor GUID-to-LID File</td>
<td>honor_guid2lid_file</td>
<td>FALSE (disabled)</td>
<td>RO</td>
<td></td>
<td>If TRUE, honor the guid2lid file when coming out of standby state, if the guid2lid file exists and is valid</td>
</tr>
<tr>
<td>Allowed SM GUID list</td>
<td>allowed_sm_guids</td>
<td>(null) (disabled)</td>
<td>RO</td>
<td></td>
<td>List of Host GUIDs where SM is allowed to run when specified. OpenSM ignores SM running on port that is not in this list. If 0, does not allow any other SM. If null, the feature is disabled.</td>
</tr>
<tr>
<td>Threading</td>
<td>Max Wire SMPs</td>
<td>max_wire_smps</td>
<td>8</td>
<td>RW</td>
<td>Maximum number of SMPs sent in parallel</td>
</tr>
<tr>
<td></td>
<td>Transaction Timeout</td>
<td>transaction_timeout</td>
<td>200</td>
<td>RO</td>
<td>The maximum time in [msec] allowed for a transaction to complete</td>
</tr>
<tr>
<td>Category</td>
<td>Property</td>
<td>Config File Attribute</td>
<td>Default</td>
<td>Mode / Field</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>------------------------</td>
<td>-----------------------------</td>
<td>---------</td>
<td>--------------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Max Message FIFO Timeout</strong></td>
<td>max_msg_fifo_timeout</td>
<td>10000</td>
<td>RO</td>
<td></td>
<td>Maximum time in [msec] a message can stay in the incoming message queue</td>
</tr>
<tr>
<td><strong>Routing Threads</strong></td>
<td>routing_threads_num</td>
<td>0</td>
<td>RW</td>
<td></td>
<td>Number of threads to be used for parallel minhop/updn calculations.</td>
</tr>
<tr>
<td><strong>Routing Threads Per Core</strong></td>
<td>max_threads_per_core</td>
<td>0</td>
<td>RW</td>
<td></td>
<td>Max number of threads that are allowed to run on the same processor during parallel computing. If 0, threads assignment per processor is up to operating system initial assignment.</td>
</tr>
<tr>
<td><strong>Logging</strong></td>
<td>log_file</td>
<td>/opt/ufm/files/log/opensm.log</td>
<td>RO</td>
<td></td>
<td>Path of Log file to be used</td>
</tr>
<tr>
<td><strong>Log Flags</strong></td>
<td>log_flags</td>
<td>Error and Info 0x03</td>
<td>RW</td>
<td></td>
<td>The log flags, or debug level being used.</td>
</tr>
<tr>
<td><strong>Force Log Flush</strong></td>
<td>force_log_flush</td>
<td>FALSE (disabled)</td>
<td>RO</td>
<td></td>
<td>Force flush of the log file after each log message</td>
</tr>
<tr>
<td><strong>Log Max Size</strong></td>
<td>log_max_size</td>
<td>4096</td>
<td>RW</td>
<td></td>
<td>Limit the size of the log file in MB. If overrun, log is restarted</td>
</tr>
<tr>
<td><strong>Accumulate Log File</strong></td>
<td>accum_log_file</td>
<td>TRUE (enabled)</td>
<td>RO</td>
<td></td>
<td>If TRUE, will accumulate the log over multiple OpenSM sessions</td>
</tr>
<tr>
<td><strong>Dump Files Directory</strong></td>
<td>dump_files_dir</td>
<td>/opt/ufm/files/log</td>
<td>RO</td>
<td></td>
<td>The directory to hold the file SM dumps (for multicast forwarding tables for example). The file is used collects information.</td>
</tr>
<tr>
<td><strong>Syslog log</strong></td>
<td>syslog_log</td>
<td>0x0</td>
<td>RW</td>
<td></td>
<td>Sets a verbosity of messages to be printed in syslog</td>
</tr>
<tr>
<td><strong>Misc</strong></td>
<td>node_name_map_name</td>
<td>Null</td>
<td>RW</td>
<td></td>
<td>Node name map for mapping node's to more descriptive node descriptions</td>
</tr>
<tr>
<td><strong>SA database File</strong></td>
<td>sa_db_file</td>
<td>Null</td>
<td>RO</td>
<td></td>
<td>SA database file name</td>
</tr>
<tr>
<td><strong>No Clients Reregistration</strong></td>
<td>no_clients_rereg</td>
<td>FALSE (disabled)</td>
<td>RO</td>
<td></td>
<td>If TRUE, disables client reregistration</td>
</tr>
<tr>
<td><strong>Exit On Fatal Event</strong></td>
<td>exit_on_fatal</td>
<td>TRUE (enabled)</td>
<td>RO</td>
<td></td>
<td>If TRUE (enabled), the SM exits for fatal initialization issues</td>
</tr>
<tr>
<td>Category</td>
<td>Property</td>
<td>Config File Attribute</td>
<td>Default</td>
<td>Mode/Field</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>---------------------------------------</td>
<td>---------------------------</td>
<td>------------------</td>
<td>------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Switch Isolation From Routing</td>
<td>held_back_sw_file</td>
<td>Null</td>
<td>RW</td>
<td></td>
<td>File that contains GUIDs of switches isolated from routing</td>
</tr>
<tr>
<td>Enable NVIDIA SHARP support</td>
<td>sharp_enabled</td>
<td>Enabled</td>
<td>RW</td>
<td></td>
<td>Defines whether to enable/disable NVIDIA SHARP on supporting ports.</td>
</tr>
<tr>
<td>Multicast</td>
<td>Disable Multicast</td>
<td>disable_multicast</td>
<td>FALSE (disabled)</td>
<td>RO</td>
<td>If TRUE, OpenSM should disable multicast support and no multicast routing is performed</td>
</tr>
<tr>
<td>Multicast</td>
<td>Default MC group parameters</td>
<td>default_mcg_mtu</td>
<td>0</td>
<td>RW</td>
<td>Default MC group MTU for dynamic group creation. 0 disables this feature, otherwise, the value is a valid IB encoded MTU</td>
</tr>
<tr>
<td>Multicast</td>
<td>Default MC group rate</td>
<td>default_mcg_rate</td>
<td>0</td>
<td>RW</td>
<td>Default MC group rate for dynamic group creation. 0 disables this feature, otherwise, the value is a valid IB encoded rate</td>
</tr>
<tr>
<td>Multicast</td>
<td>Enable incremental multicast routing</td>
<td>enable_inc_mc_routing</td>
<td>FALSE</td>
<td>RW</td>
<td>Enable incremental multicast routing</td>
</tr>
<tr>
<td>Multicast</td>
<td>MC root file</td>
<td>mc_roots_file</td>
<td>null</td>
<td>RW</td>
<td>Specify predefined MC groups root guids</td>
</tr>
<tr>
<td>QoS</td>
<td>Settings</td>
<td>qos</td>
<td>FALSE (disabled)</td>
<td>RW</td>
<td>If FALSE (disabled), SM will not apply QoS settings</td>
</tr>
<tr>
<td>Unhealth y Ports</td>
<td>Enabling Unhealthy Ports</td>
<td>hm_unhealthy_ports_checks</td>
<td>TRUE</td>
<td>RW</td>
<td>Enables Unhealthy Ports configuration</td>
</tr>
<tr>
<td></td>
<td>Configuration file</td>
<td>hm_ports_health_policy_ file</td>
<td>null</td>
<td>RW</td>
<td>Specifies configuration file for health policy</td>
</tr>
<tr>
<td></td>
<td>Unhealthy actions</td>
<td>hm_sw_manual_action</td>
<td>no_discover</td>
<td>RW</td>
<td>Specifies what to do with switch ports which were manually added to health policy file</td>
</tr>
<tr>
<td></td>
<td>MADs validation</td>
<td>validate_smp</td>
<td>TRUE</td>
<td>RW</td>
<td>If set to TRUE, opensm will ignore nodes sending non-spec compliant MADs. When set to FALSE, opensm will log the warning in the opensm log file about non-compliant node</td>
</tr>
<tr>
<td>Category</td>
<td>Property</td>
<td>Config File Attribute</td>
<td>Default</td>
<td>Mode/Fiel</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------</td>
<td>----------------------------------------</td>
<td>-------------------------------------</td>
<td>--------------</td>
<td>-----------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Routing</td>
<td>Unicast Routing engine</td>
<td>routing_engine</td>
<td>(null)</td>
<td>RW</td>
<td>By default, ar_updn routing engine is used by the SM. Supported routing engines are minhop, updn, dnup, ftree, dor, torus-2QoS, kdor-hc, kdor-ghc, dfp, dfp2, ar_updn, ar_ftree and ar_dor.</td>
</tr>
<tr>
<td>Randomization</td>
<td>scatter_ports</td>
<td></td>
<td>8</td>
<td>RW</td>
<td>Assigns ports in a random order instead of round-robin. If 0, the feature is disabled, otherwise use the value as a random seed. Applicable to the MINHOP/UPDN routing algorithms</td>
</tr>
<tr>
<td>Randomization</td>
<td>guid_routing_order_no_scatter</td>
<td></td>
<td>TRUE</td>
<td>RO</td>
<td>Do not use scatter for ports defined in guid_routing_order_file</td>
</tr>
<tr>
<td>Unicast Routing Caching</td>
<td>use_ucast_cache</td>
<td></td>
<td>TRUE</td>
<td>RW</td>
<td>Use unicast routing cache for routing computation time improvement</td>
</tr>
<tr>
<td>GUID Ordering During Routing</td>
<td>guid_routing_order_file</td>
<td></td>
<td>NULL</td>
<td>RW</td>
<td>The file holding guid routing order of particular guids (for MinHop, Up/Down)</td>
</tr>
<tr>
<td>Torus Routing</td>
<td>torus_config</td>
<td>/opt/ufm/files/conf/opensm/torus-2QoS.con</td>
<td></td>
<td>RW</td>
<td>Torus-2QoS configuration file name</td>
</tr>
<tr>
<td>Routing Chains</td>
<td>pgrp_policy_file</td>
<td></td>
<td>NULL</td>
<td>RW</td>
<td>The file holding the port groups policy</td>
</tr>
<tr>
<td></td>
<td>topo_policy_file</td>
<td></td>
<td>NULL</td>
<td>RW</td>
<td>The file holding the topology policy</td>
</tr>
<tr>
<td></td>
<td>rch_policy_file</td>
<td></td>
<td>NULL</td>
<td>RW</td>
<td>The file holding the routing chains policy</td>
</tr>
<tr>
<td></td>
<td>max_topologies_per_sw</td>
<td></td>
<td>1</td>
<td>RO</td>
<td>Defines maximal number of topologies to which a single switch may be assigned during routing engine chain configuration.</td>
</tr>
<tr>
<td>Incremental Multicast Routing (IMR)</td>
<td>enable_inc_mc_routing</td>
<td></td>
<td>TRUE</td>
<td>RW</td>
<td>If TRUE, MC nodes will be added to the MC tree incrementally. When set to FALSE, the tree will be recalculated per each change.</td>
</tr>
<tr>
<td>MC Global root</td>
<td>mc_primary_root_guid/mc_secondary_root_guid</td>
<td>0x00000000000000000000 (for both)</td>
<td></td>
<td>RW</td>
<td>Primary and Secondary global mc root guid</td>
</tr>
<tr>
<td>Scatter ports</td>
<td>use_scatter_for_switch_lid</td>
<td></td>
<td>FALSE</td>
<td>RW</td>
<td>Use scatter when routing to the switch’s LiDs</td>
</tr>
<tr>
<td>Category</td>
<td>Property</td>
<td>Config File Attribute</td>
<td>Default</td>
<td>Mode</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-----------------------------------</td>
<td>-----------------------------</td>
<td>---------</td>
<td>------</td>
<td>----------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>updn lid tracking mode</td>
<td>updn_lid_tracking_mode</td>
<td>FALSE</td>
<td>RW</td>
<td>Controls whether SM will use LID tracking or not when updn or ar_updn routing engine is used</td>
</tr>
<tr>
<td>Events</td>
<td>Event Subscription Handling</td>
<td>drop_subscr_on_report_fail</td>
<td>FALSE</td>
<td>RW</td>
<td>Drop subscription on report failure (o13-17.2.1)</td>
</tr>
<tr>
<td></td>
<td>Event Subscription Handling</td>
<td>drop_event_subscriptions</td>
<td>TRUE</td>
<td>RW</td>
<td>Drop event subscriptions (InformInfo and ServiceRecords) on port removal and SM coming out of STANDBY</td>
</tr>
<tr>
<td>Virtualization</td>
<td>Virtualization enabled</td>
<td>virt_enabled</td>
<td>Enabled</td>
<td>RW</td>
<td>Enables/disables virtualization support</td>
</tr>
<tr>
<td></td>
<td>Maximum ports in virtualization process</td>
<td>virt_max_ports_in_process</td>
<td>64</td>
<td>RW</td>
<td>Sets a number of ports to be handled on each virtualization process cycle</td>
</tr>
<tr>
<td>Router</td>
<td>Router aguid enable</td>
<td>rtr_aguid_enable</td>
<td>0 (Disabled)</td>
<td>RW</td>
<td>Defines whether the SM should create alias GUIDs required for router support for each HCA port</td>
</tr>
<tr>
<td></td>
<td>Router path record flow label</td>
<td>rtr_pr_flow_label</td>
<td>0</td>
<td>RW</td>
<td>Defines flow label value to use in multi-subnet path query responses</td>
</tr>
<tr>
<td></td>
<td>Router path record tclass</td>
<td>rtr_pr_tclass</td>
<td>0</td>
<td>RW</td>
<td>Defines tclass value to use in multi-subnet path query responses</td>
</tr>
<tr>
<td></td>
<td>Router path record sl</td>
<td>rtr_pr_sl</td>
<td>0</td>
<td>RW</td>
<td>Defines sl value to use in multi-subnet path query responses</td>
</tr>
<tr>
<td></td>
<td>Router path record MTU</td>
<td>rtr_pr_mtu</td>
<td>4 (IB_MTU_LEN_2048)</td>
<td>RW</td>
<td>Define MTU value to use in multi-subnet path query responses</td>
</tr>
<tr>
<td></td>
<td>Router path record rate</td>
<td>rtr_pr_rate</td>
<td>16 (IB_PATH_RECORD_RATE_100_GBS)</td>
<td>RW</td>
<td>Defines rate value to use in multi-subnet path query responses</td>
</tr>
<tr>
<td>SA Security</td>
<td>SA Tnhanced Trust Model (SAETM)</td>
<td>sa_enhanced_trust_mode</td>
<td>FALSE</td>
<td>RW</td>
<td>Controls whether SAETM is enabled.</td>
</tr>
</tbody>
</table>
### Configuring UFM for SR-IOV

Single-root I/O virtualization (SR-IOV) enables a PCI Express (PCIe) device to appear to be multiple separate physical PCIe devices.

UFM is ready to work with SR-IOV devices by default. You can fine-tune the configuration using the SM configuration.

The following arguments are available for ConnectX-5 and later devices:

<table>
<thead>
<tr>
<th>Argument</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>virt_enabled</td>
<td>- 0: no virtualization support</td>
<td>Virtualization support</td>
</tr>
<tr>
<td></td>
<td>- 1: disable virtualization on all virtualization supporting ports</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- 2: enable virtualization on all virtualization supporting ports</td>
<td></td>
</tr>
</tbody>
</table>

**Description**

- **Configuring UFM for SR-IOV**
- **Single-root I/O virtualization (SR-IOV)** enables a PCI Express (PCIe) device to appear to be multiple separate physical PCIe devices.
- **UFM** is ready to work with SR-IOV devices by default. You can fine-tune the configuration using the SM configuration.
- **The following arguments are available for ConnectX-5 and later devices:**
  - **virt_enabled**: 0 - no virtualization support; 1 - disable virtualization on all virtualization supporting ports; 2 - enable virtualization on all virtualization supporting ports (default)
<table>
<thead>
<tr>
<th>Argument</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>virt_max_ports_in_process</td>
<td>Possible values: 0-65535; where 0 processes all pending ports Default: 64</td>
<td>Maximum number of ports to be processed simultaneously by the virtualization manager</td>
</tr>
<tr>
<td>virt_default_hop_limit</td>
<td>Possible values: 0-255 Default: 2</td>
<td>Default value for hop limit to be returned in path records where either the source or destination are virtual ports</td>
</tr>
</tbody>
</table>

13.1.2.2 Isolating Switch From Routing

UFM can isolate particular switches from routing in order to perform maintenance of the switches with minimal interruption to the existing traffic in the fabric.

Isolating a switch from routing is done via UFM Subnet Manager as follows:

1. Create a file that includes either the node GUIDs or system GUID of the switches under maintenance. For example:

   0x1234566
   0x1234567

2. Set the filename of the parameter held_back_sw_file in the /conf/opensm.conf file (the same as the file created in Step 1).

3. Run:

   ```shell
   kill -s HUP 'pidof opensm'
   ```

Once SM completes rerouting, the traffic does not go through the ports of isolated switches.

To attach the switch to the routing:

1. Remove the GUID of the switch from the list of isolated switches defined in Step 1 of the isolation process.

2. Run:

   ```shell
   kill -s HUP 'pidof opensm'
   ```

Once SM completes rerouting, traffic will go through the switch.

13.1.3 Appendix - Partitioning

Partitioning enforces isolation of the fabric. The default partition is created on all managed devices. Devices that are running an SM, all switches, routers, and gateways are added to the default partition with full membership. By default, all the HCA ports are also added to the default partition with FULL membership.

Partitioning is provisioned to the Subnet Manager via the partitions.conf configuration file, which cannot be removed or manually modified.
If required, you can add an extension to the `partitions.conf` file that is generated by UFM. You can edit the file, `/opt/ufm/files/conf/partitions.conf.user_ext`, and the content of this extension file will be added to the `partitions.conf` file. Files synchronization is done by UFM on every logical model change. However, it can also be triggered manually by running the `/opt/ufm/scripts/sync_partitions_conf.sh` script. The script validates and merges the `/opt/ufm/files/conf/partitions.conf.user_ext` file into the `/opt/ufm/files/conf/opensm/partitions.conf` file and starts the heavy sweep on the Subnet Manager.

The `partitions.conf.user_ext` uses the same format as the `partitions.conf` file. See SM Partitions.conf File Format for the format of the `partitions.conf` file.

For example, to add server ports to PKey 4:

```plaintext
IOPartition=0x4, ipoib, sl=0, defmember=full : 0x8f10001072a41; 13.1.3.1
```

13.1.3.1 SM Partitions.conf File Format

This appendix presents the content and format of the SM `partitions.conf` file.

---

For those who use NVIDIA gateway systems, for proper system functionality, disable the automatic partitioning by changing the attribute `gateway_port_partitioning = none` in the `/opt/ufm/files/conf/gv.cfg` configuration. Restart UFM for the change to take effect.

The maximum length of the line in the `partitions.conf` file is 4096 characters. However, to enable long PKeys, it is possible to split the pkey membership to multiple lines:

```plaintext
IOPartition=0x4, ipoib, sl=0, defmember=full : <port-guid1> , <port-guid2> ;
IOPartition=0x4, ipoib, sl=0, defmember=full : <port-guid3> , <port-guid4> ;
```

The `partitions.conf.user_ext` uses the same format as the `partitions.conf` file. See SM Partitions.conf File Format for the format of the `partitions.conf` file.

For example, to add server ports to PKey 4:

```plaintext
IOPartition=0x4, ipoib, sl=0, defmember=full : 0x8f10001072a41;
```
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]*
    - 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 [10Gb/s])
  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, ipoiib : 0x123456=full, 0x3456789034=limited, 0x2134af2306 ;
YetAnotherOne = 0x300 : SELF=full ;
YetAnotherOne = 0x300 : ALL=limited ;
13.1.4 Appendix - Enhanced Quality of Service

Enhanced QoS provides a higher resolution of QoS at the service level (SL). Users can configure rate limit values per SL for physical ports, virtual ports, and port groups, using enhanced_qos_policy_file configuration parameter.

Valid values of this parameter:

- Full path to the policy file through which Enhanced QoS Manager is configured
- “null” - to disable the Enhanced QoS Manager (default value)

⚠️ To enable Enhanced QoS Manager, QoS must be enabled in SM configuration file.

13.1.4.1 Enhanced QoS Policy File

The policy file is comprised of two sections:

- **BW_NAMES**: Used to define bandwidth setting and name (currently, rate limit is the only setting). Bandwidth names are defined using the syntax:

  `<name> = <rate limit in 1Mbps units>`

  **Example**:

  

  ```
  My_bandwidth = 50
  ```

- **BW_RULES**: Used to define the rules that map the bandwidth setting to a specific SL of a specific GUID. Bandwidth rules are defined using the syntax:

  `<guid>|<port group name> = <sl id>:<bandwidth name>, <sl id>:<bandwidth name>...`

  **Examples**:

  ```
  0x2c90000000025 = 5:My_bandwidth, 7:My_bandwidth
  Port_grp1 = 3:My_bandwidth, 9:My_bandwidth`
13.1.4.1.1 Notes
- Rate limit = 0 represents unlimited rate limit.
- Any unspecified SL in a rule will be set to 0 (unlimited) rate limit automatically.
- “default” is a well-known name which can be used to define a default rule used for any GUID with no defined rule (If no default rule is defined, any GUID without a specific rule will be configured with unlimited rate limit for all SLs).
- Failure to complete policy file parsing leads to an undefined behavior. User must confirm no relevant error messages in SM log in order to ensure Enhanced QoS Manager is configured properly.
- An empty file with only ‘BW_NAMES’ and ‘BW_RULES’ keywords configures the network with an unlimited rate limit.
- The VPORT_BW_RULES section is optional and includes virtual port GUIDs only (including the vport0 GUID). Physical port GUIDs added to this section are treated as vport0 GUIDs.

13.1.4.1.2 Policy File Example
The below is an example of configuring all ports in the fabric with rate limit of 50Mbps on SL1, except for GUID 0x2c9000000025, which is configured with rate limit of 100Mbps on SL1. In this example, all SLs (other than SL1) are unlimited.

```
-----------------------------------------------------------------------
BW_NAMES
bw1 = 50
bw2 = 100
BW_RULES
default: 1:bw1
0x2c9000000025: 1:bw2
------------------------------------------------------------------------
VPORT_BW_RULES
default = all:DEF_BW_2
```

13.1.5 Appendix - OpenSM Configuration Files for Congestion Control
13.1.5.1 OpenSM Configuration
13.1.5.1.1 Enabling Congestion Control
In order to enable congestion control, set the parameter `mlnx_congestion_control` in the OpenSM configuration file to 2. For example:

```
mlnx_congestion_control 2
```

To disable congestion control, set the parameter value to 1. For example:

```
mlnx_congestion_control 1
```

430
13.1.5.1.2 Defining a Congestion Control Policy File

To define a congestion control policy file, set the parameter `congestion_control_file` in OpenSM configuration file to point to congestion control policy file. For example:

```
congestion_control_policy_file /opt/ufm/files/conf/opensm/conf/congestion_control_policy_file
```

The file includes a reference to an active algorithm file name. The algorithm file has to be inside the `ppcc_algo_dir`. For Example:

```
ca_algo_import_start
  algo_start
  algo_id: 1
  algo_file_name: active_algo_file_name
  # PPCC parameter by name, as defined in algo profile
  parameters: (BW_G, 400), (ALPHA, 3932), (MAX_DEC, 63569), (MAX_INC, 69469), (AI, 36), (HAI, 1200)
  algo_end
ca_algo_import_end
```

The `ca_algo_import` block contains all the algo blocks that map an `algo_id` to an algorithm profile file. The `algo_id` field of the algo blocks must be unique and start from 1. This block is used to import the various PCC algorithms into the configuration and associate them with their `algo_id` values.

13.1.5.1.3 Defining a directory for PPCC algorithm profiles

To define a directory for the programmable congestion control algorithm profiles, set the parameter `ppcc_algo_dir` in OpenSM configuration file. For Example:

```
ppcc_algo_dir /opt/ufm/files/conf/opensm/conf/ppcc_algo_dir
```

13.1.6 Appendix - Routing Chains

The routing chains feature is offering a solution that enables one to configure different parts of the fabric and define a different routing engine to route each of them. The routings are done in a sequence (hence the name "chains") and any node in the fabric that is configured in more than one part is left with the last routing engine updated for it.

13.1.6.1 Configuring Routing Chains

The configuration for the routing chains feature consists of the following steps:

1. Define the port groups.
2. Define topologies based on previously defined port groups.
3. Define configuration files for each routing engine.
4. Define routing engine chains over defined topologies.
13.1.6.1.1 Defining Port Groups

The basic idea behind the port groups is the ability to divide the fabric into sub-groups and give each group an identifier that can be used to relate to all nodes in this group. The port groups are used to define the participants in each of the routing algorithms.

13.1.6.1.2 Defining Port Group Policy File

In order to define a port group policy file, set the parameter ‘pgrp_policy_file’ in the opensm configuration file, as follows:

```
/opt/ufm/files/conf/opensm/port_groups_policy_file.conf
```

13.1.6.1.3 Configuring Port Group Policy

The port groups policy file details the port groups in the fabric. The policy file should be composed of one or more paragraphs that define a group. Each paragraph should begin with the line ‘port-group’ and end with the line ‘end-port-group’.

For example:

```
port-group
 ...port group qualifiers...
end-port-group
```

13.1.6.1.4 Port Group Qualifiers

**Note:** Unlike the port group’s beginning and ending which do not require a colon, all qualifiers must end with a colon (‘:’). Also - a colon is a predefined mark that must not be used inside qualifier values. An inclusion of a colon in the name or the use of a port group, will result in the policy’s failure.

13.1.6.1.4.1 Table 62: Port Group Qualifiers

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>Each group must have a name. Without a name qualifier, the policy fails.</td>
<td>name: grp1</td>
</tr>
<tr>
<td>use</td>
<td>‘use’ is an optional qualifier that one can define in order to describe the usage of this port group (if undefined, an empty string is used as a default).</td>
<td>use: first port group</td>
</tr>
</tbody>
</table>

13.1.6.1.5 Rule Qualifiers

There are several qualifiers used to describe a rule that determines which ports will be added to the group. Each port group may contain one or more rules of the rule qualifiers in Table 63 (at least one rule shall be defined for each port group).
### Table 63: Rule Qualifiers

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
</table>
| guid list        | Comma separated list of guids to include in the group. If no specific physical ports were configured, all physical ports of the guid are chosen. However, for each guid, one can detail specific physical ports to be included in the group. This can be done using the following syntax:  
  • Specify a specific port in a guid to be chosen  
    port-guid: 0x283@3  
  • Specify a specific list of ports in a guid to be chosen  
    port-guid: 0x289@2-5  
  • Specify a list of specific ports and ports ranges in a guid to be chosen  
    port-guid: 0x289@2-5/7/9-13/18  
  • Complex rule  
    port-guid: 0x283@5-8/12/14, 0x286, 0x289/6/8/12 | port-guid: 0x283, 0x286, 0x289 |
| port guid range  | It is possible to configure a range of guids to be chosen to the group. However, while using the range qualifier, it is impossible to detail specific physical ports. Note: A list of ranges cannot be specified. The below example is invalid and will cause the policy to fail:  
    port-guid-range: 0x283-0x289, 0x290-0x295 | port-guid-range: 0x283-0x289 |
| port name        | One can configure a list of hostnames as a rule. Hosts with a node description that is built out of these hostnames will be chosen. Since the node description contains the network card index as well, one might also specify a network card index and a physical port to be chosen. For example, the given configuration will cause only physical port 2 of a host with the node description ‘kuku HCA-1’ to be chosen.  
  port and hca_idx parameters are optional. If the port is unspecified, all physical ports are chosen. If hca_idx is unspecified, all card numbers are chosen. Specifying a hostname is mandatory. One can configure a list of hostname/port/hca_idx sets in the same qualifier as follows:  
    port-name: hostname=kuku; port=2; hca_idx=1  
    hostname=host1; port=3, hostname=host2  
  Note: port-name qualifier is not relevant for switches, but for HCA’s only. | port-name: hostname=kuku; port=2; hca_idx=1 |
<p>| port regexp      | One can define a regular expression so that only nodes with a matching node description will be chosen to the group | port-regexp: SW.* |</p>
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>It is possible to specify one physical port to be chosen for matching nodes (there is no option to define a list or a range of ports). The given example will cause only nodes that match physical port 3 to be added to the group.</td>
<td>port-regexp: SW.*:3</td>
</tr>
<tr>
<td>union rule</td>
<td>It is possible to define a rule that unites two different port groups. This means that all ports from both groups will be included in the united group.</td>
<td>union-rule: grp1, grp2</td>
</tr>
<tr>
<td>subtract rule</td>
<td>One can define a rule that subtracts one port group from another. The given rule, for example, will cause all the ports which are a part of grp1, but not included in grp2, to be chosen. In subtraction (unlike union), the order does matter, since the purpose is to subtract the second group from the first one. There is no option to define more than two groups for union/subtraction. However, one can unite/subtract groups which are a union or a subtraction themselves, as shown in the port groups policy file example.</td>
<td>subtract-rule: grp1, grp2</td>
</tr>
</tbody>
</table>

### 13.1.6.1.6 Predefined Port Groups

There are 3 predefined port groups that are available for use, yet cannot be defined in the policy file (if a group in the policy is configured with the name of one of these predefined groups, the policy fails) -

- ALL - a group that includes all nodes in the fabric
- ALL_SWITCHES - a group that includes all switches in the fabric.
- ALL_CAS - a group that includes all HCA's in the fabric.

### 13.1.6.1.7 Port Groups Policy Examples

```plaintext
port-group
name: grp3
use: Subtract of groups grp1 and grp2
subtract-rule: grp1, grp2
end-port-group

port-group
name: grp1
port-guid: 0x281, 0x282, 0x283
end-port-group

port-group
name: grp2
port-guid-range: 0x282-0x286
port-name: hostname=server1 port=1
end-port-group

port-group
name: grp4
port-name: hostname=kika port=1 hca_idx=1
end-port-group

port-group
name: grp3
union-rule: grp3, grp4
end-port-group
```
13.1.6.2 Defining Topologies Policy File

In order to define a port group policy file, set the parameter ‘topo_policy_file’ in the opensm configuration file.

```
/opt/ufm/files/conf/opensm/topo_policy_file.conf
```

13.1.6.2.1 Configuring Topology Policy

The topologies policy file details a list of topologies. The policy file should be composed of one or more paragraphs which define a topology. Each paragraph should begin with the line ‘topology’ and end with the line ‘end-topology’.

For example:

```
topology
...topology qualifiers...
end-topology
```

13.1.6.2.2 Topology Qualifiers

Unlike topology and end-topology which do not require a colon, all qualifiers must end with a colon (‘:’). Also - a colon is a predefined mark that must not be used inside qualifier values. An inclusion of a column in the qualifier values will result in the policy’s failure.

All topology qualifiers are mandatory. Absence of any of the below qualifiers will cause the policy parsing to fail.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>Topology ID. Legal Values - any positive value.  Must be unique.</td>
<td>id: 1</td>
</tr>
<tr>
<td>sw-grp</td>
<td>Name of the port group that includes all switches and switch ports to be used in this topology.</td>
<td>sw-grp: some_switches</td>
</tr>
<tr>
<td>hca-grp</td>
<td>Name of the port group that includes all HCA’s to be used in this topology.</td>
<td>hca-grp: some_hosts</td>
</tr>
</tbody>
</table>

13.1.6.3 Configuration File per Routing Engine

Each engine in the routing chain can be provided by its own configuration file. Routing engine configuration file is the fraction of parameters defined in the main opensm configuration file.

Some rules should be applied when defining a particular configuration file for a routing engine:

- Parameters that are not specified in specific routing engine configuration file are inherited from the main opensm configuration file.
- The following configuration parameters are taking effect only in the main opensm configuration file:
- qos and qos_* settings like (vl_arb, sl2vl, etc.)
- lmc
- routing_engine

13.1.6.3.1 Defining Routing Chain Policy File

In order to define a port group policy file, set the parameter ‘rch_policy_file’ in the opensm configuration file, as follows:

```bash
/opt/ufm/files/conf/opensm/routing_chains_policy.conf
```

13.1.6.3.2 First Routing Engine in Chain

The first unicast engine in a routing chain must include all switches and HCA’s in the fabric (topology id must be 0). The path-bit parameter value is path-bit 0 and it cannot be changed.

13.1.6.3.3 Configuring Routing Chains Policy

The routing chains policy file details the routing engines (and their fallback engines) used for the fabric’s routing. The policy file should be composed of one or more paragraphs which defines an engine (or a fallback engine). Each paragraph should begin with the line ‘unicast-step’ and end with the line ‘end-unicast-step’.

For example:

```plaintext
unicast-step
  ...routing engine qualifiers...
end-unicast-step
```

13.1.6.3.4 Routing Engine Qualifiers

⚠️ Unlike unicast-step and end-unicast-step which do not require a colon, all qualifiers must end with a colon (‘:’). Also - a colon is a predefined mark that must not be used inside qualifier values. An inclusion of a colon in the qualifier values will result in the policy’s failure.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
</table>
| id        | ‘id’ is mandatory. Without an id qualifier for each engine, the policy fails.  
- Legal values - size_t value (0 is illegal).  
- The engines in the policy chain are set according to an ascending id order, so it is highly crucial to verify that the id that is given to the engines match the order in which you would like the engines to be set. | is: 1 |
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>engine</td>
<td>This is a mandatory qualifier that describes the routing algorithm used within this unicast step. Currently, on the first phase of routing chains, legal values are minhop/ftree/updn.</td>
<td>engine: minhop</td>
</tr>
<tr>
<td>use</td>
<td>This is an optional qualifier that enables one to describe the usage of this unicast step. If undefined, an empty string is used as a default.</td>
<td>use: ftree routing for cluster 1</td>
</tr>
<tr>
<td>config</td>
<td>This is an optional qualifier that enables one to define a separate opensm config file for a specific unicast step. If undefined, all parameters are taken from main opensm configuration file.</td>
<td>config: /etc/config/opensm2.cfg</td>
</tr>
<tr>
<td>topology</td>
<td>Define the topology that this engine uses.</td>
<td>topology: 1</td>
</tr>
<tr>
<td></td>
<td>• Legal value - id of an existing topology that is defined in topologies policy (or zero that represents the entire fabric and not a specific topology).</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Default value - If unspecified, a routing engine will relate to the entire fabric (as if topology zero was defined).</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Notice: The first routing engine (the engine with the lowest id) MUST be configured with topology: 0 (entire fabric) or else, the routing chain algorithm will fail.</td>
<td></td>
</tr>
<tr>
<td>fallback-to</td>
<td>This is an optional qualifier that enables one to define the current unicast step as a fallback to another unicast step. This can be done by defining the id of the unicast step that this step is a fallback to.</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>• If undefined, the current unicast step is not a fallback.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• If the value of this qualifier is a non-existent engine id, this step will be ignored.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• A fallback step is meaningless if the step it is a fallback to did not fail.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• It is impossible to define a fallback to a fallback step (such definition will be ignored)</td>
<td></td>
</tr>
<tr>
<td>path-bit</td>
<td>This is an optional qualifier that enables one to define a specific lid offset to be used by the current unicast step. Setting lmc &gt; 0 in main opensm configuration file is a prerequisite for assigning specific path-bit for the routing engine. Default value is 0 (if path-bit is not specified)</td>
<td>Path-bit: 1</td>
</tr>
</tbody>
</table>

13.1.6.3.5 Dump Files per Routing Engine

Each routing engine on the chain will dump its own data files if the appropriate log_flags is set (for instance 0x43).

- The files that are dumped by each engine are:
  - opensm-lid-matrix.dump
  - opensm-lfts.dump
  - opensm.fdbs
These files should contain the relevant data for each engine topology.

- opensm-subnet.lst

Each engine concatenates its ID and routing algorithm name in its dump files names, as follows:

- opensm-lid-matrix.2.minhop.dump
- opensm.f dbs.3.ftree
- opensm-subnet.4.updn.lst

If a fallback routing engine is used, both the routing engine that failed and the fallback engine that replaces it, dump their data.

If, for example, engine 2 runs ftree and it has a fallback engine with 3 as its id that runs minhop, one should expect to find 2 sets of dump files, one for each engine:

- opensm-lid-matrix.2.ftree.dump
- opensm-lid-matrix.3.minhop.dump
- opensm.f dbs.2.ftree
- opensm.f dbs.3.munhop

13.1.7 Appendix - Adaptive Routing

As of UFM v6.4, Adaptive Routing plugin is no longer required for Adaptive Routing and SHIELD configuration. AR is now part of the core Subnet Manager implementation. However, upgrading UFM to v6.4 from an earlier version using the AR plugin will remain possible.

For information on how to set up AR and SHIELD, please refer to How-To Configure Adaptive Routing and Self Healing Networking.

13.1.8 Appendix - Security Features

13.1.8.1 SA Enhanced Trust Model (SAETM)

Standard SA has a concept of trust-based requests on the SA_Key that is part of each SA MAD. A trusted request is when the SA_Key value is not equal to zero but equals the SA configured value, while an untrusted request is when the SA_Key value equals zero in the request. If a request has a non-zero SA_Key value that is different from the configured SA key, it will be dropped and reported.

When SAETM is enabled, the SA limits the set of untrusted requests allowed. Untrusted requests that are not allowed according to SAETM will be silently dropped (for the set of untrusted requests allowed, see the following section below).

SAETM feature is disabled by default. To enable it, set the sa_enhanced_trust_model parameter to TRUE.

Additional SAETM Configuration Parameters
### Parameter Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>sa_etm_allow_untrusted_guidinfo_rec</code></td>
<td>Defines whether to allow GUIDInfoRecord as part of the SAETM set of untrusted requests allowed (see section below)</td>
</tr>
<tr>
<td><code>sa_etm_allow_guidinfo_rec_by_vf</code></td>
<td>Defines whether to drop GUIDInfoRecord from non-physical ports (see section below)</td>
</tr>
<tr>
<td><code>sa_etm_allow_untrusted_proxy_requests</code></td>
<td>Defines the behavior for proxy requests (see section below)</td>
</tr>
<tr>
<td><code>sa_etm_max_num_mcsgs</code> / <code>sa_etm_max_num_srvcs</code> / <code>sa_etm_max_num_event_subs</code></td>
<td>Defines the registration limits in SAETM (see section below)</td>
</tr>
</tbody>
</table>

### 13.1.8.1.1 Set of Untrusted SA Requests Allowed

The following table lists the untrusted requests allowed when SAETM is enabled:

<table>
<thead>
<tr>
<th>Request</th>
<th>Request Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>MCMemberRecord</td>
<td>Get/Set/Delete</td>
</tr>
<tr>
<td>PathRecord</td>
<td>Get</td>
</tr>
<tr>
<td>PathRecord</td>
<td>GetTable (only if both destination and source are specified (e.g. only point to point))</td>
</tr>
<tr>
<td>ServiceRecord</td>
<td>Get/Set/Delete</td>
</tr>
<tr>
<td>ClassPortInfo</td>
<td>Get</td>
</tr>
<tr>
<td>InformInfo</td>
<td>Set (for non-SM security traps)</td>
</tr>
<tr>
<td>GUIDInfoRecord</td>
<td>Set/Delete - this request can only be part of this set depending on the values of <code>sa_etm_allow_untrusted_guidinfo_rec</code> and <code>sa_etm_allow_guidinfo_rec_by_vf</code> - see elaboration below.</td>
</tr>
</tbody>
</table>

When `sa_etm_allow_untrusted_guidinfo_rec` is set to FALSE (and SAETM is enabled), the SA will drop GUIDInfoRecord Set/Delete untrusted requests.

When `sa_etm_allow_guidinfo_rec_by_vf` is set to FALSE (and SAETM is enabled), the SA will drop GUIDInfoRecord Set/Delete requests from non-physical ports.

If `sa_etm_allow_untrusted_guidinfo_rec`=FALSE, GUIDInfoRecord Set/Delete requests will become part of the SAETM set of untrusted requests allowed. Note that if `sa_etm_allow_guidinfo_rec_by_vf`=FALSE, the requests will only be allowed from physical ports.

### 13.1.8.1.2 Proxy SA Requests

SA modification request (SET/DELETE) is identified as a proxy operation when the port corresponding with the requester source address (SLID from LRH/SGID from GRH) is different than the port for which the request applies:

- For MCMemberRecord, when the MCMemberRecord.PortGID field does not match the requester address
- For ServiceRecord, when the ServiceRecord.ServiceGID field does not match requester address
• For the GUIDInfoRecord, when the LID field in the RID of the record does not match the requester address

When sa_etm_allow_untrusted_proxy_requests is set to FALSE and SAETM is enabled, untrusted proxy requests will be dropped.

### 13.1.8.1.3 Registration Limits

When any of `sa_etm_max_num_mcgs`, `sa_etm_max_num_srvcs` or `sa_etm_max_num_event_subs` parameters is set to 0, the number of this parameter’s registrations can be unlimited. When the parameter’s value is different than 0, attempting to exceed the maximum number of registrations will result in the request being silently dropped. Consequently, the requester and request info will be logged, and an event will be generated for the Activity Manager.

The following parameters control the maximum number of registrations:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>sa_etm_max_num_mcgs</code></td>
<td>Maximum number of multicast groups per port/vport that can be registered.</td>
</tr>
<tr>
<td><code>sa_etm_max_num_srvcs</code></td>
<td>Maximum number of service records per port/vport that can be registered.</td>
</tr>
<tr>
<td><code>sa_etm_max_num_event_subs</code></td>
<td>Maximum number of event subscriptions (InformInfo) per port/vport that can be registered.</td>
</tr>
</tbody>
</table>

### 13.1.8.1.4 SAETM Logging

When requesting an operation that is not part of the SAETM set of untrusted requests, it will be silently dropped and eventually written to the SM log.

The logging of the dropped MADs is repressed to not overload the OpenSM log. If the request that needs to be dropped was received from the same requester many times consecutively, OpenSM logs it only if the request number is part of the following sequence:

0, 1, 2, 5, 10, 20, 50, 100, 200… (similar to the trap log repression).

### 13.1.8.2 SGID Spoofing

SA can validate requester addresses by comparing the SLID and SGID of the incoming request. SA determines the requester port by the SLID and SGID field of the request. SGID spoofing is when the SGID and SLID do not match.

When `sa_check_sgid_spoofing` parameter is enabled, SA checks for SGID spoofing in every request that includes GRH, unless the SLID belongs to a router port in that same request. In case the request SGID does not match its SLID, the request will be dropped. The default value of this parameter is TRUE.
13.1.8.3 M_Key Authentication

13.1.8.3.1 M_Key Per Port

This feature increases protection on the fabric as a unique M_Key is generated and set for each HCA, router, or switch port.

OpenSM calculates an M_Key per port by performing a hash function on the port GUID of the device and the M_Key configured in opensm.conf.

To enable M_Key per port, set the parameter below in addition to the parameters listed in the previous section:

```
m_key_per_port TRUE
```

13.1.9 Appendix - SM Activity Report

SM can produce an activity report in a form of a dump file that details the different activities done in the SM. Activities are divided into subjects. The table below specifies the different activities currently supported in the SM activity report.

Reporting of each subject can be enabled individually using the configuration parameter `activity_report_subjects`:

- **Valid values:**
  - Comma-separated list of subjects to dump. The current supported subjects are:
    - "mc" - activity IDs 1, 2 and 8
    - "prtn" - activity IDs 3, 4, and 5
    - "virt" - activity IDs 6 and 7
    - "routing" - activity IDs 8-12
  - Two predefined values can be configured as well:
    - "all" - dump all subjects
    - "none" - disable the feature by dumping none of the subjects
    - Default value: "none"

13.1.9.1 SM Supported Activities

<table>
<thead>
<tr>
<th>Activity ID</th>
<th>Activity Name</th>
<th>Additional Fields</th>
<th>Comments</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>mcm_member</td>
<td>· MLid · MGid · Port Guid · Join State</td>
<td>Join state: 1 - Join -1 - Leave</td>
<td>Member joined/left MC group</td>
</tr>
<tr>
<td>2</td>
<td>mcg_change</td>
<td>· MLid · MGid · Change</td>
<td>Change: 0 - Create 1 - Delete</td>
<td>MC group created/deleted</td>
</tr>
<tr>
<td>3</td>
<td>prtn_guid_add</td>
<td>· Port Guid · PKey · Block index · Pkey Index</td>
<td>Guid added to partition</td>
<td></td>
</tr>
<tr>
<td>Activity ID</td>
<td>Activity Name</td>
<td>Additional Fields</td>
<td>Comments</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>---------------------</td>
<td>---------------------------</td>
<td>-----------------------------------</td>
<td>------------------------</td>
</tr>
<tr>
<td>4</td>
<td>prtn_create</td>
<td>- PKey</td>
<td></td>
<td>Partition created</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Prtn Name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>prtn_delete</td>
<td>- PKey</td>
<td>Delete Reason:</td>
<td>Partition deleted</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Delete Reason</td>
<td>0 - empty prtn</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1 - duplicate prtn</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2 - sm shutdown</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>port_virt_discover</td>
<td>- Port Guid</td>
<td></td>
<td>Port virtualization</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Top Index</td>
<td></td>
<td>discovered</td>
</tr>
<tr>
<td>7</td>
<td>vport_state_change</td>
<td>- Port Guid</td>
<td>VPort State:</td>
<td>Vport state changed</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- VPort Guid</td>
<td>1 - Down</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- VPort Index</td>
<td>2 - Init</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- VNode Guid</td>
<td>3 - ARMED</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- VPort State</td>
<td>4 - Active</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>mcg_tree_calc</td>
<td>- mlid</td>
<td></td>
<td>MCast group tree</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>calculated</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>routing_succeed</td>
<td>routing engine name</td>
<td></td>
<td>Routing done</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>successfully</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>routing_failed</td>
<td>routing engine name</td>
<td></td>
<td>Routing failed</td>
</tr>
<tr>
<td>11</td>
<td>ucast_cache_invalidated</td>
<td></td>
<td></td>
<td>ucast cache invalidated</td>
</tr>
<tr>
<td>12</td>
<td>ucast_cache_routing_done</td>
<td></td>
<td></td>
<td>ucast cache routing done</td>
</tr>
</tbody>
</table>

### 13.2 Appendix - Diagnostic Utilities

⚠️ For UFM-SDN Appliance, all the below diagnostics commands have ib prefix.

For example, for UFM-SDN Appliance, the command `ibstat` is `ib ibstat`.

### 13.2.1 InfiniBand Diagnostics Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ibstat</td>
<td>Shows the host adapters status.</td>
</tr>
<tr>
<td>ibstatus</td>
<td>Similar to ibstat but implemented as a script.</td>
</tr>
<tr>
<td>ibnetdiscover</td>
<td>Scans the topology.</td>
</tr>
<tr>
<td>ibaddr</td>
<td>Shows the LID range and default GID of the target (default is the local port).</td>
</tr>
<tr>
<td>ibroute</td>
<td>Displays unicast and multicast forwarding tables of the switches.</td>
</tr>
<tr>
<td>ibtracert</td>
<td>Displays unicast or multicast route from source to destination.</td>
</tr>
<tr>
<td>ibping</td>
<td>Uses vendor MADs to validate connectivity between InfiniBand nodes. On exit, (IP) ping-like output is shown.</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ibsysstat</td>
<td>Obtains basic information for the specific node which may be remote. This information includes: hostname, CPUs, memory utilization.</td>
</tr>
<tr>
<td>sminfo</td>
<td>Queries the SMInfo attribute on a node.</td>
</tr>
<tr>
<td>smpdump</td>
<td>A general purpose SMP utility which gets SM attributes from a specified SMA. The result is dumped in hex by default.</td>
</tr>
<tr>
<td>smpquery</td>
<td>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.</td>
</tr>
<tr>
<td>perfquery</td>
<td>Dumps (and optionally clears) the performance counters of the destination port (including error counters).</td>
</tr>
<tr>
<td>ibswitches</td>
<td>Scans the net or uses existing net topology file and lists all switches.</td>
</tr>
<tr>
<td>ibhosts</td>
<td>Scans the net or uses existing net topology file and lists all hosts.</td>
</tr>
<tr>
<td>ibnodes</td>
<td>Scans the net or uses existing net topology file and lists all nodes.</td>
</tr>
<tr>
<td>ibportstate</td>
<td>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.</td>
</tr>
<tr>
<td>saquery</td>
<td>Issues SA queries.</td>
</tr>
<tr>
<td>ibdiagnet</td>
<td>ibdiagnet scans the fabric using directed route packets and extracts all the available information regarding its connectivity and devices.</td>
</tr>
<tr>
<td>ibnetsplit</td>
<td>Automatically groups hosts and creates scripts that can be run to split the network into sub-networks each containing one group of hosts.</td>
</tr>
<tr>
<td>ibqueryerrors</td>
<td>Queries IB spec-defined errors from all fabric ports. Note: This tool can change reset port counters Should be used with caution.</td>
</tr>
<tr>
<td>smparquery</td>
<td>Queries adaptive-routing related settings from a particular switch. Note: This tool can change reset port counters Should be used with caution.</td>
</tr>
</tbody>
</table>

### 13.2.2 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.

### 13.2.2.1 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).

### 13.2.3 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(list_of_cas)] [-p(port_list)] [-short] <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:**

- `-m`  multicast path tracing
-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]
s2vl <addr> [portnum]
vlnb <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.

⚠️ 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
- `RxSL (-X)` Shows Rx SL port counters
- `RcvSL (-S)` Shows Rcv SL port counters
- `RxDisc (-D)` Shows Rx Discard Details
- `rcvErr, (-E)` Shows Rcv Error Details
- `extended_speeds` Shows port extended speeds counters
- `opRcvCounters` Shows Rcv Counters per Op code
- `flowCtrlCounters` Shows flow control counters
- `vloopCounters` Shows packets received per Op code per VL
- `vloopData` Shows data received per Op code per VL
- `vloopFlowErr` Shows flow control update errors per VL
- `vlooptCounts` Shows ticks waiting to transmit counters per VL
- `vloopRcv` Shows Rcv congestion control counters
- `vloopFecn` Shows SL Rcv FECN counters
- `vloopBecn` Shows SL Rcv BECN counters
- `vloopTcp` Shows SL Tcp congestion control counters
- `vloopTimeC` Shows VL Xmit Time congestion control counters
- `loop_ports (-l)` Iterated 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 # reset performance counters of lid 32 port 1 only
perfquery -R 32 1 # 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:

1 Lists connected nodes
2 Lists connected switches
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, vs, mtu, lid, smlid, lmc, mkey, mkeylease, mkeyprot

Examples:

```
ibportstate 3 1 disable # by lid
ibportstate -D 0x2C90001800051 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
- --report-port, -r report port link information
- --threshold-file <val> specify the alternate threshold file, default: /etc/infiniband-diags/error_thresholds
- --skip-sl don't obtain SL to all destinations
- --switch print data for switch only
- --ca print data for CA's only
- --router print data for routers only
- --details include transmit discard details
- --counters print counters only
- --clear-errors, -k Clear error counters after read
- --clear-counts, -K Clear data counters after read
- --load-cache <file> file name of ibnetdiscover cache to load
- --outstanding_smps, -o <val> specify the number of outstanding SMP's which should be issued during the scan
- --config, -c <configfile> 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
- --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]
PortsToLFTMap (PLFTM) <addr> [plft] [control_map]
RMbsGroupDirectionTable (DIRT) <addr> [blocknum]
RNGenStringTable (GSTR) <addr> [plft] [blocknum]
RNGenBySubGroupPriority (GSGP) <addr>
RNRcvString (RSTR) <addr> [blocknum]
RNXmitPortMask (RNXM) <addr> [blocknum]
PortRNCounters (RNPC) <addr>

Options:
- Ca, -C ca Ca name to use
- P, -P port Ca port number to use
- Direct, -D <dr_path> use Direct address argument
- Lid, -L Lid address argument
- help, -h help message
- V, -V show version
- d, -d Print debug logs
saquery

Issues SA queries.

Syntax

saquery [-h -d -N -L -G -s -g][-I][-m]<name>

Queries node records by default.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>d</td>
<td>Enables debugging</td>
</tr>
<tr>
<td>P</td>
<td>Gets PathRecord info</td>
</tr>
<tr>
<td>N</td>
<td>Gets NodeRecord info</td>
</tr>
<tr>
<td>L (-L)</td>
<td>Returns just the Lid of the name specified</td>
</tr>
<tr>
<td>G (-G)</td>
<td>Returns just the Guid of the name specified</td>
</tr>
<tr>
<td>S (-S)</td>
<td>Returns the PortInfoRecords with isSM capability mask bit on</td>
</tr>
<tr>
<td>G (-g)</td>
<td>Gets multicast group info</td>
</tr>
<tr>
<td>L (-l)</td>
<td>Returns the unique Lid of the name specified</td>
</tr>
<tr>
<td>O (-O)</td>
<td>Returns name for the Lid specified</td>
</tr>
<tr>
<td>m (-m)</td>
<td>Gets multicast member info (if multicast group specified, list member GIDs only for group specified for example 'saquery -m 0xC000')</td>
</tr>
<tr>
<td>x (-x)</td>
<td>Gets ServiceRecord info</td>
</tr>
<tr>
<td>c (-c)</td>
<td>Gets InformInfoRecord (subscription) info</td>
</tr>
<tr>
<td>S (-S)</td>
<td>Gets the SA's class port info</td>
</tr>
<tr>
<td>list (-d)</td>
<td>Gets LinkRecord info</td>
</tr>
<tr>
<td>src-to-dst (<a href="">src:dst</a>)</td>
<td>Gets the node desc of the CA's</td>
</tr>
<tr>
<td>sgid-to-dgid (&lt;sgid-dgid&gt;)</td>
<td>Gets a PathRecord for <a href="">src:dst</a> where src and dst are either node names or LIDs</td>
</tr>
<tr>
<td>node-name-map</td>
<td>Gets a PathRecord for &lt;sgid-dgid&gt; where sgid and dgid are addresses in IPv6 format</td>
</tr>
<tr>
<td>smkey &lt;val&gt;</td>
<td>Specifies a node name map file</td>
</tr>
<tr>
<td>slid &lt;lid&gt;</td>
<td>SA SM_Key value for the query. If non-numeric value (like 'x') is specified then saquery will prompt for a value. Default 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 &quot;untrusted&quot;)</td>
</tr>
<tr>
<td>dlid &lt;lid&gt;</td>
<td>Source LID (PathRecord)</td>
</tr>
<tr>
<td>mld &lt;lid&gt;</td>
<td>Destination LID (PathRecord)</td>
</tr>
<tr>
<td>sgid &lt;gid&gt;</td>
<td>Multicast LID (MCMemberRecord)</td>
</tr>
<tr>
<td>dgid &lt;gid&gt;</td>
<td>Multicast GID (MCMemberRecord)</td>
</tr>
<tr>
<td>gid &lt;gid&gt;</td>
<td>Destination GID (IPv6 format) (PathRecord)</td>
</tr>
<tr>
<td>mgid &lt;gid&gt;</td>
<td>Port GID (MCMemberRecord)</td>
</tr>
<tr>
<td>mkey &lt;val&gt;</td>
<td>Port key: P_Key (PathRecord, MCMemberRecord).</td>
</tr>
<tr>
<td>mtu : (-M)</td>
<td>QoS Class (PathRecord)</td>
</tr>
<tr>
<td>rate (-R)</td>
<td>Service level (PathRecord, MCMemberRecord)</td>
</tr>
<tr>
<td>pkt_lifetime</td>
<td>Rate and selector (PathRecord, MCMemberRecord)</td>
</tr>
<tr>
<td>qkey (-q)</td>
<td>Packet lifetime and selector (PathRecord, MCMemberRecord)</td>
</tr>
<tr>
<td>qos_class (-Q)</td>
<td>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 &quot;untrusted&quot;)</td>
</tr>
<tr>
<td>tclass (-T)</td>
<td>Source GID (IPv6 format) (PathRecord)</td>
</tr>
<tr>
<td>flow_label : (-F)</td>
<td>Destination GID (IPv6 format) (PathRecord)</td>
</tr>
<tr>
<td>hop_limit : (-H)</td>
<td>Port GID (MCMemberRecord)</td>
</tr>
<tr>
<td>scope</td>
<td>Multicast GID (MCMemberRecord)</td>
</tr>
<tr>
<td>join_state (-J)</td>
<td>Reversible path (PathRecord)</td>
</tr>
<tr>
<td>proxy_join (-X)</td>
<td>Number of paths (PathRecord)</td>
</tr>
<tr>
<td>service_id</td>
<td>QoS Class (PathRecord)</td>
</tr>
</tbody>
</table>

Dependencies: OpenSM libvendor, OpenSM libopensm, libibumad

ibsysstat

ibsysstat [options] <dest lid|guid> [<op>]

Nonstandard flags:

Current supported operations:
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:**

  \[ \text{ibnetsplit} \ [-v][-h][-g \text{grp-file}] \ [-s \text{.lst|.net|.topo}] \ [-r \text{head-ports|-d max-dist}] \]

- **Split:**

  \[ \text{ibnetsplit} \ [-v][-h][-g \text{grp-file}] \ [-s \text{.lst|.net|.topo}] \ [-o \text{out-dir}] \]

- **Combined:**

  \[ \text{ibnetsplit} \ [-v][-h][-g \text{grp-file}] \ [-s \text{.lst|.net|.topo}] \ [-r \text{head-ports|-d max-dist}] \ [-o \text{out-dir}] \]

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.

  \[ \text{This method of analysis may not be suitable for some topologies.} \]

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.f dbs" - A dump of unicast forwarding tables of the fabric switches.
- "ibdiagnet2.mcf dbs" - A dump of multicast forwarding tables of the fabric switches.
- "ibdiagnet2.slv l" - 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.vl2vl" - A dump of VL to VL configuration of the fabric switches.

Load plugins from:
```
/tmphutils2/share/ibdiagnet2.1.1/plugins/
```

You can specify additional paths to be looked in with "IBDIAGNET_PLUGINS_PATH" env variable.
<table>
<thead>
<tr>
<th>Plugin Name</th>
<th>Result</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>libibdiagnet_cable_diag_plugin-2.1.1</td>
<td>Succeeded</td>
<td>Plugin loaded</td>
</tr>
<tr>
<td>libibdiagnet_phy_diag_plugin-2.1.1</td>
<td>Succeeded</td>
<td>Plugin loaded</td>
</tr>
</tbody>
</table>

Syntax

```
[-i|--device <dev-name>] [-p|--port <port-num>]
[-g|--guid <GUID in hex>] [-g|--guid <GUID in hex>]
--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).
--ac                         : Provides a report of Mellanox counters
--scr                        : Reset all the Mellanox counters (if -ac option selected).
--pc                         : Reset all the fabric PM counters.
[-P|--counter <<PM>=<value>>      : If any of the provided PM is greater than 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 exceeded 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).
--mgm_window <num>               : Max mgm MADs on wire. (default=8).
--mgm_window <num>               : Max mgm 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. Indiaget will use this mapping for
```

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 the 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).
- `--ac` : Provides a report of Mellanox counters
- `--scr` : Reset all the Mellanox counters (if -ac option selected).
- `--pc` : Reset all the fabric PM counters.
- `[-P|--counter <<PM>=<value>>      : If any of the provided PM is greater than 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 exceeded 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).
- `--mgm_window <num>` : Max mgm MADs on wire. (default=8).
- `--mgm_window <num>` : Max mgm 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. Indiaget will use this mapping for

454
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 variable.

<table>
<thead>
<tr>
<th>Plugin Name</th>
<th>Result</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>libibdiagnet_cable_diag_plugin-2.1.1</td>
<td>Succeeded</td>
<td>Plugin loaded</td>
</tr>
<tr>
<td>libibdiagnet_phy_diag_plugin-2.1.1</td>
<td>Succeeded</td>
<td>Plugin loaded</td>
</tr>
</tbody>
</table>

Syntax

[-i|--device <dev-name>] [-p|--port <port-num>]
[-g|--guid <GUID in hex>] [-s|--skip <stage>]
[-s|--skip <library name>] [-r|--remote]
[-s|--skip <stage>] [-p|--port <port-num>]
[-l|--lib_name <library name>]
[-b|--base <base test>]
[-c|--counter <PM>]
[-t|--test <test>]

Options

[-i|--device <dev-name>] [-p|--port <port-num>]
[-g|--guid <GUID in hex>] [-s|--skip <stage>]
[-s|--skip <library name>] [-r|--remote]
[-s|--skip <stage>] [-p|--port <port-num>]
[-l|--lib_name <library name>]
[-b|--base <base test>]
[-c|--counter <PM>]
[-t|--test <test>]

Cable Diagnostic (Plugin)
[get_cable_info] [--cable_info_disconnected]

Phy Diagnostic (Plugin)
[get_phy_info] [--reset_phy_info]
-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
:sList 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_compact_db <version_sub_version>
--V|--version
--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

: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).
:sList all counters per lane (when available).
: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: 0x8-9a-fa-f) *name). Mapping file can also be specified by environment variable "IBUTILS_NODE_NAME_MAP_FILE_PATH".
:source lid
:destination 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.
:Indicates to query all QSFP cables for cable information.
:Indicates to clear all ports phy information.
13.3 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)

13.3.1 InfiniBand Port Counters

The following tables list and describe the port counters and events currently supported:

- InfiniBand Port Counters
- Calculated Port Counters

<table>
<thead>
<tr>
<th>InfiniBand Port Counters</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Counter</strong></td>
</tr>
<tr>
<td>Xmit Data (in bytes)</td>
</tr>
<tr>
<td>Rcv Data (in bytes)</td>
</tr>
<tr>
<td>Xmit Packets</td>
</tr>
<tr>
<td>Rcv Packets</td>
</tr>
</tbody>
</table>
| 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)
  - Packets 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. |
### InfiniBand Port Counters

<table>
<thead>
<tr>
<th>Counter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Symbol Errors</td>
<td>Total number of minor link errors detected on one or more physical lanes.</td>
</tr>
<tr>
<td>Link Error Recovery</td>
<td>Total number of times the Port Training state machine has successfully completed the link error recovery process.</td>
</tr>
<tr>
<td>Link Error Downed</td>
<td>Total number of times the Port Training state machine has failed the link error recovery process and downed the link.</td>
</tr>
<tr>
<td>Local Integrity Error</td>
<td>The number of times that the count of local physical errors exceeded the threshold specified by LocalPhyErrors.</td>
</tr>
<tr>
<td>Rcv Remote Physical Error</td>
<td>Total number of packets marked with the EBP delimiter received on the port.</td>
</tr>
<tr>
<td>Xmit Constraint Error</td>
<td>Total number of packets not transmitted from the switch physical port for the following reasons:</td>
</tr>
<tr>
<td></td>
<td>• FilterRawOutbound is true and packet is raw</td>
</tr>
<tr>
<td></td>
<td>• PartitionEnforcementOutbound is true and packet fails partition key check or IP version check</td>
</tr>
<tr>
<td>Rcv Constraint Error</td>
<td>Total number of packets received on the switch physical port that are discarded for the following reasons:</td>
</tr>
<tr>
<td></td>
<td>• FilterRawInbound is true and packet is raw</td>
</tr>
<tr>
<td></td>
<td>• PartitionEnforcementInbound is true and packet fails partition key check or IP version check</td>
</tr>
<tr>
<td>Excess Buffer Overrun Error</td>
<td>The number of times that OverrunErrors consecutive flow control update periods occurred, each having at least one overrun error</td>
</tr>
<tr>
<td>Rcv Switch Relay Error</td>
<td>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:</td>
</tr>
<tr>
<td></td>
<td>• DLID mapping</td>
</tr>
<tr>
<td></td>
<td>• VL mapping</td>
</tr>
<tr>
<td></td>
<td>• Looping (output port = input port)</td>
</tr>
<tr>
<td>VL15 Dropped</td>
<td>Number of incoming VL15 packets dropped because of resource limitations (e.g., lack of buffers) in the port</td>
</tr>
<tr>
<td>XmitWait</td>
<td>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.</td>
</tr>
</tbody>
</table>

### InfiniBand Calculated Port Counters

<table>
<thead>
<tr>
<th>Counter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normalized XmitData</td>
<td>Effective port bandwidth utilization in % XmitData incremental/ Link Capacity</td>
</tr>
<tr>
<td>Normalized Congested Bandwidth</td>
<td>Amount of bandwidth that was suppressed due to congestion (XmitWait incremental/ Time) * Link Capacity Separate counters are used for Tier 4 ports and for the rest of the ports.</td>
</tr>
</tbody>
</table>
13.3.2 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.

<table>
<thead>
<tr>
<th>Alarm ID</th>
<th>Alarm Name</th>
<th>To Log</th>
<th>Alarm Severity</th>
<th>Default Severity</th>
<th>Default Threshold</th>
<th>Default TTL</th>
<th>Related Object</th>
<th>Category</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>64</td>
<td>GID Address In Service</td>
<td>1</td>
<td>0</td>
<td>Info</td>
<td>1</td>
<td>0</td>
<td>Port</td>
<td>Fabric Notification</td>
<td>SM</td>
</tr>
<tr>
<td>65</td>
<td>GID Address Out of Service</td>
<td>1</td>
<td>0</td>
<td>Warning</td>
<td>1</td>
<td>0</td>
<td>Port</td>
<td>Fabric Notification</td>
<td>SM</td>
</tr>
<tr>
<td>66</td>
<td>New MCast Group Created</td>
<td>1</td>
<td>0</td>
<td>Info</td>
<td>1</td>
<td>0</td>
<td>Port</td>
<td>Fabric Notification</td>
<td>SM</td>
</tr>
<tr>
<td>67</td>
<td>MCast Group Deleted</td>
<td>1</td>
<td>0</td>
<td>Info</td>
<td>1</td>
<td>0</td>
<td>Port</td>
<td>Fabric Notification</td>
<td>SM</td>
</tr>
<tr>
<td>110</td>
<td>Symbol Error</td>
<td>1</td>
<td>1</td>
<td>Warning</td>
<td>200</td>
<td>0</td>
<td>Port</td>
<td>Hardware</td>
<td>Telemetry</td>
</tr>
<tr>
<td>111</td>
<td>Link Error Recovery</td>
<td>1</td>
<td>1</td>
<td>Minor</td>
<td>1</td>
<td>0</td>
<td>Port</td>
<td>Hardware</td>
<td>Telemetry</td>
</tr>
<tr>
<td>112</td>
<td>Link Downed</td>
<td>1</td>
<td>1</td>
<td>Critical</td>
<td>0</td>
<td>0</td>
<td>Port</td>
<td>Hardware</td>
<td>Telemetry</td>
</tr>
<tr>
<td>113</td>
<td>Port Receive Errors</td>
<td>1</td>
<td>1</td>
<td>Minor</td>
<td>5</td>
<td>0</td>
<td>Port</td>
<td>Hardware</td>
<td>Telemetry</td>
</tr>
<tr>
<td>114</td>
<td>Port Receive Remote Physical Errors</td>
<td>1</td>
<td>0</td>
<td>Minor</td>
<td>5</td>
<td>0</td>
<td>Port</td>
<td>Hardware</td>
<td>Telemetry</td>
</tr>
<tr>
<td>115</td>
<td>Port Receive Switch Relay Errors</td>
<td>1</td>
<td>1</td>
<td>Minor</td>
<td>50</td>
<td>0</td>
<td>Port</td>
<td>Fabric Configuration</td>
<td>Telemetry</td>
</tr>
<tr>
<td>116</td>
<td>Port Xmit Discards</td>
<td>1</td>
<td>1</td>
<td>Minor</td>
<td>200</td>
<td>0</td>
<td>Port</td>
<td>Communication Error</td>
<td>Telemetry</td>
</tr>
<tr>
<td>117</td>
<td>Port Xmit Constraint Errors</td>
<td>1</td>
<td>1</td>
<td>Minor</td>
<td>1</td>
<td>0</td>
<td>Port</td>
<td>Communication Error</td>
<td>Telemetry</td>
</tr>
<tr>
<td>118</td>
<td>Port Receive Constraint Errors</td>
<td>1</td>
<td>1</td>
<td>Minor</td>
<td>1</td>
<td>0</td>
<td>Port</td>
<td>Communication Error</td>
<td>Telemetry</td>
</tr>
<tr>
<td>119</td>
<td>Local Link Integrity Errors</td>
<td>1</td>
<td>1</td>
<td>Minor</td>
<td>5</td>
<td>0</td>
<td>Port</td>
<td>Hardware</td>
<td>Telemetry</td>
</tr>
<tr>
<td>120</td>
<td>Excessive Buffer Overrun Errors</td>
<td>1</td>
<td>1</td>
<td>Minor</td>
<td>1</td>
<td>0</td>
<td>Port</td>
<td>Communication Error</td>
<td>Telemetry</td>
</tr>
<tr>
<td>121</td>
<td>VL15 Dropped</td>
<td>1</td>
<td>1</td>
<td>Minor</td>
<td>500</td>
<td>0</td>
<td>Port</td>
<td>Communication Error</td>
<td>Telemetry</td>
</tr>
<tr>
<td>Alarm ID</td>
<td>Alarm Name</td>
<td>To Log</td>
<td>Alarm Severity</td>
<td>Default Threshold</td>
<td>Default TTL</td>
<td>Related Object</td>
<td>Category</td>
<td>Source</td>
<td></td>
</tr>
<tr>
<td>----------</td>
<td>-------------------------------------</td>
<td>--------</td>
<td>----------------</td>
<td>-------------------</td>
<td>-------------</td>
<td>----------------</td>
<td>----------</td>
<td>-----------------</td>
<td></td>
</tr>
<tr>
<td>122</td>
<td>Congested Bandwidth (%) Threshold Reached</td>
<td>1</td>
<td>Minor</td>
<td>10</td>
<td>0</td>
<td>Port</td>
<td>Hardware</td>
<td>Telemetry</td>
<td></td>
</tr>
<tr>
<td>123</td>
<td>Port Bandwidth (%) Threshold Reached</td>
<td>1</td>
<td>Minor</td>
<td>95</td>
<td>0</td>
<td>Port</td>
<td>Communication Error</td>
<td>Telemetry</td>
<td></td>
</tr>
<tr>
<td>130</td>
<td>Non-optimal link width</td>
<td>1</td>
<td>Minor</td>
<td>1</td>
<td>0</td>
<td>Port</td>
<td>Hardware</td>
<td>SM</td>
<td></td>
</tr>
<tr>
<td>134</td>
<td>T4 Port Congested Bandwidth</td>
<td>1</td>
<td>Warning</td>
<td>10</td>
<td>0</td>
<td>Port</td>
<td>Communication Error</td>
<td>Telemetry</td>
<td></td>
</tr>
<tr>
<td>141</td>
<td>Flow Control Update Watchdog Timer Expired</td>
<td>1</td>
<td>Warning</td>
<td>1</td>
<td>0</td>
<td>Port</td>
<td>Hardware</td>
<td>SM</td>
<td></td>
</tr>
<tr>
<td>144</td>
<td>Capability Mask Modified</td>
<td>1</td>
<td>Info</td>
<td>1</td>
<td>0</td>
<td>Port</td>
<td>Fabric Notification</td>
<td>SM</td>
<td></td>
</tr>
<tr>
<td>145</td>
<td>System Image GUID changed</td>
<td>1</td>
<td>Info</td>
<td>1</td>
<td>0</td>
<td>Port</td>
<td>Communication Error</td>
<td>SM</td>
<td></td>
</tr>
<tr>
<td>156</td>
<td>Link Speed Enforcement Disabled</td>
<td>1</td>
<td>Critical</td>
<td>0</td>
<td>0</td>
<td>Site</td>
<td>Fabric Notification</td>
<td>SM</td>
<td></td>
</tr>
<tr>
<td>250</td>
<td>Running in Limited Mode</td>
<td>1</td>
<td>Critical</td>
<td>1</td>
<td>0</td>
<td>Grid</td>
<td>Maintenance</td>
<td>Licensing</td>
<td></td>
</tr>
<tr>
<td>251</td>
<td>Switching to Limited Mode</td>
<td>1</td>
<td>Critical</td>
<td>1</td>
<td>0</td>
<td>Grid</td>
<td>Maintenance</td>
<td>Licensing</td>
<td></td>
</tr>
<tr>
<td>252</td>
<td>License Expired</td>
<td>1</td>
<td>Warning</td>
<td>1</td>
<td>0</td>
<td>Grid</td>
<td>Maintenance</td>
<td>Licensing</td>
<td></td>
</tr>
<tr>
<td>253</td>
<td>Duplicated licenses</td>
<td>1</td>
<td>Critical</td>
<td>1</td>
<td>0</td>
<td>Grid</td>
<td>Maintenance</td>
<td>Licensing</td>
<td></td>
</tr>
<tr>
<td>254</td>
<td>License Limit Exceeded</td>
<td>1</td>
<td>Critical</td>
<td>1</td>
<td>0</td>
<td>Grid</td>
<td>Maintenance</td>
<td>Licensing</td>
<td></td>
</tr>
<tr>
<td>255</td>
<td>License is About to Expire</td>
<td>1</td>
<td>Warning</td>
<td>1</td>
<td>0</td>
<td>Grid</td>
<td>Maintenance</td>
<td>Licensing</td>
<td></td>
</tr>
<tr>
<td>256</td>
<td>Bad M_Key</td>
<td>1</td>
<td>Minor</td>
<td>1</td>
<td>0</td>
<td>Port</td>
<td>Security</td>
<td>SM</td>
<td></td>
</tr>
<tr>
<td>257</td>
<td>Bad P_Key</td>
<td>1</td>
<td>Minor</td>
<td>1</td>
<td>0</td>
<td>Port</td>
<td>Security</td>
<td>SM</td>
<td></td>
</tr>
<tr>
<td>258</td>
<td>Bad Q_Key</td>
<td>1</td>
<td>Minor</td>
<td>1</td>
<td>0</td>
<td>Port</td>
<td>Security</td>
<td>SM</td>
<td></td>
</tr>
<tr>
<td>259</td>
<td>Bad P_Key Switch External Port</td>
<td>1</td>
<td>Critical</td>
<td>1</td>
<td>0</td>
<td>Port</td>
<td>Security</td>
<td>SM</td>
<td></td>
</tr>
<tr>
<td>328</td>
<td>Link is Up</td>
<td>1</td>
<td>Info</td>
<td>1</td>
<td>10</td>
<td>Link</td>
<td>Fabric Topology</td>
<td>SM</td>
<td></td>
</tr>
<tr>
<td>329</td>
<td>Link is Down</td>
<td>1</td>
<td>Warning</td>
<td>1</td>
<td>10</td>
<td>Site</td>
<td>Fabric Topology</td>
<td>SM</td>
<td></td>
</tr>
<tr>
<td>331</td>
<td>Node is Down</td>
<td>1</td>
<td>Warning</td>
<td>1</td>
<td>0</td>
<td>Site</td>
<td>Fabric Topology</td>
<td>SM</td>
<td></td>
</tr>
<tr>
<td>Alarm ID</td>
<td>Alarm Name</td>
<td>To Log</td>
<td>Alarm</td>
<td>Default Severity</td>
<td>Default Threshold</td>
<td>Default TTL</td>
<td>Related Object</td>
<td>Category</td>
<td>Source</td>
</tr>
<tr>
<td>----------</td>
<td>-----------------------------------------</td>
<td>--------</td>
<td>-------</td>
<td>------------------</td>
<td>-------------------</td>
<td>-------------</td>
<td>----------------</td>
<td>----------------</td>
<td>--------</td>
</tr>
<tr>
<td>332</td>
<td>Node is Up</td>
<td>1</td>
<td>0</td>
<td>Info</td>
<td>1</td>
<td>0</td>
<td>Site</td>
<td>Fabric Topology</td>
<td>SM</td>
</tr>
<tr>
<td>336</td>
<td>Port Action Succeeded</td>
<td>1</td>
<td>0</td>
<td>Info</td>
<td>1</td>
<td>0</td>
<td>Port</td>
<td>Maintenance</td>
<td>UFM</td>
</tr>
<tr>
<td>337</td>
<td>Port Action Failed</td>
<td>1</td>
<td>0</td>
<td>Minor</td>
<td>1</td>
<td>0</td>
<td>Port</td>
<td>Maintenance</td>
<td>UFM</td>
</tr>
<tr>
<td>338</td>
<td>Device Action Succeeded</td>
<td>1</td>
<td>0</td>
<td>Info</td>
<td>1</td>
<td>0</td>
<td>Port</td>
<td>Maintenance</td>
<td>UFM</td>
</tr>
<tr>
<td>339</td>
<td>Device Action Failed</td>
<td>1</td>
<td>0</td>
<td>Minor</td>
<td>1</td>
<td>0</td>
<td>Port</td>
<td>Maintenance</td>
<td>UFM</td>
</tr>
<tr>
<td>344</td>
<td>Partial Switch ASIC Failure</td>
<td>1</td>
<td>1</td>
<td>Critical</td>
<td>1</td>
<td>0</td>
<td>Switch</td>
<td>Maintenance</td>
<td>UFM</td>
</tr>
<tr>
<td>352</td>
<td>Network Added</td>
<td>1</td>
<td>0</td>
<td>Info</td>
<td>1</td>
<td>0</td>
<td>Network</td>
<td>Logical Model</td>
<td>UFM</td>
</tr>
<tr>
<td>353</td>
<td>Network Removed</td>
<td>1</td>
<td>0</td>
<td>Info</td>
<td>1</td>
<td>0</td>
<td>Network</td>
<td>Logical Model</td>
<td>UFM</td>
</tr>
<tr>
<td>380</td>
<td>Switch Upgrade Error</td>
<td>1</td>
<td>1</td>
<td>Critical</td>
<td>1</td>
<td>0</td>
<td>Switch</td>
<td>Maintenance</td>
<td>UFM</td>
</tr>
<tr>
<td>381</td>
<td>Switch Upgrade Failed</td>
<td>1</td>
<td>0</td>
<td>Info</td>
<td>1</td>
<td>0</td>
<td>Switch</td>
<td>Maintenance</td>
<td>UFM</td>
</tr>
<tr>
<td>382</td>
<td>Module status NOT PRESENT</td>
<td>1</td>
<td>1</td>
<td>Warning</td>
<td>1</td>
<td>0</td>
<td>Switch</td>
<td>Module Status</td>
<td>UFM</td>
</tr>
<tr>
<td>383</td>
<td>Host Upgrade Failed</td>
<td>1</td>
<td>0</td>
<td>Info</td>
<td>1</td>
<td>0</td>
<td>Computer</td>
<td>Maintenance</td>
<td>UFM</td>
</tr>
<tr>
<td>384</td>
<td>Switch Module Powered Off</td>
<td>1</td>
<td>1</td>
<td>Info</td>
<td>1</td>
<td>0</td>
<td>Switch</td>
<td>Module Status</td>
<td>UFM</td>
</tr>
<tr>
<td>385</td>
<td>Switch FW Upgrade Started</td>
<td>1</td>
<td>0</td>
<td>Info</td>
<td>1</td>
<td>0</td>
<td>Switch</td>
<td>Maintenance</td>
<td>UFM</td>
</tr>
<tr>
<td>386</td>
<td>Switch SW Upgrade Started</td>
<td>1</td>
<td>0</td>
<td>Info</td>
<td>1</td>
<td>0</td>
<td>Switch</td>
<td>Maintenance</td>
<td>UFM</td>
</tr>
<tr>
<td>387</td>
<td>Switch Upgrade Finished</td>
<td>1</td>
<td>0</td>
<td>Info</td>
<td>1</td>
<td>0</td>
<td>Switch</td>
<td>Maintenance</td>
<td>UFM</td>
</tr>
<tr>
<td>388</td>
<td>Host FW Upgrade Started</td>
<td>1</td>
<td>0</td>
<td>Info</td>
<td>1</td>
<td>0</td>
<td>Computer</td>
<td>Maintenance</td>
<td>UFM</td>
</tr>
<tr>
<td>389</td>
<td>Host OFED Upgrade Started</td>
<td>1</td>
<td>0</td>
<td>Info</td>
<td>1</td>
<td>0</td>
<td>Computer</td>
<td>Maintenance</td>
<td>UFM</td>
</tr>
<tr>
<td>391</td>
<td>Switch Module Removed</td>
<td>1</td>
<td>0</td>
<td>Info</td>
<td>1</td>
<td>0</td>
<td>Switch</td>
<td>Fabric Notification</td>
<td>Switch</td>
</tr>
<tr>
<td>392</td>
<td>Module Temperature Threshold Reached</td>
<td>1</td>
<td>0</td>
<td>Info</td>
<td>60</td>
<td>0</td>
<td>Module</td>
<td>Hardware</td>
<td>Switch</td>
</tr>
<tr>
<td>Alarm ID</td>
<td>Alarm Name</td>
<td>To Log</td>
<td>Alarm Severity</td>
<td>Default Severity</td>
<td>Default Threshold</td>
<td>Default TTL</td>
<td>Related Object</td>
<td>Category</td>
<td>Source</td>
</tr>
<tr>
<td>----------</td>
<td>------------------------------------------------</td>
<td>--------</td>
<td>----------------</td>
<td>------------------</td>
<td>-------------------</td>
<td>-------------</td>
<td>----------------</td>
<td>----------------</td>
<td>------------</td>
</tr>
<tr>
<td>393</td>
<td>Switch Module Added</td>
<td>1</td>
<td>Info</td>
<td>1</td>
<td>0</td>
<td>Switch</td>
<td>Fabric Notification</td>
<td>Switch</td>
<td></td>
</tr>
<tr>
<td>394</td>
<td>Module Status FAULT</td>
<td>1</td>
<td>Critical</td>
<td>1</td>
<td>0</td>
<td>Switch</td>
<td>Module Status</td>
<td>Switch</td>
<td></td>
</tr>
<tr>
<td>395</td>
<td>Device Action Started</td>
<td>1</td>
<td>Info</td>
<td>1</td>
<td>0</td>
<td>Port</td>
<td>Maintenance</td>
<td>UFM</td>
<td></td>
</tr>
<tr>
<td>396</td>
<td>Site Action Started</td>
<td>1</td>
<td>Info</td>
<td>1</td>
<td>0</td>
<td>Port</td>
<td>Maintenance</td>
<td>UFM</td>
<td></td>
</tr>
<tr>
<td>397</td>
<td>Site Action Failed</td>
<td>1</td>
<td>Minor</td>
<td>1</td>
<td>0</td>
<td>Port</td>
<td>Maintenance</td>
<td>UFM</td>
<td></td>
</tr>
<tr>
<td>398</td>
<td>Switch Chip Added</td>
<td>1</td>
<td>Info</td>
<td>1</td>
<td>0</td>
<td>Switch</td>
<td>Fabric Notification</td>
<td>Switch</td>
<td></td>
</tr>
<tr>
<td>399</td>
<td>Switch Chip Removed</td>
<td>1</td>
<td>Critical</td>
<td>1</td>
<td>0</td>
<td>Switch</td>
<td>Fabric Notification</td>
<td>Switch</td>
<td></td>
</tr>
<tr>
<td>403</td>
<td>Device Pending Reboot</td>
<td>1</td>
<td>Warning</td>
<td>0</td>
<td>0</td>
<td>Device</td>
<td>Maintenance</td>
<td>UFM</td>
<td></td>
</tr>
<tr>
<td>404</td>
<td>System Information is missing</td>
<td>1</td>
<td>Warning</td>
<td>1</td>
<td>0</td>
<td>Switch</td>
<td>Communication Error</td>
<td>UFM</td>
<td></td>
</tr>
<tr>
<td>405</td>
<td>Switch Identity Validation Failed</td>
<td>1</td>
<td>Warning</td>
<td>1</td>
<td>0</td>
<td>Switch</td>
<td>Communication Error</td>
<td>UFM</td>
<td></td>
</tr>
<tr>
<td>406</td>
<td>Switch System Information is missing</td>
<td>1</td>
<td>Warning</td>
<td>1</td>
<td>0</td>
<td>Switch</td>
<td>Communication Error</td>
<td>UFM</td>
<td></td>
</tr>
<tr>
<td>407</td>
<td>COMEX Ambient Temperature Threshold Reached</td>
<td>1</td>
<td>Minor</td>
<td>60</td>
<td>0</td>
<td>Switch</td>
<td>Hardware</td>
<td>Switch</td>
<td></td>
</tr>
<tr>
<td>408</td>
<td>Switch is Unresponsive</td>
<td>1</td>
<td>Critical</td>
<td>1</td>
<td>0</td>
<td>Switch</td>
<td>Communication Error</td>
<td>UFM</td>
<td></td>
</tr>
<tr>
<td>502</td>
<td>Device Upgrade Finished</td>
<td>1</td>
<td>Info</td>
<td>1</td>
<td>0</td>
<td>Device</td>
<td>Maintenance</td>
<td>UFM</td>
<td></td>
</tr>
<tr>
<td>506</td>
<td>Device Upgrade Finished</td>
<td>1</td>
<td>Info</td>
<td>1</td>
<td>0</td>
<td>Device</td>
<td>Maintenance</td>
<td>UFM</td>
<td></td>
</tr>
<tr>
<td>508</td>
<td>Core Dump Created</td>
<td>1</td>
<td>Info</td>
<td>1</td>
<td>0</td>
<td>Grid</td>
<td>Maintenance</td>
<td>UFM</td>
<td></td>
</tr>
<tr>
<td>510</td>
<td>SM Failover</td>
<td>0</td>
<td>Critical</td>
<td>1</td>
<td>0</td>
<td>Grid</td>
<td>Fabric Notification</td>
<td>SM</td>
<td></td>
</tr>
<tr>
<td>Alarm ID</td>
<td>Alarm Name</td>
<td>To Log</td>
<td>Alarm Name</td>
<td>Default Severity</td>
<td>Default Threshold</td>
<td>Related Object</td>
<td>Category</td>
<td>Source</td>
<td></td>
</tr>
<tr>
<td>----------</td>
<td>--------------------------------</td>
<td>--------</td>
<td>------------</td>
<td>------------------</td>
<td>-------------------</td>
<td>----------------</td>
<td>-------------------</td>
<td>--------</td>
<td></td>
</tr>
<tr>
<td>511</td>
<td>SM State Change</td>
<td>0</td>
<td>1</td>
<td>Info</td>
<td>1</td>
<td>0</td>
<td>Grid</td>
<td>SM</td>
<td></td>
</tr>
<tr>
<td>512</td>
<td>SM UP</td>
<td>0</td>
<td>1</td>
<td>Info</td>
<td>1</td>
<td>0</td>
<td>Grid</td>
<td>SM</td>
<td></td>
</tr>
<tr>
<td>513</td>
<td>SM System Log Message</td>
<td>0</td>
<td>1</td>
<td>Minor</td>
<td>1</td>
<td>0</td>
<td>Grid</td>
<td>SM</td>
<td></td>
</tr>
<tr>
<td>514</td>
<td>SM LiD Change</td>
<td>0</td>
<td>1</td>
<td>Warning</td>
<td>1</td>
<td>0</td>
<td>Grid</td>
<td>SM</td>
<td></td>
</tr>
<tr>
<td>515</td>
<td>Fabric Health Report</td>
<td>1</td>
<td>1</td>
<td>Info</td>
<td>1</td>
<td>0</td>
<td>Grid</td>
<td>UFM</td>
<td></td>
</tr>
<tr>
<td>516</td>
<td>Fabric Health Report Warning</td>
<td>1</td>
<td>1</td>
<td>Warning</td>
<td>1</td>
<td>0</td>
<td>Grid</td>
<td>UFM</td>
<td></td>
</tr>
<tr>
<td>517</td>
<td>Fabric Health Report Error</td>
<td>1</td>
<td>1</td>
<td>Critical</td>
<td>1</td>
<td>0</td>
<td>Grid</td>
<td>UFM</td>
<td></td>
</tr>
<tr>
<td>518</td>
<td>UFM-related process is down</td>
<td>1</td>
<td>1</td>
<td>Critical</td>
<td>1</td>
<td>0</td>
<td>Grid</td>
<td>UFM</td>
<td></td>
</tr>
<tr>
<td>519</td>
<td>Logs purge failure</td>
<td>1</td>
<td>1</td>
<td>Minor</td>
<td>1</td>
<td>0</td>
<td>Grid</td>
<td>UFM</td>
<td></td>
</tr>
<tr>
<td>520</td>
<td>Restart of UFM-related process succeeded</td>
<td>1</td>
<td>1</td>
<td>Info</td>
<td>1</td>
<td>0</td>
<td>Grid</td>
<td>Maintenance</td>
<td>UFM</td>
</tr>
<tr>
<td>521</td>
<td>UFM is being stopped</td>
<td>1</td>
<td>1</td>
<td>Critical</td>
<td>1</td>
<td>0</td>
<td>Grid</td>
<td>UFM</td>
<td></td>
</tr>
<tr>
<td>522</td>
<td>UFM is being restarted</td>
<td>1</td>
<td>1</td>
<td>Critical</td>
<td>1</td>
<td>0</td>
<td>Grid</td>
<td>UFM</td>
<td></td>
</tr>
<tr>
<td>523</td>
<td>UFM failover is being attempted</td>
<td>1</td>
<td>1</td>
<td>Info</td>
<td>1</td>
<td>0</td>
<td>Grid</td>
<td>UFM</td>
<td></td>
</tr>
<tr>
<td>524</td>
<td>UFM cannot connect to DB</td>
<td>1</td>
<td>1</td>
<td>Critical</td>
<td>1</td>
<td>0</td>
<td>Grid</td>
<td>UFM</td>
<td></td>
</tr>
<tr>
<td>525</td>
<td>Disk utilization threshold reached</td>
<td>1</td>
<td>1</td>
<td>Critical</td>
<td>1</td>
<td>0</td>
<td>Grid</td>
<td>Maintenance</td>
<td>UFM</td>
</tr>
<tr>
<td>526</td>
<td>Memory utilization threshold reached</td>
<td>1</td>
<td>1</td>
<td>Critical</td>
<td>1</td>
<td>0</td>
<td>Grid</td>
<td>Maintenance</td>
<td>UFM</td>
</tr>
<tr>
<td>527</td>
<td>CPU utilization threshold reached</td>
<td>1</td>
<td>1</td>
<td>Critical</td>
<td>1</td>
<td>0</td>
<td>Grid</td>
<td>Maintenance</td>
<td>UFM</td>
</tr>
<tr>
<td>528</td>
<td>Fabric interface is down</td>
<td>1</td>
<td>1</td>
<td>Critical</td>
<td>1</td>
<td>0</td>
<td>Grid</td>
<td>UFM</td>
<td></td>
</tr>
<tr>
<td>529</td>
<td>UFM standby server problem</td>
<td>1</td>
<td>1</td>
<td>Critical</td>
<td>1</td>
<td>0</td>
<td>Grid</td>
<td>UFM</td>
<td></td>
</tr>
<tr>
<td>Alarm ID</td>
<td>Alarm Name</td>
<td>To Log</td>
<td>Alarm Severity</td>
<td>Default Severity</td>
<td>Default Threshold</td>
<td>Default TTL</td>
<td>Related Object</td>
<td>Category</td>
<td>Source</td>
</tr>
<tr>
<td>----------</td>
<td>-------------------------------------------------</td>
<td>--------</td>
<td>----------------</td>
<td>------------------</td>
<td>-------------------</td>
<td>-------------</td>
<td>----------------</td>
<td>----------------</td>
<td>----------</td>
</tr>
<tr>
<td>530</td>
<td>SM is down</td>
<td>1</td>
<td>1</td>
<td>Critical</td>
<td>1</td>
<td>0</td>
<td>Grid</td>
<td>Maintenance</td>
<td>UFM</td>
</tr>
<tr>
<td>531</td>
<td>DRBD Bad Condition</td>
<td>1</td>
<td>1</td>
<td>Critical</td>
<td>1</td>
<td>0</td>
<td>Grid</td>
<td>Maintenance</td>
<td>UFM</td>
</tr>
<tr>
<td>532</td>
<td>Remote UFM-SM Sync</td>
<td>1</td>
<td>1</td>
<td>Info</td>
<td>1</td>
<td>0</td>
<td>Grid</td>
<td>Maintenance</td>
<td>UFM</td>
</tr>
<tr>
<td>533</td>
<td>Remote UFM-SM problem</td>
<td>1</td>
<td>1</td>
<td>Critical</td>
<td>1</td>
<td>0</td>
<td>Site</td>
<td>Maintenance</td>
<td>UFM</td>
</tr>
<tr>
<td>536</td>
<td>UFM Health Watchdog Info</td>
<td>1</td>
<td>1</td>
<td>Info</td>
<td>1</td>
<td>0</td>
<td>Grid</td>
<td>Maintenance</td>
<td>UFM</td>
</tr>
<tr>
<td>537</td>
<td>UFM Health Watchdog Critical</td>
<td>1</td>
<td>1</td>
<td>Critical</td>
<td>1</td>
<td>0</td>
<td>Grid</td>
<td>Maintenance</td>
<td>UFM</td>
</tr>
<tr>
<td>538</td>
<td>Time Diff Between HA Servers</td>
<td>1</td>
<td>1</td>
<td>Warning</td>
<td>1</td>
<td>0</td>
<td>Grid</td>
<td>Maintenance</td>
<td>UFM</td>
</tr>
<tr>
<td>539</td>
<td>DRBD TCP Connection Performance</td>
<td>1</td>
<td>1</td>
<td>Warning</td>
<td>1</td>
<td>0</td>
<td>Grid</td>
<td>Maintenance</td>
<td>UFM</td>
</tr>
<tr>
<td>540</td>
<td>Daily Report Completed successfully</td>
<td>1</td>
<td>0</td>
<td>Info</td>
<td>1</td>
<td>0</td>
<td>Grid</td>
<td>Maintenance</td>
<td>UFM</td>
</tr>
<tr>
<td>541</td>
<td>Daily Report Completed with Error</td>
<td>1</td>
<td>0</td>
<td>Minor</td>
<td>1</td>
<td>0</td>
<td>Grid</td>
<td>Maintenance</td>
<td>UFM</td>
</tr>
<tr>
<td>542</td>
<td>Daily Report Failed</td>
<td>1</td>
<td>0</td>
<td>Critical</td>
<td>1</td>
<td>0</td>
<td>Grid</td>
<td>Maintenance</td>
<td>UFM</td>
</tr>
<tr>
<td>543</td>
<td>Daily Report Mail Sent successfully</td>
<td>1</td>
<td>0</td>
<td>Info</td>
<td>1</td>
<td>0</td>
<td>Grid</td>
<td>Maintenance</td>
<td>UFM</td>
</tr>
<tr>
<td>544</td>
<td>Daily Report Mail Sent Failed</td>
<td>1</td>
<td>0</td>
<td>Minor</td>
<td>1</td>
<td>0</td>
<td>Grid</td>
<td>Maintenance</td>
<td>UFM</td>
</tr>
<tr>
<td>545</td>
<td>SM is not responding</td>
<td>1</td>
<td>1</td>
<td>Critical</td>
<td>1</td>
<td>0</td>
<td>Grid</td>
<td>Maintenance</td>
<td>UFM</td>
</tr>
<tr>
<td>546</td>
<td>Management interface is down</td>
<td>1</td>
<td>1</td>
<td>Critical</td>
<td>1</td>
<td>0</td>
<td>Grid</td>
<td>Maintenance</td>
<td>UFM</td>
</tr>
<tr>
<td>547</td>
<td>UFM stopped polling SM for updates</td>
<td>1</td>
<td>1</td>
<td>Critical</td>
<td>1</td>
<td>0</td>
<td>Grid</td>
<td>Maintenance</td>
<td>UFM</td>
</tr>
<tr>
<td>548</td>
<td>Failed to run generic script test</td>
<td>1</td>
<td>1</td>
<td>Critical</td>
<td>1</td>
<td>0</td>
<td>Grid</td>
<td>Maintenance</td>
<td>UFM</td>
</tr>
<tr>
<td>551</td>
<td>General External Event Notification</td>
<td>1</td>
<td>0</td>
<td>Info</td>
<td>0</td>
<td>0</td>
<td>Grid</td>
<td>Maintenance</td>
<td>UFM</td>
</tr>
<tr>
<td>552</td>
<td>General External Event Notice</td>
<td>1</td>
<td>0</td>
<td>Minor</td>
<td>0</td>
<td>0</td>
<td>Grid</td>
<td>Maintenance</td>
<td>UFM</td>
</tr>
<tr>
<td>553</td>
<td>General External Event Alert</td>
<td>1</td>
<td>0</td>
<td>Warning</td>
<td>0</td>
<td>0</td>
<td>Grid</td>
<td>Maintenance</td>
<td>UFM</td>
</tr>
<tr>
<td>554</td>
<td>General External Event Error</td>
<td>1</td>
<td>0</td>
<td>Critical</td>
<td>0</td>
<td>0</td>
<td>Grid</td>
<td>Maintenance</td>
<td>UFM</td>
</tr>
<tr>
<td>560</td>
<td>User Connected</td>
<td>1</td>
<td>0</td>
<td>Info</td>
<td>1</td>
<td>0</td>
<td>Grid</td>
<td>Security</td>
<td>UFM</td>
</tr>
<tr>
<td>Alarm ID</td>
<td>Alarm Name</td>
<td>To Log</td>
<td>Alarm</td>
<td>Default Severity</td>
<td>Default Threshold</td>
<td>Default TTL</td>
<td>Related Object</td>
<td>Category</td>
<td>Source</td>
</tr>
<tr>
<td>----------</td>
<td>------------------------------------</td>
<td>--------</td>
<td>-------</td>
<td>------------------</td>
<td>-------------------</td>
<td>-------------</td>
<td>----------------</td>
<td>----------------</td>
<td>--------</td>
</tr>
<tr>
<td>561</td>
<td>User Disconnected</td>
<td>1</td>
<td>0</td>
<td>Info</td>
<td>1</td>
<td>0</td>
<td>Grid</td>
<td>Security</td>
<td>UFM</td>
</tr>
<tr>
<td>602</td>
<td>UFM Server Failover</td>
<td>1</td>
<td>1</td>
<td>Critical</td>
<td>1</td>
<td>0</td>
<td>Site</td>
<td>Fabric Notification</td>
<td>UFM</td>
</tr>
<tr>
<td>603</td>
<td>Events Suppression</td>
<td>1</td>
<td>0</td>
<td>Critical</td>
<td>0</td>
<td>0</td>
<td>Site</td>
<td>Maintenance</td>
<td>UFM</td>
</tr>
<tr>
<td>604</td>
<td>Report Succeeded</td>
<td>1</td>
<td>1</td>
<td>Info</td>
<td>1</td>
<td>0</td>
<td>Grid</td>
<td>Maintenance</td>
<td>UFM</td>
</tr>
<tr>
<td>605</td>
<td>Report Failed</td>
<td>1</td>
<td>1</td>
<td>Critical</td>
<td>1</td>
<td>0</td>
<td>Grid</td>
<td>Maintenance</td>
<td>UFM</td>
</tr>
<tr>
<td>606</td>
<td>Correction Attempts Paused</td>
<td>1</td>
<td>0</td>
<td>Warning</td>
<td>1</td>
<td>0</td>
<td>Site</td>
<td>Fabric Notification</td>
<td>UFM</td>
</tr>
<tr>
<td>610</td>
<td>Monitoring History Enabled</td>
<td>1</td>
<td>0</td>
<td>Info</td>
<td>1</td>
<td>0</td>
<td>Grid</td>
<td>Maintenance</td>
<td>UFM</td>
</tr>
<tr>
<td>611</td>
<td>Monitoring History Disabled</td>
<td>1</td>
<td>0</td>
<td>Info</td>
<td>1</td>
<td>0</td>
<td>Grid</td>
<td>Maintenance</td>
<td>UFM</td>
</tr>
<tr>
<td>612</td>
<td>Monitoring History Bad Connection</td>
<td>1</td>
<td>1</td>
<td>Critical</td>
<td>1</td>
<td>0</td>
<td>Grid</td>
<td>Maintenance</td>
<td>UFM</td>
</tr>
<tr>
<td>613</td>
<td>Monitoring History Connection Established</td>
<td>1</td>
<td>0</td>
<td>Info</td>
<td>1</td>
<td>0</td>
<td>Grid</td>
<td>Maintenance</td>
<td>UFM</td>
</tr>
<tr>
<td>614</td>
<td>Monitoring History Inconsistent Version</td>
<td>1</td>
<td>1</td>
<td>Critical</td>
<td>1</td>
<td>0</td>
<td>Grid</td>
<td>Maintenance</td>
<td>UFM</td>
</tr>
<tr>
<td>615</td>
<td>Monitoring History Failed save metadata</td>
<td>1</td>
<td>1</td>
<td>Critical</td>
<td>1</td>
<td>0</td>
<td>Grid</td>
<td>Maintenance</td>
<td>UFM</td>
</tr>
<tr>
<td>616</td>
<td>Monitoring History Failed update metadata</td>
<td>1</td>
<td>1</td>
<td>Critical</td>
<td>1</td>
<td>0</td>
<td>Grid</td>
<td>Maintenance</td>
<td>UFM</td>
</tr>
<tr>
<td>617</td>
<td>Monitoring History Failed save data</td>
<td>1</td>
<td>1</td>
<td>Critical</td>
<td>1</td>
<td>0</td>
<td>Grid</td>
<td>Maintenance</td>
<td>UFM</td>
</tr>
<tr>
<td>618</td>
<td>Monitoring History Failed get metadata</td>
<td>1</td>
<td>1</td>
<td>Critical</td>
<td>1</td>
<td>0</td>
<td>Grid</td>
<td>Maintenance</td>
<td>UFM</td>
</tr>
<tr>
<td>619</td>
<td>Monitoring History Failed get data</td>
<td>1</td>
<td>1</td>
<td>Critical</td>
<td>1</td>
<td>0</td>
<td>Grid</td>
<td>Maintenance</td>
<td>UFM</td>
</tr>
<tr>
<td>620</td>
<td>Monitoring History Failed remove file</td>
<td>1</td>
<td>1</td>
<td>Critical</td>
<td>1</td>
<td>0</td>
<td>Grid</td>
<td>Maintenance</td>
<td>UFM</td>
</tr>
<tr>
<td>621</td>
<td>Monitoring History version not matches UFM version</td>
<td>1</td>
<td>1</td>
<td>Critical</td>
<td>1</td>
<td>0</td>
<td>Grid</td>
<td>Maintenance</td>
<td>UFM</td>
</tr>
<tr>
<td>622</td>
<td>Monitoring History Purge DB Occurred</td>
<td>1</td>
<td>1</td>
<td>Warning</td>
<td>1</td>
<td>0</td>
<td>Grid</td>
<td>Maintenance</td>
<td>UFM</td>
</tr>
<tr>
<td>623</td>
<td>Monitoring History Migration is not completed</td>
<td>1</td>
<td>1</td>
<td>Warning</td>
<td>1</td>
<td>0</td>
<td>Grid</td>
<td>Maintenance</td>
<td>UFM</td>
</tr>
<tr>
<td>Alarm ID</td>
<td>Alarm Name</td>
<td>To Log</td>
<td>Alarm</td>
<td>Default Severity</td>
<td>Default Threshold</td>
<td>Default TTL</td>
<td>Related Object</td>
<td>Category</td>
<td>Source</td>
</tr>
<tr>
<td>----------</td>
<td>---------------------------------------------------------------------------</td>
<td>--------</td>
<td>-------</td>
<td>------------------</td>
<td>-------------------</td>
<td>-------------</td>
<td>----------------</td>
<td>--------------</td>
<td>--------</td>
</tr>
<tr>
<td>624</td>
<td>Monitoring History partition utilization threshold reached</td>
<td>1</td>
<td>1</td>
<td>Critical</td>
<td>1</td>
<td>0</td>
<td>Grid</td>
<td>Maintenance</td>
<td>UFM</td>
</tr>
<tr>
<td>625</td>
<td>Monitoring History local report files are about to be cleaned</td>
<td>1</td>
<td>0</td>
<td>Info</td>
<td>1</td>
<td>0</td>
<td>Grid</td>
<td>Maintenance</td>
<td>UFM</td>
</tr>
<tr>
<td>626</td>
<td>Monitoring History oldest DB table is about to be removed</td>
<td>1</td>
<td>0</td>
<td>Info</td>
<td>1</td>
<td>0</td>
<td>Grid</td>
<td>Maintenance</td>
<td>UFM</td>
</tr>
<tr>
<td>627</td>
<td>Monitoring History partition is not mounted</td>
<td>1</td>
<td>0</td>
<td>Critical</td>
<td>1</td>
<td>0</td>
<td>Grid</td>
<td>Maintenance</td>
<td>UFM</td>
</tr>
<tr>
<td>701</td>
<td>Non-optimal Link Speed</td>
<td>1</td>
<td>1</td>
<td>Minor</td>
<td>1</td>
<td>0</td>
<td>Port</td>
<td>Hardware</td>
<td>UFM</td>
</tr>
<tr>
<td>702</td>
<td>Unhealthy IB Port</td>
<td>1</td>
<td>1</td>
<td>Warning</td>
<td>1</td>
<td>0</td>
<td>Port</td>
<td>Hardware</td>
<td>SM</td>
</tr>
<tr>
<td>703</td>
<td>Fabric Collector Connected</td>
<td>1</td>
<td>0</td>
<td>Info</td>
<td>1</td>
<td>0</td>
<td>Grid</td>
<td>Maintenance</td>
<td>UFM</td>
</tr>
<tr>
<td>704</td>
<td>Fabric Collector Disconnected</td>
<td>1</td>
<td>1</td>
<td>Critical</td>
<td>1</td>
<td>0</td>
<td>Grid</td>
<td>Maintenance</td>
<td>UFM</td>
</tr>
<tr>
<td>750</td>
<td>High data retransmission count on port</td>
<td>1</td>
<td>1</td>
<td>Warning</td>
<td>500</td>
<td>0</td>
<td>Port</td>
<td>Hardware</td>
<td>SM</td>
</tr>
<tr>
<td>901</td>
<td>Fabric Configuration Started</td>
<td>0</td>
<td>1</td>
<td>Info</td>
<td>1</td>
<td>0</td>
<td>Grid</td>
<td>Fabric Notification</td>
<td>UFM</td>
</tr>
<tr>
<td>902</td>
<td>Fabric Configuration Completed</td>
<td>0</td>
<td>1</td>
<td>Info</td>
<td>1</td>
<td>0</td>
<td>Grid</td>
<td>Fabric Notification</td>
<td>UFM</td>
</tr>
<tr>
<td>903</td>
<td>Fabric Configuration Failed</td>
<td>0</td>
<td>1</td>
<td>Critical</td>
<td>1</td>
<td>0</td>
<td>Grid</td>
<td>Fabric Notification</td>
<td>UFM</td>
</tr>
<tr>
<td>904</td>
<td>Device Configuration Failure</td>
<td>0</td>
<td>1</td>
<td>Critical</td>
<td>1</td>
<td>0</td>
<td>Device</td>
<td>Fabric Notification</td>
<td>UFM</td>
</tr>
<tr>
<td>905</td>
<td>Device Configuration Timeout</td>
<td>0</td>
<td>1</td>
<td>Critical</td>
<td>1</td>
<td>0</td>
<td>Device</td>
<td>Fabric Notification</td>
<td>UFM</td>
</tr>
<tr>
<td>906</td>
<td>Provisioning Validation Failure</td>
<td>0</td>
<td>1</td>
<td>Critical</td>
<td>1</td>
<td>0</td>
<td>Grid</td>
<td>Fabric Notification</td>
<td>UFM</td>
</tr>
<tr>
<td>907</td>
<td>Switch is Down</td>
<td>1</td>
<td>1</td>
<td>Critical</td>
<td>1</td>
<td>0</td>
<td>Site</td>
<td>Fabric Topology</td>
<td>UFM</td>
</tr>
<tr>
<td>908</td>
<td>Switch is Up</td>
<td>1</td>
<td>1</td>
<td>Info</td>
<td>1</td>
<td>0</td>
<td>Site</td>
<td>Fabric Topology</td>
<td>UFM</td>
</tr>
<tr>
<td>909</td>
<td>Director Switch is Down</td>
<td>1</td>
<td>1</td>
<td>Critical</td>
<td>1</td>
<td>0</td>
<td>Site</td>
<td>Fabric Topology</td>
<td>UFM</td>
</tr>
<tr>
<td>910</td>
<td>Director Switch is Up</td>
<td>1</td>
<td>1</td>
<td>Info</td>
<td>1</td>
<td>0</td>
<td>Site</td>
<td>Fabric Topology</td>
<td>UFM</td>
</tr>
<tr>
<td>Alarm ID</td>
<td>Alarm Name</td>
<td>To Log</td>
<td>Alarm</td>
<td>Default Severity</td>
<td>Default Threshold</td>
<td>Default TTL</td>
<td>Related Object</td>
<td>Category</td>
<td>Source</td>
</tr>
<tr>
<td>----------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------</td>
<td>-------</td>
<td>------------------</td>
<td>------------------</td>
<td>-------------</td>
<td>----------------</td>
<td>----------</td>
<td>--------</td>
</tr>
<tr>
<td>911</td>
<td>Module Temperature Low Threshold Reached (Unmanaged IB switches only)</td>
<td>1</td>
<td>1</td>
<td>Warning</td>
<td>0</td>
<td>0</td>
<td>Module</td>
<td>Hardware</td>
<td>Telemetry</td>
</tr>
<tr>
<td>912</td>
<td>Module Temperature High Threshold Reached (Unmanaged IB switches only)</td>
<td>1</td>
<td>1</td>
<td>Critical</td>
<td>60</td>
<td>0</td>
<td>Module</td>
<td>Hardware</td>
<td>Telemetry</td>
</tr>
<tr>
<td>913</td>
<td>Module High Voltage</td>
<td>1</td>
<td>1</td>
<td>Warning</td>
<td>15</td>
<td>0</td>
<td>Switch</td>
<td>Module Status</td>
<td>Telemetry</td>
</tr>
<tr>
<td>914</td>
<td>Module High Current</td>
<td>1</td>
<td>1</td>
<td>Warning</td>
<td>10</td>
<td>0</td>
<td>Switch</td>
<td>Module Status</td>
<td>Telemetry</td>
</tr>
<tr>
<td>915</td>
<td>Critical BER reported</td>
<td>1</td>
<td>1</td>
<td>Critical</td>
<td>10</td>
<td>0</td>
<td>Port</td>
<td>Hardware</td>
<td>Telemetry</td>
</tr>
<tr>
<td>916</td>
<td>High BER reported</td>
<td>1</td>
<td>1</td>
<td>Warning</td>
<td>10</td>
<td>0</td>
<td>Port</td>
<td>Hardware</td>
<td>Telemetry</td>
</tr>
<tr>
<td>917</td>
<td>Critical Symbol BER reported</td>
<td>1</td>
<td>1</td>
<td>Critical</td>
<td>10</td>
<td>0</td>
<td>Port</td>
<td>Hardware</td>
<td>Telemetry</td>
</tr>
<tr>
<td>918</td>
<td>High Symbol BER reported</td>
<td>1</td>
<td>1</td>
<td>Warning</td>
<td>10</td>
<td>0</td>
<td>Port</td>
<td>Hardware</td>
<td>Telemetry</td>
</tr>
<tr>
<td>919</td>
<td>Cable Temperature High</td>
<td>1</td>
<td>1</td>
<td>Critical</td>
<td>1</td>
<td>0</td>
<td>Port</td>
<td>Hardware</td>
<td>UFM</td>
</tr>
<tr>
<td>920</td>
<td>Cable Temperature Low</td>
<td>1</td>
<td>1</td>
<td>Critical</td>
<td>1</td>
<td>0</td>
<td>Port</td>
<td>Hardware</td>
<td>UFM</td>
</tr>
<tr>
<td>1300</td>
<td>SA Key violation</td>
<td>1</td>
<td>1</td>
<td>Warning</td>
<td>5</td>
<td>0</td>
<td>Port</td>
<td>Security</td>
<td>SM</td>
</tr>
<tr>
<td>1301</td>
<td>SGID Spoofed</td>
<td>1</td>
<td>1</td>
<td>Warning</td>
<td>5</td>
<td>0</td>
<td>Port</td>
<td>Security</td>
<td>SM</td>
</tr>
<tr>
<td>1302</td>
<td>SA High Rate detected</td>
<td>1</td>
<td>1</td>
<td>Warning</td>
<td>5</td>
<td>0</td>
<td>Port</td>
<td>Security</td>
<td>SM</td>
</tr>
<tr>
<td>1303</td>
<td>Multicast subscriptions over limit</td>
<td>1</td>
<td>1</td>
<td>Warning</td>
<td>5</td>
<td>0</td>
<td>Port</td>
<td>Security</td>
<td>SM</td>
</tr>
<tr>
<td>1304</td>
<td>Service Record subscriptions over limit</td>
<td>1</td>
<td>1</td>
<td>Warning</td>
<td>5</td>
<td>0</td>
<td>Port</td>
<td>Security</td>
<td>SM</td>
</tr>
<tr>
<td>1305</td>
<td>Event subscriptions over limit</td>
<td>1</td>
<td>1</td>
<td>Warning</td>
<td>5</td>
<td>0</td>
<td>Port</td>
<td>Security</td>
<td>SM</td>
</tr>
<tr>
<td>1306</td>
<td>Unallowed SM was detected in the fabric</td>
<td>1</td>
<td>1</td>
<td>Warning</td>
<td>0</td>
<td>0</td>
<td>Port</td>
<td>Fabric Notification</td>
<td>SM</td>
</tr>
<tr>
<td>1307</td>
<td>SMIInfo SET request was received from unallowed SM</td>
<td>1</td>
<td>1</td>
<td>Warning</td>
<td>0</td>
<td>0</td>
<td>Port</td>
<td>Fabric Notification</td>
<td>SM</td>
</tr>
<tr>
<td>1309</td>
<td>SM was detected with non-matching SMKey</td>
<td>1</td>
<td>1</td>
<td>Warning</td>
<td>0</td>
<td>0</td>
<td>Port</td>
<td>Fabric Notification</td>
<td>SM</td>
</tr>
<tr>
<td>1310</td>
<td>Duplicated node GUID was detected</td>
<td>1</td>
<td>1</td>
<td>Critical</td>
<td>1</td>
<td>0</td>
<td>Device</td>
<td>Fabric Notification</td>
<td>SM</td>
</tr>
<tr>
<td>Alarm ID</td>
<td>Alarm Name</td>
<td>To Log</td>
<td>Alarm Name</td>
<td>Default Severity</td>
<td>Default Threshold</td>
<td>Default TTL</td>
<td>Related Object</td>
<td>Category</td>
<td>Source</td>
</tr>
<tr>
<td>----------</td>
<td>-----------------------------------------------</td>
<td>--------</td>
<td>------------</td>
<td>------------------</td>
<td>-------------------</td>
<td>-------------</td>
<td>----------------</td>
<td>------------</td>
<td>--------</td>
</tr>
<tr>
<td>1311</td>
<td>Duplicated port GUID was detected</td>
<td>1</td>
<td>Critical</td>
<td>1</td>
<td>0</td>
<td>Port</td>
<td>Fabric Notification</td>
<td>SM</td>
<td></td>
</tr>
<tr>
<td>1312</td>
<td>Switch was Rebooted</td>
<td>1</td>
<td>Info</td>
<td>1</td>
<td>0</td>
<td>Device</td>
<td>Fabric Notification</td>
<td>UFM</td>
<td></td>
</tr>
<tr>
<td>1315</td>
<td>Topo Config File Error</td>
<td>1</td>
<td>Critical</td>
<td>1</td>
<td>0</td>
<td>Grid</td>
<td>Fabric Notification</td>
<td>UFM</td>
<td></td>
</tr>
<tr>
<td>1316</td>
<td>Topo Config Subnet Mismatch</td>
<td>1</td>
<td>Critical</td>
<td>1</td>
<td>0</td>
<td>Grid</td>
<td>Fabric Notification</td>
<td>Topodiff</td>
<td></td>
</tr>
<tr>
<td>1400</td>
<td>High Ambient Temperature</td>
<td>1</td>
<td>Warning</td>
<td>0</td>
<td>0</td>
<td>Switch</td>
<td>Hardware</td>
<td>Switch</td>
<td></td>
</tr>
<tr>
<td>1401</td>
<td>High Fluid Temperature</td>
<td>1</td>
<td>Warning</td>
<td>0</td>
<td>0</td>
<td>Switch</td>
<td>Hardware</td>
<td>Switch</td>
<td></td>
</tr>
<tr>
<td>1402</td>
<td>Low Fluid Level</td>
<td>1</td>
<td>Warning</td>
<td>0</td>
<td>0</td>
<td>Switch</td>
<td>Hardware</td>
<td>Switch</td>
<td></td>
</tr>
<tr>
<td>1403</td>
<td>Low Supply Pressure</td>
<td>1</td>
<td>Warning</td>
<td>0</td>
<td>0</td>
<td>Switch</td>
<td>Hardware</td>
<td>Switch</td>
<td></td>
</tr>
<tr>
<td>1404</td>
<td>High Supply Pressure</td>
<td>1</td>
<td>Warning</td>
<td>0</td>
<td>0</td>
<td>Switch</td>
<td>Hardware</td>
<td>Switch</td>
<td></td>
</tr>
<tr>
<td>1405</td>
<td>Low Return Pressure</td>
<td>1</td>
<td>Warning</td>
<td>0</td>
<td>0</td>
<td>Switch</td>
<td>Hardware</td>
<td>Switch</td>
<td></td>
</tr>
<tr>
<td>1406</td>
<td>High Return Pressure</td>
<td>1</td>
<td>Warning</td>
<td>0</td>
<td>0</td>
<td>Switch</td>
<td>Hardware</td>
<td>Switch</td>
<td></td>
</tr>
<tr>
<td>1407</td>
<td>High Differential Pressure</td>
<td>1</td>
<td>Warning</td>
<td>0</td>
<td>0</td>
<td>Switch</td>
<td>Hardware</td>
<td>Switch</td>
<td></td>
</tr>
<tr>
<td>1408</td>
<td>Low Differential Pressure</td>
<td>1</td>
<td>Warning</td>
<td>0</td>
<td>0</td>
<td>Switch</td>
<td>Hardware</td>
<td>Switch</td>
<td></td>
</tr>
<tr>
<td>1409</td>
<td>System Fail Safe</td>
<td>1</td>
<td>Warning</td>
<td>0</td>
<td>0</td>
<td>Switch</td>
<td>Hardware</td>
<td>Switch</td>
<td></td>
</tr>
<tr>
<td>1410</td>
<td>Fault Critical</td>
<td>1</td>
<td>Critical</td>
<td>0</td>
<td>0</td>
<td>Switch</td>
<td>Hardware</td>
<td>Switch</td>
<td></td>
</tr>
<tr>
<td>1411</td>
<td>Fault Pump1</td>
<td>1</td>
<td>Critical</td>
<td>0</td>
<td>0</td>
<td>Switch</td>
<td>Hardware</td>
<td>Switch</td>
<td></td>
</tr>
<tr>
<td>1412</td>
<td>Fault Pump2</td>
<td>1</td>
<td>Critical</td>
<td>0</td>
<td>0</td>
<td>Switch</td>
<td>Hardware</td>
<td>Switch</td>
<td></td>
</tr>
<tr>
<td>1413</td>
<td>Fault Fluid Level Critical</td>
<td>1</td>
<td>Critical</td>
<td>0</td>
<td>0</td>
<td>Switch</td>
<td>Hardware</td>
<td>Switch</td>
<td></td>
</tr>
<tr>
<td>1414</td>
<td>Fault Fluid Over Temperature</td>
<td>1</td>
<td>Critical</td>
<td>0</td>
<td>0</td>
<td>Switch</td>
<td>Hardware</td>
<td>Switch</td>
<td></td>
</tr>
<tr>
<td>1415</td>
<td>Fault Primary DC</td>
<td>1</td>
<td>Critical</td>
<td>0</td>
<td>0</td>
<td>Switch</td>
<td>Hardware</td>
<td>Switch</td>
<td></td>
</tr>
<tr>
<td>1416</td>
<td>Fault Redundant DC</td>
<td>1</td>
<td>Critical</td>
<td>0</td>
<td>0</td>
<td>Switch</td>
<td>Hardware</td>
<td>Switch</td>
<td></td>
</tr>
<tr>
<td>1417</td>
<td>Fault Fluid Leak</td>
<td>1</td>
<td>Critical</td>
<td>0</td>
<td>0</td>
<td>Switch</td>
<td>Hardware</td>
<td>Switch</td>
<td></td>
</tr>
<tr>
<td>1418</td>
<td>Fault Sensor Failure</td>
<td>1</td>
<td>Critical</td>
<td>0</td>
<td>0</td>
<td>Switch</td>
<td>Hardware</td>
<td>Switch</td>
<td></td>
</tr>
<tr>
<td>1419</td>
<td>Cooling Device Monitoring Error</td>
<td>1</td>
<td>Critical</td>
<td>0</td>
<td>0</td>
<td>Grid</td>
<td>Hardware</td>
<td>Switch</td>
<td></td>
</tr>
<tr>
<td>1420</td>
<td>Cooling Device Communication Error</td>
<td>1</td>
<td>Critical</td>
<td>0</td>
<td>0</td>
<td>Switch</td>
<td>Hardware</td>
<td>Switch</td>
<td></td>
</tr>
<tr>
<td>1500</td>
<td>New cable detected</td>
<td>1</td>
<td>Info</td>
<td>1</td>
<td>0</td>
<td>Link</td>
<td>Security</td>
<td>UFM</td>
<td></td>
</tr>
<tr>
<td>Alarm ID</td>
<td>Alarm Name</td>
<td>To Log</td>
<td>Alarm</td>
<td>Default Severity</td>
<td>Default Threshold</td>
<td>Default TTL</td>
<td>Related Object</td>
<td>Category</td>
<td>Source</td>
</tr>
<tr>
<td>----------</td>
<td>------------------------------------------------</td>
<td>--------</td>
<td>-------</td>
<td>------------------</td>
<td>-------------------</td>
<td>-------------</td>
<td>----------------</td>
<td>----------------</td>
<td>--------</td>
</tr>
<tr>
<td>1502</td>
<td>Cable detected in a new location</td>
<td>1</td>
<td>0</td>
<td>Warning</td>
<td>1</td>
<td>0</td>
<td>Link</td>
<td>Security</td>
<td>UFM</td>
</tr>
<tr>
<td>1503</td>
<td>Duplicate Cable Detected</td>
<td>1</td>
<td>0</td>
<td>Critical</td>
<td>1</td>
<td>0</td>
<td>Link</td>
<td>Security</td>
<td>UFM</td>
</tr>
<tr>
<td>1504</td>
<td>SHARP Allocation Succeeded</td>
<td>1</td>
<td>1</td>
<td>Info</td>
<td>1</td>
<td>0</td>
<td>Grid</td>
<td>SHARP</td>
<td>SHARP</td>
</tr>
<tr>
<td>1505</td>
<td>SHARP Allocation Failed</td>
<td>1</td>
<td>0</td>
<td>Warning</td>
<td>1</td>
<td>0</td>
<td>Grid</td>
<td>SHARP</td>
<td>SHARP</td>
</tr>
<tr>
<td>1506</td>
<td>SHARP Deallocation Succeeded</td>
<td>1</td>
<td>0</td>
<td>Info</td>
<td>1</td>
<td>0</td>
<td>Grid</td>
<td>SHARP</td>
<td>SHARP</td>
</tr>
<tr>
<td>1507</td>
<td>SHARP Deallocation Failed</td>
<td>1</td>
<td>0</td>
<td>Warning</td>
<td>1</td>
<td>0</td>
<td>Grid</td>
<td>SHARP</td>
<td>SHARP</td>
</tr>
<tr>
<td>1508</td>
<td>Device Collect System Dump Started</td>
<td>1</td>
<td>0</td>
<td>Info</td>
<td>1</td>
<td>0</td>
<td>Device</td>
<td>Maintenance</td>
<td>UFM</td>
</tr>
<tr>
<td>1509</td>
<td>Device Collect System Dump Finished</td>
<td>1</td>
<td>0</td>
<td>Info</td>
<td>1</td>
<td>0</td>
<td>Device</td>
<td>Maintenance</td>
<td>UFM</td>
</tr>
<tr>
<td>1510</td>
<td>Device Collect System Dump Error</td>
<td>1</td>
<td>0</td>
<td>Critical</td>
<td>1</td>
<td>0</td>
<td>Device</td>
<td>Maintenance</td>
<td>UFM</td>
</tr>
<tr>
<td>1511</td>
<td>Virtual Port Added</td>
<td>1</td>
<td>0</td>
<td>Info</td>
<td>1</td>
<td>0</td>
<td>Port</td>
<td>Fabric Notification</td>
<td>SM</td>
</tr>
<tr>
<td>1512</td>
<td>Virtual Port Removed</td>
<td>1</td>
<td>0</td>
<td>Warning</td>
<td>1</td>
<td>0</td>
<td>Port</td>
<td>Fabric Notification</td>
<td>SM</td>
</tr>
<tr>
<td>1513</td>
<td>Burn Cables Transceivers Started</td>
<td>1</td>
<td>0</td>
<td>Info</td>
<td>1</td>
<td>0</td>
<td>Device</td>
<td>Maintenance</td>
<td>UFM</td>
</tr>
<tr>
<td>1514</td>
<td>Burn Cables Transceivers Finished</td>
<td>1</td>
<td>0</td>
<td>Info</td>
<td>1</td>
<td>0</td>
<td>Device</td>
<td>Maintenance</td>
<td>UFM</td>
</tr>
<tr>
<td>1515</td>
<td>Burn Cables Transceivers Failed</td>
<td>1</td>
<td>0</td>
<td>Warning</td>
<td>1</td>
<td>0</td>
<td>Device</td>
<td>Maintenance</td>
<td>UFM</td>
</tr>
<tr>
<td>1516</td>
<td>Activate Cables Transceivers FW Finished</td>
<td>1</td>
<td>0</td>
<td>Info</td>
<td>1</td>
<td>0</td>
<td>Device</td>
<td>Maintenance</td>
<td>UFM</td>
</tr>
<tr>
<td>1517</td>
<td>Activate Cables Transceivers FW Failed</td>
<td>1</td>
<td>0</td>
<td>Warning</td>
<td>1</td>
<td>0</td>
<td>Device</td>
<td>Maintenance</td>
<td>UFM</td>
</tr>
<tr>
<td>1520</td>
<td>Aggregation Node Discovery Failed</td>
<td>1</td>
<td>0</td>
<td>Critical</td>
<td>1</td>
<td>0</td>
<td>SHARP AM</td>
<td>SHARP</td>
<td>SHARP</td>
</tr>
<tr>
<td>1521</td>
<td>Job Started</td>
<td>1</td>
<td>0</td>
<td>Info</td>
<td>1</td>
<td>0</td>
<td>SHARP AM</td>
<td>SHARP</td>
<td>SHARP</td>
</tr>
<tr>
<td>1522</td>
<td>Job Ended</td>
<td>1</td>
<td>0</td>
<td>Info</td>
<td>1</td>
<td>0</td>
<td>SHARP AM</td>
<td>SHARP</td>
<td>SHARP</td>
</tr>
<tr>
<td>1523</td>
<td>Job Start Failed</td>
<td>1</td>
<td>0</td>
<td>Critical</td>
<td>1</td>
<td>0</td>
<td>SHARP AM</td>
<td>SHARP</td>
<td>SHARP</td>
</tr>
<tr>
<td>Alarm ID</td>
<td>Alarm Name</td>
<td>To Log</td>
<td>Alarm Severity</td>
<td>Default Severity</td>
<td>Default Threshold</td>
<td>Related Object</td>
<td>Category</td>
<td>Source</td>
<td></td>
</tr>
<tr>
<td>----------</td>
<td>------------------------------------</td>
<td>--------</td>
<td>----------------</td>
<td>------------------</td>
<td>-------------------</td>
<td>----------------</td>
<td>----------</td>
<td>--------</td>
<td></td>
</tr>
<tr>
<td>1524</td>
<td>Job Error</td>
<td>1</td>
<td>0</td>
<td>Critical</td>
<td>1</td>
<td>0</td>
<td>SHARP</td>
<td>SHARP</td>
<td></td>
</tr>
<tr>
<td>1525</td>
<td>Trap QP Error</td>
<td>1</td>
<td>0</td>
<td>Critical</td>
<td>1</td>
<td>0</td>
<td>SHARP</td>
<td>SHARP</td>
<td></td>
</tr>
<tr>
<td>1526</td>
<td>Trap Invalid Request</td>
<td>1</td>
<td>0</td>
<td>Critical</td>
<td>1</td>
<td>0</td>
<td>SHARP</td>
<td>SHARP</td>
<td></td>
</tr>
<tr>
<td>1527</td>
<td>Trap Sharp Error</td>
<td>1</td>
<td>0</td>
<td>Critical</td>
<td>1</td>
<td>0</td>
<td>SHARP</td>
<td>SHARP</td>
<td></td>
</tr>
<tr>
<td>1528</td>
<td>Trap QP Alloc timeout</td>
<td>1</td>
<td>0</td>
<td>Critical</td>
<td>1</td>
<td>0</td>
<td>SHARP</td>
<td>SHARP</td>
<td></td>
</tr>
<tr>
<td>1529</td>
<td>Trap AMKey Violation</td>
<td>1</td>
<td>0</td>
<td>Critical</td>
<td>1</td>
<td>0</td>
<td>SHARP</td>
<td>SHARP</td>
<td></td>
</tr>
<tr>
<td>1530</td>
<td>Unsupported Trap</td>
<td>1</td>
<td>0</td>
<td>Critical</td>
<td>1</td>
<td>0</td>
<td>SHARP</td>
<td>SHARP</td>
<td></td>
</tr>
<tr>
<td>1531</td>
<td>Reservation Updated</td>
<td>1</td>
<td>0</td>
<td>Info</td>
<td>1</td>
<td>0</td>
<td>SHARP</td>
<td>SHARP</td>
<td></td>
</tr>
<tr>
<td>1532</td>
<td>Sharp is not Responding</td>
<td>1</td>
<td>0</td>
<td>Critical</td>
<td>1</td>
<td>0</td>
<td>SHARP</td>
<td>SHARP</td>
<td></td>
</tr>
<tr>
<td>1533</td>
<td>Agg Node Active</td>
<td>1</td>
<td>0</td>
<td>Info</td>
<td>1</td>
<td>0</td>
<td>SHARP</td>
<td>SHARP</td>
<td></td>
</tr>
<tr>
<td>1534</td>
<td>Agg Node Inactive</td>
<td>1</td>
<td>0</td>
<td>Warning</td>
<td>1</td>
<td>0</td>
<td>SHARP</td>
<td>SHARP</td>
<td></td>
</tr>
<tr>
<td>1535</td>
<td>Trap AMKey Violation Triggered by AM</td>
<td>1</td>
<td>0</td>
<td>Warning</td>
<td>1</td>
<td>0</td>
<td>SHARP</td>
<td>SHARP</td>
<td></td>
</tr>
<tr>
<td>1550</td>
<td>GUIDs Were Added to Pkey</td>
<td>1</td>
<td>0</td>
<td>Info</td>
<td>1</td>
<td>0</td>
<td>Port</td>
<td>UFM</td>
<td></td>
</tr>
<tr>
<td>1551</td>
<td>GUIDs Were Removed from Pkey</td>
<td>1</td>
<td>0</td>
<td>Info</td>
<td>1</td>
<td>0</td>
<td>Port</td>
<td>UFM</td>
<td></td>
</tr>
<tr>
<td>1600</td>
<td>VS/CC Classes Key Violation</td>
<td>1</td>
<td>0</td>
<td>Warning</td>
<td>1</td>
<td>0</td>
<td>Port</td>
<td>Security</td>
<td>SM</td>
</tr>
<tr>
<td>1601</td>
<td>Tenant Access Violation</td>
<td>1</td>
<td>0</td>
<td>Warning</td>
<td>1</td>
<td>0</td>
<td>Port</td>
<td>Security</td>
<td>SM</td>
</tr>
<tr>
<td>1602</td>
<td>PCI Speed Degradation Warning</td>
<td>1</td>
<td>1</td>
<td>Critical</td>
<td>1</td>
<td>0</td>
<td>Port</td>
<td>UFM</td>
<td></td>
</tr>
<tr>
<td>1603</td>
<td>PCI Width Degradation Warning</td>
<td>1</td>
<td>1</td>
<td>Critical</td>
<td>1</td>
<td>0</td>
<td>Port</td>
<td>UFM</td>
<td></td>
</tr>
<tr>
<td>1604</td>
<td>Non-optimal aggregated port bandwidth</td>
<td>1</td>
<td>1</td>
<td>Warning</td>
<td>1</td>
<td>0</td>
<td>Port</td>
<td>Hardware</td>
<td>UFM</td>
</tr>
</tbody>
</table>
13.4 Appendix - Used Ports

The following is the list of ports used by the UFM Server for internal and external communication:

<table>
<thead>
<tr>
<th>Port</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>80(tcp), 443(tcp)</td>
<td>Used by WS clients (Apache Web Server)</td>
</tr>
<tr>
<td>8000(udp)</td>
<td>Used for UFM server listening for REST API requests (redirected by Apache web server)</td>
</tr>
<tr>
<td>6306(udp)</td>
<td>Used for Multicast requests - communication with latest UFM Agents</td>
</tr>
<tr>
<td>8005(udp)</td>
<td>Used as UFM monitoring listening port</td>
</tr>
<tr>
<td>8089(tcp)</td>
<td>Used for internal communication between UFM server and MonitoringHistoryEngine</td>
</tr>
<tr>
<td>8888(tcp)</td>
<td>Used by DRBD - communication between UFM Primary and Standby server</td>
</tr>
<tr>
<td>15800(tcp)</td>
<td>Used for communication with legacy UFM Agents on Mellanox Grid Director DDR switches</td>
</tr>
<tr>
<td>8081(tcp), 8082(tcp)</td>
<td>Used for internal communication with Subnet Manager</td>
</tr>
</tbody>
</table>

13.5 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 (comma-separated) UFM, SM, SHARP, Telemetry. Or ALL for all the files.
track_conf_files = ALL
```

The below lists the configuration files that can be tracked:

<table>
<thead>
<tr>
<th>Conf File Alias</th>
<th>Configuration Files</th>
</tr>
</thead>
<tbody>
<tr>
<td>UFM</td>
<td>/opt/ufm/files/conf/gv.cfg</td>
</tr>
<tr>
<td>SM</td>
<td>/opt/ufm/files/conf/opensm/opensm.conf</td>
</tr>
<tr>
<td>SHARP</td>
<td>/opt/ufm/files/conf/sharp2/sharp_am.cfg</td>
</tr>
<tr>
<td>Telemetry</td>
<td>/opt/ufm/files/conf/telemetry/launch_ibdiagnet_config.ini</td>
</tr>
<tr>
<td>ALL</td>
<td>All the above configuration files.</td>
</tr>
</tbody>
</table>
Once the feature is activated and the UFM server is restarted, the UFM generates a file which lists 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:

```plaintext
$ cat /opt/ufm/files/auditing/gv.cfg.audit.log
*** Change occurred at 2022-07-24 07:31:48.679247 ***
---
+++ 
@@ -45,7 +45,7 @@
 mon_node_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.
 1-ibport_check_retries = 90
 +ibport_check_retries = 92
 ws_address = UNDEFINED
 ws_port = 8088
 ws_protocol = https
```

### 13.6 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**
13.6.1 IB Router Scripts

The following scripts are supplied as part of UFM installation package.

13.6.1.1 set_num_of_subnets.sh

- **Arguments**

  ```
  ```

- **Description** - Configures system profile to InfiniBand allowing multiple switch IDs

- **Syntax Description**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>hostname</td>
<td>IB router hostname or IP address</td>
</tr>
<tr>
<td>username</td>
<td>IB router username</td>
</tr>
<tr>
<td>password</td>
<td>IB router user password</td>
</tr>
<tr>
<td>num-of-subnets</td>
<td>Specified number of subnets (AKA SWIDs) to be initialized by the system. Value range: 2-6</td>
</tr>
</tbody>
</table>

- **Example**

  ```
  ```

  ! As a result of running this script, reboot is performed and all configuration is removed

13.6.1.2 add_interfaces_to_subnet.sh

- **Arguments**
• **Description**
Maps an interface to a subnet and enables it

• **Syntax**

```bash
/bin/ufm/scripts/ib_router/add_interfaces_to_subnet.sh  --hostname <hostname> --username <username> --password <password> --interface <interface | interface-range> --subnet <subnet>
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>hostname</td>
<td>IB router hostname or IP address</td>
</tr>
<tr>
<td>username</td>
<td>IB router username</td>
</tr>
<tr>
<td>password</td>
<td>IB router user password</td>
</tr>
</tbody>
</table>
| 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**

```bash
/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
```

13.6.1.3 **remove_interfaces_from_subnet.sh**

• **Arguments**

```bash
/bin/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**

```bash
/bin/ufm/scripts/ib_router/remove_interfaces_from_subnet.sh  --hostname <hostname> --username <username> --password <password> --interface <interface | interface-range>
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>hostname</td>
<td>IB router hostname or IP address</td>
</tr>
<tr>
<td>username</td>
<td>IB router username</td>
</tr>
<tr>
<td>password</td>
<td>IB router user password</td>
</tr>
</tbody>
</table>
| 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**

```bash
/opt/ufm/scripts/ib_router/remove_interfaces_from_subnet.sh  --hostname 10.6.204.12 --username admin --password admin --interface 1/6
```

13.6.1.4 **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**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>hostname</td>
<td>IB router hostname or IP address</td>
</tr>
<tr>
<td>username</td>
<td>IB router username</td>
</tr>
<tr>
<td>password</td>
<td>IB router user password</td>
</tr>
<tr>
<td>subnet</td>
<td>Name of IB subnet (AKA SWID): infiniband-default, infiniband-1..., infiniband-5</td>
</tr>
</tbody>
</table>

- **Example**

```
/opt/ufm/scripts/ib_router/add_subnet_to_router.sh --hostname 10.6.204.12 --username admin --password admin --subnet infiniband-3
```

⚠️ As a result of running this script, the set of commands that allow control of IB router functionality is being enabled

13.6.1.5  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**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>hostname</td>
<td>IB router hostname or IP address</td>
</tr>
<tr>
<td>username</td>
<td>IB router username</td>
</tr>
<tr>
<td>password</td>
<td>IB router user password</td>
</tr>
<tr>
<td>subnet</td>
<td>Name of IB subnet (AKA SWID): infiniband-default, infiniband-1..., infiniband-5</td>
</tr>
</tbody>
</table>

- **Example**

```
/opt/ufm/scripts/ib_router/remove_subnet_from_router.sh --hostname 10.6.204.12 --username admin --password admin --subnet infiniband-default
```

13.6.1.6  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**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-c</td>
<td>Configure new IB subnet prefix. Should be followed by new IB router subnet prefix value</td>
</tr>
<tr>
<td>-r</td>
<td>Reset to default</td>
</tr>
<tr>
<td>-h</td>
<td>Show help</td>
</tr>
</tbody>
</table>

- **Example**

  ```bash
  /opt/ufm/scripts/ib_router/set_ufm_sm_router_support.sh -c 0xfec0000000001234
  /opt/ufm/scripts/ib_router/set_ufm_sm_router_support.sh -r
  ```

### 13.6.2 IB Router Configuration

**Step 1:** Configure multi-switch. Run:

```bash
/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:

```bash
/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:

```bash
/opt/ufm/scripts/add_subnet_to_router.sh --hostname 10.6.204.12 --username admin --password admin --subnet infiniband-default
```

### 13.7 Appendix - NVIDIA SHARP Integration

#### 13.7.1 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.

13.7.2 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.

13.7.2.1 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:

```sh
[Sharp]
sharp_enabled = true
```
13.7.2.2 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 SHARP AM, you can edit the following NVIDIA SHARP AM configuration file: /opt/ufm/files/conf/sharp/sharp_am.cfg.

13.7.3 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 sm 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
```

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.
13.7.4 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
```
13.7.5 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.

13.7.6 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.

13.7.7 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.

13.7.8 NVIDIA SHARP AM Version

NVIDIA SHARP AM version can be found at /opt/ufm/sharp/share/doc/SHARP_VERSION.

13.8 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.

13.8.1 Prerequisites

- UFM 6.9.0 (or newer)
- Python 3.0 on SLURM controller
- UFM-SLURM integration files (provided independently)

13.8.2 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.

13.8.3 Manual Installation

To install the UFM-SLURM integration manually:

1. Extract the UFM-SLURM integration tar file:

   ```sh
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
   ```

13.8.4 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:

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
<th>Optionality</th>
</tr>
</thead>
<tbody>
<tr>
<td>ufm_server</td>
<td>IP of UFM server to connect to</td>
<td>Mandatory</td>
</tr>
<tr>
<td>auth_type</td>
<td>Should be <code>token_auth</code>, or <code>basic_auth</code>. If you select <code>basic_auth</code>, you need to set <code>ufm_server_user</code> and <code>ufm_server_pass</code> If you select <code>token_auth</code>, you need to set <code>token_auth</code></td>
<td>Mandatory</td>
</tr>
<tr>
<td>ufm_server_user</td>
<td>Username of UFM server used to connect to UFM, if you set <code>auth_type=basic_auth</code></td>
<td>Mandatory, depends on the <code>auth_type</code></td>
</tr>
<tr>
<td>ufm_server_pass</td>
<td>UFM server user password</td>
<td>Mandatory, depends on the <code>auth_type</code></td>
</tr>
<tr>
<td>token</td>
<td>Generated token when you set <code>auth_type</code> to <code>token_auth</code></td>
<td>Mandatory, depends on the <code>auth_type</code></td>
</tr>
<tr>
<td>pkey_allocation</td>
<td>By setting <code>pkey_allocation</code> to true, UFM SLURM Integration will use static Pkey assignment to create new Pkey, otherwise it will use the default management Pkey <code>0x7fff</code></td>
<td>Mandatory, default is True.</td>
</tr>
<tr>
<td>pkey</td>
<td>Hexadecimal string between <code>&quot;0x0001&quot;-&quot;0x7ffe&quot;</code> exclusive</td>
<td>Optional, default is &quot;0x7fff&quot; (This is the default management pkey)</td>
</tr>
<tr>
<td>ip_over_ib</td>
<td>PKey is a member in a multicast group that uses IP over InfiniBand</td>
<td>Hidden param, default is True</td>
</tr>
<tr>
<td>Attribute Name</td>
<td>Description</td>
<td>Optionality</td>
</tr>
<tr>
<td>---------------------</td>
<td>------------------------------------------------------------------------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>index0</td>
<td>If true, the API will store the PKey at index 0 of the PKey table of the GUID</td>
<td>Hidden param, default is False</td>
</tr>
<tr>
<td>sharp_allocation</td>
<td>By setting <code>sharp_allocation</code> to true, UFM SLURM Integration will create new SHARP allocation with all SLURM job IDs allocated to hosts</td>
<td>Mandatory, default is False</td>
</tr>
<tr>
<td>partially_alloc</td>
<td>By setting this to false, UFM will fail the SHARP allocation request if at least one node does not exist in the fabric</td>
<td>Optional, default is False</td>
</tr>
<tr>
<td>app_resources_limit</td>
<td>Application resources limitation</td>
<td>Hidden param, default is -1</td>
</tr>
<tr>
<td>log_file_name</td>
<td>Name of integration logging file</td>
<td>Optional</td>
</tr>
</tbody>
</table>

### 13.8.5 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.

#### 13.8.5.1 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:

```bash
curl -H "X-Remote-User:admin" -X POST 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.

### 13.8.6 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.

### 13.8.7 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
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

### 13.8.8 Running UFM-SLURM Integration

Using the SLURM controller, execute the following commands to run your batch job:

```bash
$ sbatch -N4 slurm_demo.sh
Submitted batch job 1
```

⚠️ 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:

```bash
$ 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`.

---

484
13.9 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_configuration.cfg`.

It is important to note that the file must be located in the `/opt/ufm/files` 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:

```yaml
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
    - name: "rack A" # Director switch rack name
      location:
        dc-grid-row: "A" # formalized rack location in DC
        dc-grid-column: "1" # formalized
      leaves: # List of Director switch leaves (for the rack specified)
        - guid: "0x043f720300922a00" # required field. Switch GUID.
          location-u: 1 # required field. Device location in rack: "U#"
          description: "MF0;gorilla-01:MQM9700/U1" # optional field.
      spines: # List of Director switch spines (for the rack specified)
        - guid: "0x043f720900922a00" # required field. Switch GUID.
          location-u: 10 # required field. Device location in rack: "U#"
          description: "MF0;gorilla-02:MQM9700/U1" # 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
    - name: "rack B" # Director switch rack name
      location:
        dc-grid-row: "B" # formalized rack location in DC
        dc-grid-column: "1" # formalized
      leaves: # List of Director switch leaves (for the rack specified)
        - guid: "0x093f720300922a00" # required field. Switch GUID.
          location-u: 1 # required field. Device location in rack: "U#"
          description: "MF0;gorilla-03:MQM9700/U1" # optional field.
      spines: # List of Director switch spines (for the rack specified)
        - guid: "0x093f720900922a00" # required field. Switch GUID.
          location-u: 10 # required field. Device location in rack: "U#"
          description: "MF0;gorilla-04:MQM9700/U1" # optional field.
```

13.9.1 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.

Each SuperSwitch definition can include one or more racks where each embedded rack can include multiple leafs and spines switches.

13.10 Appendix - Device Management Feature Support

The following table describes the management features available on supported devices.
<table>
<thead>
<tr>
<th><strong>Discovery</strong></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>IB L2 Discovery</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Advanced Discovery (IP, hostname, Hosts: CPU, memory, FW version)</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes with UFM Host Agent</td>
<td>No</td>
</tr>
<tr>
<td>Ethernet access Management interface</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Provisioning/Configuration</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IB Partitioning (pkey)</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>QoS: SL (SM configuration)</td>
<td>N/A</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>QoS: Rate Limit (SM configuration)</td>
<td>N/A</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Interface/VIF Configuration (IP, hostname, mtu, Bonding)</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>No</td>
<td>N/A</td>
<td>Yes with UFM Host Agent</td>
<td>No</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Device Monitoring</strong></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Device Resources: CPU, Memory, Disk</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes with UFM Host Agent</td>
<td>No</td>
</tr>
<tr>
<td>Get device alerts (Temperature, PS, Fan) <strong>Note:</strong> This feature is not supported on Switch-X switches.</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>L1 (Physical Port) - Monitoring</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>L2-3 (Interface/VIF) - Monitoring</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes with UFM Host Agent</td>
<td>No</td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>--------------------------</td>
<td>----</td>
</tr>
<tr>
<td>Congestion Monitoring per port (enables congestion map)</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Congestion Monitoring per flow (Advanced Package)</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>

## Device Management

<table>
<thead>
<tr>
<th>Add/remove to/from Rack</th>
<th>Yes</th>
<th>Yes</th>
<th>Yes</th>
<th>Yes</th>
<th>Yes</th>
<th>Yes</th>
<th>Yes</th>
<th>Yes</th>
<th>Yes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add/remove to/from Logical Server</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>View/clear Alarms</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>SSH terminal to device</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Power On</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes with IPMI</td>
<td>No</td>
</tr>
<tr>
<td>Reboot</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes (SX3606 only)</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes with IPMI</td>
<td>No</td>
</tr>
<tr>
<td>Shutdown</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes with IPMI</td>
<td>No</td>
</tr>
<tr>
<td>Port Enable/Disable</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Firmware Upgrade (HCA &amp; switch)</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes (Upon SW upgrade - SX6036 only)</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Inband Firmware Upgrade (over InfiniBand connection)</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Software Upgrade (OFED &amp; switch)</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes (SX3606 only)</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes with UFM Host Agent</td>
<td>No</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>Protocols</th>
<th>IB/</th>
<th>IB/UDP/SSH</th>
<th>IB/</th>
<th>IB/HTTP/SSH</th>
<th>IB</th>
<th>IB</th>
<th>IB</th>
<th>IB, SSH, IPMI, UDP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Communication</td>
<td>SNMP</td>
<td>IB</td>
<td>IB</td>
<td>IB/HTTP</td>
<td>IB</td>
<td>IB</td>
<td>IB</td>
<td>IB</td>
</tr>
<tr>
<td>Server - Device</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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.
# 14 Document Revision History

<table>
<thead>
<tr>
<th>Release</th>
<th>Date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.17.2</td>
<td>Jun 24, 2024</td>
<td>Added bug fix (Ref #3912416) to Bug Fixes in This Release</td>
</tr>
<tr>
<td></td>
<td>Jun 26, 2024</td>
<td>Updated Upgrading UFM on Bare Metal Server and Upgrading UFM on Docker Container</td>
</tr>
<tr>
<td>6.17.1</td>
<td>May 28, 2024</td>
<td>Updated: Bug Fixes in This Release, Known Issues in This Release Added Fast-API Plugin</td>
</tr>
<tr>
<td></td>
<td>Jun 17, 2024</td>
<td>Updated the Fast-API Plugin name, Updated Enabling UFM Authentication Server</td>
</tr>
<tr>
<td>6.17.0</td>
<td>May 7, 2024</td>
<td>Introduced a minor reorganization of the document. Updated: Changes and New Features, Bug Fixes in This Release, GNMI-Telemetry Plugin, Events Policy, Updated event policy parameters, Inventory Window, Devices Window, UFM Health Tab, UFM System Dump Tab, Device Access, Updated screenshot and parameters, Configurations of the UFM Authentication Server, Updated that it is turned on by default, Autonomous Link Maintenance (ALM) Plugin, PDR Deterministic Plugin, updated the configuration table, Low-Frequency (Secondary) Telemetry Fields, Appendix, Supported Port Counters and Events, Added and removed alarm IDs, Added: Events and Alarms, Reports, Telemetry, Events Policy Simulation, UFM Telemetry Manager (UTM) Plugin, UFM Consumer Plugin</td>
</tr>
<tr>
<td></td>
<td>May 13, 2024</td>
<td>Updated: Known Issues in This Release, Added issue #3862847, Installation Notes, Updated the latest tested FW versions in Supported NVIDIA Externally/Internally Managed Switches, PDR Deterministic Plugin</td>
</tr>
<tr>
<td></td>
<td>May 24, 2024</td>
<td>Updated: Known Issues in This Release, Added issue #3872303, UFM Telemetry Fluentd Streaming (TFS) Plugin, GNMI-Telemetry Plugin, PDR Deterministic Plugin</td>
</tr>
<tr>
<td>Release</td>
<td>Date</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>6.16.0</td>
<td>Feb 8, 2024</td>
<td>Updated:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Changes and New Features</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Bug Fixes in This Release</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Installation Notes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Secondary Telemetry - Added the secondary_slvl_support flag and information on the default counters and added Secondary Telemetry Exposing IPv6 Counters</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• PDR Deterministic Plugin - Updated instructions</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Link Analysis - Updated GUI screenshots</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Device Cable Tab - Updated GUI screenshots</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Cables Window - Updated GUI screenshots</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Packet Mirroring Collector (PMC) Plugin - Updated overview and GUI screenshots</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• General Tab, Inventory Tab and Inventory Window - Updated GUI screenshots</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• SM Congestion Control Configuration - Updated GUI screenshot</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Autonomous Link Maintenance (ALM) Plugin - Added GUI screenshots in ALM UI</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Appendix - UFM Subnet Manager Default Properties - Updated the max_op_vls from 3 to 2.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Added:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Kerberos Authentication and Enabling Kerberos Authentication</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Secondary Telemetry Exposing IPv6 Counters</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Dynamic Telemetry</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Configuring Syslog</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Configuring UFM Logging</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Appendix - OpenSM Configuration Files for Congestion Control</td>
</tr>
<tr>
<td></td>
<td>Mar 6, 2024</td>
<td>Updated:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Troubleshooting</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Known Issues in This Release</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Installation Notes</td>
</tr>
<tr>
<td>6.15.2</td>
<td>Jan 4, 2024</td>
<td>Updated:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Changes and New Features</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Known Issues in This Release</td>
</tr>
<tr>
<td></td>
<td>Jan 23, 2024</td>
<td>Added a note to Installation Notes about the UFM SM version</td>
</tr>
<tr>
<td>6.15.1</td>
<td>Dec 14, 2023</td>
<td>Updated:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Bug Fixes in This Release</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Known Issues in This Release</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Supported NVIDIA Internally Managed Switches - Removed MTX6100, MTX6240 and MTX6280 switches and the SX6036G (FDR) gateway</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Installation Notes - Updated with the new MFT package version</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• System Requirements - Added MLNX_OFED23.x</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Unsupported Functionalities/Features</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Added:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Cable Validation Report in Subnet Merger</td>
</tr>
<tr>
<td></td>
<td>Dec 19, 2023</td>
<td>Updated:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Changes and New Features</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Added a Known issue to Bug Fixes in This Release</td>
</tr>
<tr>
<td>Release</td>
<td>Date</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>------------</td>
<td>-------------</td>
</tr>
<tr>
<td>6.15.0</td>
<td>Nov 5, 2023</td>
<td>Updated:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Changes and New Features</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Bug Fixes in This Release</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Azure Authentication Login Page - Introduced new Azure authentication login page</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Enabling Azure AD Authentication - Added further instructions</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- UFM Logs Tab - Added log occurrences display</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Added:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Events History</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Device Status Events</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Link Status Events</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- GNMI-Telemetry Plugin</td>
</tr>
<tr>
<td></td>
<td></td>
<td>In Secondary Telemetry, added instructions on Exposing Switch Aggregation Nodes Telemetry and Stopping Telemetry Endpoint Using CLI Command</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- UFM Authentication Server</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Enabling UFM Authentication Server</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Low-Frequency (Secondary) Telemetry Fields</td>
</tr>
<tr>
<td>6.14.1</td>
<td>Aug 31, 2023</td>
<td>Updated:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Changes and New Features</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Bug Fixes in This Release</td>
</tr>
<tr>
<td></td>
<td>Oct 17, 2023</td>
<td>Updated:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- System Requirements</td>
</tr>
<tr>
<td>6.14.0</td>
<td>Aug 10, 2023</td>
<td>Updated:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Changes and New Features</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Bug Fixes in This Release</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Known Issues in This Release</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Plugin Management</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Secondary Telemetry</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- PDR Deterministic Plugin - Updated step 3 in “Deployment”</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- rest-rdma Plugin</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- NDT Plugin</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Autonomous Link Maintenance (ALM) Plugin</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Appendix - Supported Port Counters and Events - Added alarm ID #134, 1602 and 1603 and status column for all alarm IDs.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Added:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Disabling Rest Roles Access Control</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Enabling Azure AD Authentication</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Azure AD Authentication</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Health Policy Management</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Rest Roles Access Control</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- UFM Factory Reset</td>
</tr>
<tr>
<td>6.13.1</td>
<td>May 18, 2023</td>
<td>Updated:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Changes and New Features</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Bug Fixes in This Release</td>
</tr>
<tr>
<td>Release</td>
<td>Date</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>------------</td>
<td>-------------</td>
</tr>
<tr>
<td>6.13.0</td>
<td>May 5, 2023</td>
<td>Updated:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Changes and New Features</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Bug Fixes in This Release</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Known Issues in This Release</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Email - Added time zone preference</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- NDT Plugin</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- UFM Telemetry Fluentd Streaming (TFS) Plugin - Updated REST API</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- UFM System Dump Tab</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Appendix - Supported Port Counters and Events</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Added:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Multi-Subnet UFM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Enable Network Fast Recovery</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- NDT Format Merger</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Subnet Merger UI</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Added the following Plugins:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- UFM Bright Cluster Integration Plugin</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- UFM Cyber-AI Plugin</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Autonomous Link Maintenance (ALM) Plugin</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- DTS Plugin</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Sysinfo Plugin</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- SNMP Plugin</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Packet Mirroring Collector (PMC) Plugin</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- PDR Deterministic Plugin</td>
</tr>
<tr>
<td></td>
<td>May 9, 2023</td>
<td>Updated</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Known Issues in This Release</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Appendix - Enhanced Quality of Service - Updated notes and example</td>
</tr>
<tr>
<td>6.12.1</td>
<td>Feb 19, 2023</td>
<td>Updated</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Changes and New Features</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Bug Fixes in This Release</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Known Issues in This Release</td>
</tr>
<tr>
<td></td>
<td>Mar 1, 2023</td>
<td>Updated Changes and New Features</td>
</tr>
<tr>
<td></td>
<td>Mar 16, 2023</td>
<td>Updated Changes and New Features - Added MFT package integration details</td>
</tr>
<tr>
<td></td>
<td>Mar 27, 2023</td>
<td>Updated UFM Server Communication with Externally Managed Switches</td>
</tr>
<tr>
<td>6.12.0</td>
<td>Feb 2, 2023</td>
<td>Updated:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Changes and New Features</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Bug Fixes in This Release</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Known Issues in This Release</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Configuring Partial Switch ASIC Failure Events</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Updated example in Multi-port SM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- UFM System Dump Tab</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Appendix - Used Ports</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Appendix - UFM SLURM Integration</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Added:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Added a note under Ports Window</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Added a note under Unhealthy Ports Window</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Delegate Authentication to a Proxy</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Removed:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- UFM Logical Elements tab from the Web UI</td>
</tr>
<tr>
<td></td>
<td>Feb 6, 2023</td>
<td>Updated Troubleshooting</td>
</tr>
<tr>
<td>Release</td>
<td>Date</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>------------</td>
<td>-------------</td>
</tr>
<tr>
<td>6.11.1</td>
<td>Dec 1, 2022</td>
<td>Updated:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Changes and New Features to include the upgrade of NVIDIA SHARP SW version</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Installation Notes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Known Issues in This Release</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Troubleshooting</td>
</tr>
<tr>
<td>Dec 19, 2022</td>
<td>Updated Changes and New Features</td>
<td></td>
</tr>
<tr>
<td>6.11.0</td>
<td>Nov 21, 2022</td>
<td>Updated:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Added a link to UFM SDK 3.0 under Related Documentation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Changes and New Features</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Installation Notes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Bug Fixes in This Release</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Known Issues in This Release</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Installing UFM HA Package</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Network Map with new screenshots and new instructions for Map Information and Settings</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Devices Window with new screenshots</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- PSID and Firmware Version In-Band Discovery</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Groups Window with new screenshots</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Table Enhancements with new screenshots</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- UFM Telemetry Fluentd Streaming (TFS) Plugin</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Enabling UFM Telemetry</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Added:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- CPU Affinity on UFM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Switch Management IP Address Discovery</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- UFM Events Fluent Streaming (EFS) Plugin</td>
</tr>
<tr>
<td></td>
<td></td>
<td>In Telemetry - User-Defined Sessions</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Changing UFM Telemetry Default Configuration</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Supporting Generic Counters Parsing and Display</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Supporting Multiple Telemetry Instances Fetch</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Secondary Telemetry</td>
</tr>
<tr>
<td>Release</td>
<td>Date</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| 6.10.0    | July 31, 2022 | Updated:  
- Release Notes  
- UFM Installation and Initial Configuration  
- Installation Notes  
- UFM Software Architecture  
- Network Management  
- Subnet Manager Tab  
- Non-Optimal Links  
- Cable Transceiver Temperatures  
- Telemetry - User-Defined Sessions  
- Network Management  
- Supported Actions for Internally Managed Switches  
- Appendix - NVIDIA SHARP Integration  
- Appendix - SM Default Files  
- Appendix - UFM Subnet Manager Default Properties  
- Appendix - SM Activity Report  
- Appendix - Configuration Files Auditing  
- Appendix - Enhanced Quality of Service  
- Appendix – Partitioning  
- Appendix – Diagnostic Utilities  
- Appendix – Adaptive Routing  
- Appendix – UFM SLURM Integration  
Added:  
- Showing UFM Processes Status  
- Plugin Management  
- Appendix - Configuration Files Auditing  |
| September 2022 |            | Updated:  
- Appendix – UFM Event Forwarder  
- NDR switches firmware version in Supported NVIDIA Externally Managed Switches.  
- Licensing  
- License Devices limit in UFM Health Tab  
- Operating NVIDIA SHARP AM with UFM  
- Changes and New Features  
- Unsupported Functionalities/Features  |
| October 2022 |            | Updated the examples in Docker Installation |
15 EULA, Legal Notices and 3rd Party Licenses

15.1 Legal Notice

15.2 Third-Party Licenses

15.3 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. 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. **Export Compliance.** 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)
15.3.1 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](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.

<table>
<thead>
<tr>
<th>Component name</th>
<th>Version</th>
<th>Home Page</th>
<th>License</th>
</tr>
</thead>
<tbody>
<tr>
<td>@candlefw/wick</td>
<td>0.8.12</td>
<td><a href="https://github.com/galactrax/cfw-wick#readme">https://github.com/galactrax/cfw-wick#readme</a></td>
<td>MIT License</td>
</tr>
<tr>
<td>ABSender</td>
<td>master-20121122</td>
<td><a href="https://github.com/100Continue/ABSender">https://github.com/100Continue/ABSender</a></td>
<td>Apache License 2.0</td>
</tr>
<tr>
<td>APBS</td>
<td>apbs-0.3.1</td>
<td><a href="https://sourceforge.net/projects/apbs">https://sourceforge.net/projects/apbs</a></td>
<td>GNU General Public License v2.0 or later</td>
</tr>
<tr>
<td>Amiga Research OS</td>
<td>20120217</td>
<td><a href="https://aros.sourceforge.io/license.html">https://aros.sourceforge.io/license.html</a></td>
<td>Aros Public License V 1.1</td>
</tr>
<tr>
<td>Apache ActiveMQ</td>
<td>2.2.2</td>
<td><a href="http://activemq.apache.org/">http://activemq.apache.org/</a></td>
<td>Apache License 2.0</td>
</tr>
<tr>
<td>Apache HTTP Server</td>
<td>1.3.7, 1.3.8</td>
<td><a href="http://httpd.apache.org/">http://httpd.apache.org/</a></td>
<td>Apache License 1.0</td>
</tr>
<tr>
<td>Apache HTTP Server</td>
<td>2, 2.0.11, 2.0.23, 2.0.25, 2.0.26, 2.0.30, 2.0.33, 2.0.35, 2.0.36, 2.0.38, 2.0.39, 2.0.40, 2.0.41, 2.0.43, 2.1.0</td>
<td><a href="http://httpd.apache.org/">http://httpd.apache.org/</a></td>
<td>Apache License 1.1</td>
</tr>
<tr>
<td>Apache HTTP Server</td>
<td>2.0.59, 2.1.1, 2.1.10, 2.1.2, 2.1.3, 2.1.4, 2.1.5, 2.1.6, 2.1.7, 2.1.8, 2.1.9, 2.2.1, 2.2.2, 2.2.12, 2.2.13, 2.2.14, 2.2.15, 2.2.16, 2.2.17, 2.2.22, 2.2.26, 2.2.3, 2.2.4, 2.2.5, 2.2.6, 2.2.7, 2.2.9, 2.3.0, 2.3.1, 2.3.4</td>
<td><a href="http://httpd.apache.org/">http://httpd.apache.org/</a></td>
<td>Apache License 2.0</td>
</tr>
<tr>
<td>Apache HTTP Server</td>
<td>STRIKER_2_1_0_RC1</td>
<td><a href="http://httpd.apache.org/">http://httpd.apache.org/</a></td>
<td>Apache License 2.0</td>
</tr>
<tr>
<td>Component name</td>
<td>Version</td>
<td>Home Page</td>
<td>License</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>----------------------------------------------</td>
<td>---------------------------------</td>
<td>----------------------------------------------</td>
</tr>
<tr>
<td>Apache Portable Runtime</td>
<td>0.9.13, 0.9.15, 1.2.0, 1.2.10, 1.2.11, 1.2.12, 1.2.7, 1.2.8, 1.2.9, 1.3.0, 1.3.1, 1.3.10, 1.3.12, 1.3.2, 1.3.3, 1.3.4, 1.3.5, 1.3.7, 1.3.8, 1.3.9, 1.4.7, 1.5.1, 1.5.2; APR_1_0_RC2; JCW_0_9_5_PRE1</td>
<td><a href="http://apr.apache.org/">http://apr.apache.org/</a></td>
<td>Apache License 2.0</td>
</tr>
<tr>
<td>Apache Portable Runtime</td>
<td>0.9.4  APACHE_2_0_37 APACHE_2_0_40 APACHE_2_0_44 APACHE_2_0_48</td>
<td><a href="http://apr.apache.org/">http://apr.apache.org/</a></td>
<td>Apache License 1.1</td>
</tr>
<tr>
<td>Apache Portable Runtime</td>
<td>APU_1_0_RC1</td>
<td><a href="http://apr.apache.org/">http://apr.apache.org/</a></td>
<td>(MIT License AND RSA Message-Digest License AND Apache License 2.0 AND Beerware License AND RSA MD4 or MD5 Message-Digest Algorithm License AND Christian Michelsen Research License AND Apache License 1.1)</td>
</tr>
<tr>
<td>Apache Tomcat</td>
<td>1.1.0, 6.0.24</td>
<td><a href="http://tomcat.apache.org/">http://tomcat.apache.org/</a></td>
<td>Apache License 2.0</td>
</tr>
<tr>
<td>BIND9 (Berkeley Internet Name Domain)</td>
<td>9.9.11</td>
<td><a href="https://www.isc.org/wordpress/software/bind/">https://www.isc.org/wordpress/software/bind/</a></td>
<td>Mozilla Public License 2.0</td>
</tr>
<tr>
<td>Berkeley DB</td>
<td>4.5.20</td>
<td><a href="http://www.oracle.com/technology/products/berkeley-db/db/index.html">http://www.oracle.com/technology/products/berkeley-db/db/index.html</a></td>
<td>BSD 3-clause &quot;New&quot; or &quot;Revised&quot; License</td>
</tr>
<tr>
<td>Chromium (Google Chrome)</td>
<td>32.0.1700.102</td>
<td><a href="http://code.google.com/chromium/">http://code.google.com/chromium/</a></td>
<td>BSD 3-clause &quot;New&quot; or &quot;Revised&quot; License</td>
</tr>
<tr>
<td>Cinder</td>
<td>v0.8.0</td>
<td><a href="http://libcinder.org">http://libcinder.org</a></td>
<td>BSD 3-clause &quot;New&quot; or &quot;Revised&quot; License</td>
</tr>
<tr>
<td>Component name</td>
<td>Version</td>
<td>Home Page</td>
<td>License</td>
</tr>
<tr>
<td>-----------------------------------------</td>
<td>------------------</td>
<td>---------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Clonezilla</td>
<td>1.2.10</td>
<td><a href="http://clonezilla.org/">http://clonezilla.org/</a></td>
<td>GNU General Public License v3.0 or later</td>
</tr>
<tr>
<td>Cron</td>
<td>3.0pl1</td>
<td><a href="https://alioth.debian.org/projects/pkg-cron/">https://alioth.debian.org/projects/pkg-cron/</a></td>
<td>Cron License</td>
</tr>
<tr>
<td>CyanogenMod - android_external_busybox</td>
<td>cm-10.1-M1, cm-10.1-M2</td>
<td><a href="https://github.com/CyanogenMod/android_external_busybox/blob/cm-12.0/LICENSE">https://github.com/CyanogenMod/android_external_busybox/blob/cm-12.0/LICENSE</a></td>
<td>GNU General Public License v2.0 or later</td>
</tr>
<tr>
<td>D-Bus</td>
<td>1.2.6</td>
<td><a href="http://www.freedesktop.org/wiki/Software/dbus">http://www.freedesktop.org/wiki/Software/dbus</a></td>
<td>Academic Free License v2.1</td>
</tr>
<tr>
<td>Darik's Boot and Nuke</td>
<td>dban-2.0.0</td>
<td><a href="http://sourceforge.net/projects/dban">http://sourceforge.net/projects/dban</a></td>
<td>(GNU Lesser General Public License v3.0 or later AND GNU General Public License v3.0 or later)</td>
</tr>
<tr>
<td>Debian Games</td>
<td>11.04.1+repack</td>
<td><a href="http://wiki.debian.org/Games">http://wiki.debian.org/Games</a></td>
<td>BSD 3-clause “New” or “Revised” License</td>
</tr>
<tr>
<td>FLAC - Free Lossless Audio Codec</td>
<td>flac-1.1.1-beta1-src</td>
<td><a href="http://flac.sourceforge.net">http://flac.sourceforge.net</a></td>
<td>BSD 3-clause “New” or “Revised” License</td>
</tr>
<tr>
<td>FarGroup/FarManager</td>
<td>builds/3.0.2890</td>
<td><a href="https://github.com/FarGroup/FarManager/blob/master/LICENSE">https://github.com/FarGroup/FarManager/blob/master/LICENSE</a></td>
<td>BSD 3-clause “New” or “Revised” License</td>
</tr>
<tr>
<td>FreeBSD</td>
<td>5.5, 6, 9.0-BETA1, release/11.2.0.12.2, 2.2.0, 2.2.6, 5.0.0cvs</td>
<td><a href="https://github.com/freebsd/freebsd">https://github.com/freebsd/freebsd</a></td>
<td>BSD 2-clause “Simplified” License</td>
</tr>
<tr>
<td>FreeBSD</td>
<td>bsd_44_lite</td>
<td><a href="https://github.com/freebsd/freebsd">https://github.com/freebsd/freebsd</a></td>
<td>BSD 4-clause “Original” or “Old” License</td>
</tr>
<tr>
<td>FreeBSD Ports</td>
<td>RELEASE_4_5_0</td>
<td><a href="https://www.freebsd.org/ports/">https://www.freebsd.org/ports/</a></td>
<td>BSD 2-clause FreeBSD License</td>
</tr>
<tr>
<td>FreeBSD Ports</td>
<td>RELEASE_4_6_0</td>
<td><a href="https://www.freebsd.org/ports/">https://www.freebsd.org/ports/</a></td>
<td>BSD 2-clause FreeBSD License</td>
</tr>
<tr>
<td>FreeNAS</td>
<td>0.7</td>
<td><a href="https://www.freenas.org/">https://www.freenas.org/</a></td>
<td>BSD 3-clause “New” or “Revised” License</td>
</tr>
<tr>
<td>GD</td>
<td>2.0.1beta, 2.0.32, 2.0.33, 2.0.34RC1, 2.0.35, 2.0.35RC5</td>
<td><a href="http://www.libgd.org">http://www.libgd.org</a></td>
<td>GD License</td>
</tr>
<tr>
<td>GD</td>
<td>2.0.36_rc1</td>
<td><a href="http://www.libgd.org">http://www.libgd.org</a></td>
<td>(X11 License OR MIT License)</td>
</tr>
<tr>
<td>GLib</td>
<td>1.2.3, 2.14.6, 2.19.5</td>
<td><a href="http://library.gnome.org/devel/lib/">http://library.gnome.org/devel/lib/</a></td>
<td>Apache License 2.0</td>
</tr>
<tr>
<td>GNU Compiler Collection</td>
<td>4.7.0</td>
<td><a href="http://gcc.gnu.org/">http://gcc.gnu.org/</a></td>
<td>(GD License OR Unknown License)</td>
</tr>
<tr>
<td>GNU Libtool</td>
<td>1.4.1</td>
<td><a href="http://www.gnu.org/software/libtool/">http://www.gnu.org/software/libtool/</a></td>
<td>BSD 3-clause “New” or “Revised” License</td>
</tr>
<tr>
<td>Component name</td>
<td>Version</td>
<td>Home Page</td>
<td>License</td>
</tr>
<tr>
<td>----------------------------------------</td>
<td>------------------------------</td>
<td>-------------------------------------</td>
<td>--------------------------------------------------------------------------</td>
</tr>
<tr>
<td>GNU Parted</td>
<td>1.8.1, 2.4</td>
<td><a href="http://www.gnu.org/software/parted">http://www.gnu.org/software/parted</a></td>
<td>GNU General Public License v2.0 or later</td>
</tr>
<tr>
<td>GNU Parted</td>
<td>2.4</td>
<td><a href="http://www.gnu.org/software/parted">http://www.gnu.org/software/parted</a></td>
<td>GNU General Public License v3.0 or later</td>
</tr>
<tr>
<td>Gentoo Linux</td>
<td>release_1_3_17</td>
<td><a href="https://www.gentoo.org/">https://www.gentoo.org/</a></td>
<td>GNU General Public License v2.0 or later</td>
</tr>
<tr>
<td>Heimdal Kerberos</td>
<td>heimdali-0.0n</td>
<td><a href="http://www.h5i.org/">http://www.h5i.org/</a></td>
<td>BSD 3-clause &quot;New&quot; or &quot;Revised&quot; License</td>
</tr>
<tr>
<td>HipHop Virtual Machine for PHP</td>
<td>HHVM-3.1.0</td>
<td><a href="https://github.com/facebook/hhvm">https://github.com/facebook/hhvm</a></td>
<td>(PHP License v3.01 AND Zend License v2.0)</td>
</tr>
<tr>
<td>Kablink</td>
<td>1.1 Alpha1</td>
<td><a href="https://www.kablink.org/">https://www.kablink.org/</a></td>
<td>Apache License 2.0</td>
</tr>
<tr>
<td>Less</td>
<td>374</td>
<td><a href="http://www.greenwoodsoftware.com/less/">http://www.greenwoodsoftware.com/less/</a></td>
<td>BSD 2-clause “Simplified” License</td>
</tr>
<tr>
<td>Less</td>
<td>429</td>
<td><a href="http://www.greenwoodsoftware.com/less/">http://www.greenwoodsoftware.com/less/</a></td>
<td>GNU General Public License v2.0 or later OR Less License</td>
</tr>
<tr>
<td>LineageOS</td>
<td>cm-10.1.0-RC1</td>
<td><a href="https://lineageos.org/">https://lineageos.org/</a></td>
<td>(FSF Unlimited License AND BSD 3-clause &quot;New&quot; or &quot;Revised&quot; License)</td>
</tr>
<tr>
<td>Linux Test Project</td>
<td>2004</td>
<td><a href="https://github.com/linux-test-project/ltpp">https://github.com/linux-test-project/ltpp</a></td>
<td>GNU General Public License v2.0 or later</td>
</tr>
<tr>
<td>Linux-Pam</td>
<td>0.59, 0.72, 0.74, 0.76, 0.99.1.0, 0.99.2.0, 0.99.4.0, 0.99.5.0, 0.99.6.1, 0.99.6.2, 1.0.0</td>
<td><a href="http://www.linux-pam.org">http://www.linux-pam.org</a></td>
<td>BSD 3-clause &quot;New&quot; or &quot;Revised&quot; License</td>
</tr>
<tr>
<td>Linux-Pam</td>
<td>1.0.1</td>
<td><a href="http://www.linux-pam.org">http://www.linux-pam.org</a></td>
<td>(X11 License AND FSF Unlimited License)</td>
</tr>
<tr>
<td>MapServer</td>
<td>rel-1-0-0</td>
<td><a href="http://mapserver.org">http://mapserver.org</a></td>
<td>(X11 License AND MIT License)</td>
</tr>
<tr>
<td>Merruk-Technology</td>
<td>2.0-20121113</td>
<td><a href="http://www.merruk.ma">http://www.merruk.ma</a></td>
<td>GNU General Public License v2.0 only</td>
</tr>
<tr>
<td>MinGW - Minimalist GNU for Windows</td>
<td>binutils-2.20</td>
<td><a href="http://mingw.sourceforge.net/">http://mingw.sourceforge.net/</a></td>
<td>Public Domain</td>
</tr>
<tr>
<td>MythTV</td>
<td>v0.13</td>
<td><a href="http://www.mythtv.org">http://www.mythtv.org</a></td>
<td>GNU General Public License v2.0 or later</td>
</tr>
<tr>
<td>NFS</td>
<td>1.0.6</td>
<td><a href="http://linux-nfs.org/">http://linux-nfs.org/</a></td>
<td>GNU General Public License v2.0 or later</td>
</tr>
<tr>
<td>Component name</td>
<td>Version</td>
<td>Home Page</td>
<td>License</td>
</tr>
<tr>
<td>----------------</td>
<td>------------------------------------------------------------------------</td>
<td>---------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Net-SNMP</td>
<td>5.0.9, 5.4.2.1, 5.5.2.pre1, 5.7.3, END-UCD-SNMP Ext-5-3-cvs20050331, JBNP-CBL-1, 5.0.11.1, 5.2.2</td>
<td><a href="http://www.net-snmp.org">http://www.net-snmp.org</a></td>
<td>(CMU License AND BSD 3-clause “New” or “Revised” License)</td>
</tr>
<tr>
<td>Net-SNMP</td>
<td>5.1.2, Ext-5-0, Ext-5-0-2, Ext-5-0-4, Ext-5-4-1-1, V4-2-patches-merge2</td>
<td><a href="http://www.net-snmp.org">http://www.net-snmp.org</a></td>
<td>(Diffstat License OR BSD 3-clause “New” or “Revised” License)</td>
</tr>
<tr>
<td>Net-SNMP</td>
<td>Ext-5-0, Ext-5-0-4</td>
<td><a href="http://www.net-snmp.org">http://www.net-snmp.org</a></td>
<td>(Diffstat License AND BSD 3-clause “New” or “Revised” License AND Christian Michelsen Research License)</td>
</tr>
<tr>
<td>Net-SNMP</td>
<td>Ext-5-4-1-1</td>
<td><a href="http://www.net-snmp.org">http://www.net-snmp.org</a></td>
<td>(Diffstat License AND BSD 3-clause “New” or “Revised” License AND Christian Michelsen Research License AND Bzip2 License)</td>
</tr>
<tr>
<td>Net-SNMP</td>
<td>V4-2-patches-merge2</td>
<td><a href="http://www.net-snmp.org">http://www.net-snmp.org</a></td>
<td>Diffstat License AND Christian Michelsen Research License</td>
</tr>
<tr>
<td>Net-SNMP</td>
<td>5.2.4 source code, 5.2.5 pre-releases, 5.3.1, 5.3.2 pre-releases, 5.4.2 pre-releases, 5.5, Ext-4-0-pre5, Ext-4-1-pre1, Ext-5-0-2-pre1,Ext-5-0-7-pre1, Ext-5-0-8-pre1, Ext-5-2-2rc6, Ext-5-2-pre2, Ext-5-2-pre3, Ext-5-3-pre1, Ext-5-3-pre3, Ext-5-3-pre4, Ext-5-4-1-pre1, Ext-5-4-1-pre3, Ext-5-4-pre1, Ext-5-4-pre4, Ext-5-5-pre1, Ext-5-5-pre2, Ext-5-5-pre3, Ext-5-5-rc1, Ext-5-5-rc3, 5.3.0.1. 5.8.1.pre1, 5.8.1.pre2</td>
<td><a href="http://www.net-snmp.org">http://www.net-snmp.org</a></td>
<td>BSD 3-clause “New” or “Revised” License</td>
</tr>
<tr>
<td>NetBSD</td>
<td>1.1, 1.5, 2</td>
<td><a href="http://www.netbsd.org">http://www.netbsd.org</a></td>
<td>BSD 3-clause “New” or “Revised” License</td>
</tr>
<tr>
<td>Component name</td>
<td>Version</td>
<td>Home Page</td>
<td>License</td>
</tr>
<tr>
<td>-----------------------------------------------------</td>
<td>-------------------</td>
<td>-----------------------------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>OpenFabrics Enterprise Distribution - OFED</td>
<td>1.2, 1.5, 3.3.2018</td>
<td><a href="https://www.openfabrics.org/downloads/rdmacm/">https://www.openfabrics.org/downloads/rdmacm/</a></td>
<td>BSD 2-clause “Simplified” License</td>
</tr>
<tr>
<td>OpenFabrics Enterprise Distribution - OFED</td>
<td>3.1.8</td>
<td><a href="https://www.openfabrics.org/downloads/rdmacm/">https://www.openfabrics.org/downloads/rdmacm/</a></td>
<td>BSD 3-clause “New” or “Revised” License</td>
</tr>
<tr>
<td>OpenLDAP</td>
<td>2.4.44</td>
<td><a href="http://www.openldap.org/">http://www.openldap.org/</a></td>
<td>Open LDAP Public License v2.8</td>
</tr>
<tr>
<td>OpenSSH</td>
<td>5.3p1, 7.4p1,7.7, 7.7p1, 7.8, 7.8p1, 7.9, 7.9p1, 8.0p1, pre-reorder</td>
<td><a href="http://www.openssh.com/">http://www.openssh.com/</a></td>
<td>BSD 3-clause “New” or “Revised” License</td>
</tr>
<tr>
<td>OpenSSH</td>
<td>7.2p2, 7.6p1</td>
<td><a href="http://www.openssh.com/">http://www.openssh.com/</a></td>
<td>X11 License</td>
</tr>
<tr>
<td>OpenWrt</td>
<td>12.09, 14.07</td>
<td><a href="http://openwrt.org/">http://openwrt.org/</a></td>
<td>GNU General Public License v2.0 or later</td>
</tr>
<tr>
<td>PCRE</td>
<td>7.1, 7.4, 7.6</td>
<td><a href="http://www.pcre.org/">http://www.pcre.org/</a></td>
<td>PCRE License</td>
</tr>
<tr>
<td>PCRE</td>
<td>4, 7.6, 7.7, 7.8</td>
<td><a href="http://www.pcre.org/">http://www.pcre.org/</a></td>
<td>BSD 3-clause “New” or “Revised” License</td>
</tr>
<tr>
<td>PHP</td>
<td>MERGE_FROM_NEW_LO OK_2001_TAG_1</td>
<td><a href="http://svn.php.net">http://svn.php.net</a></td>
<td>BSD 2-clause “Simplified” License</td>
</tr>
<tr>
<td>PortableApps.com</td>
<td>WinMerge 2.10.0 , 2.6.12Source</td>
<td><a href="http://portableapps.com/">http://portableapps.com/</a></td>
<td>Apache License 2.0</td>
</tr>
<tr>
<td>Python programming language</td>
<td>v2.4a2</td>
<td><a href="https://www.python.org">https://www.python.org</a></td>
<td>Python Software Foundation License 2.0</td>
</tr>
<tr>
<td>Qualcomm Kernel Tree for MSM/QSD family and Android 4.4</td>
<td>ath-201808291719</td>
<td><a href="https://www.codeaurora.org/projects/all-active-projects/linux-msm">https://www.codeaurora.org/projects/all-active-projects/linux-msm</a></td>
<td>ISC License</td>
</tr>
<tr>
<td>TACACS+ client library and PAM module</td>
<td>1.2.10, 1.2.9</td>
<td><a href="https://sourceforge.net/projects/tacplus">https://sourceforge.net/projects/tacplus</a></td>
<td>BSD 3-clause “New” or “Revised” License</td>
</tr>
<tr>
<td>TACACS+ client library and PAM module</td>
<td>1.3.2</td>
<td><a href="https://sourceforge.net/projects/tacplus">https://sourceforge.net/projects/tacplus</a></td>
<td>GNU General Public License v2.0 or later</td>
</tr>
<tr>
<td>Tarifa</td>
<td>Tarifa019.tar</td>
<td><a href="http://sourceforge.net/projects/tarifa">http://sourceforge.net/projects/tarifa</a></td>
<td>GNU General Public License v2.0 or later</td>
</tr>
<tr>
<td>Tcl/Tk</td>
<td>8.1.1</td>
<td><a href="http://www.tcl.tk/">http://www.tcl.tk/</a></td>
<td>TCL/TK License</td>
</tr>
<tr>
<td>Tecla Library</td>
<td>1.2.3, 1.4.0, 1.4.1, 1.5.0, 1.6.0, 1.6.2</td>
<td><a href="http://www.astro.caltech.edu/~mcs/tecla/index.html">http://www.astro.caltech.edu/~mcs/tecla/index.html</a></td>
<td>MIT License</td>
</tr>
<tr>
<td>Component name</td>
<td>Version</td>
<td>Home Page</td>
<td>License</td>
</tr>
<tr>
<td>------------------------</td>
<td>---------------</td>
<td>-----------------------------------------------</td>
<td>----------------------------------------------</td>
</tr>
<tr>
<td>The GWARE Project</td>
<td>2.10.2</td>
<td><a href="http://sourceforge.net/projects/gware">sourceforge.net/projects/gware</a></td>
<td>GNU Lesser General Public License v2.1 or later</td>
</tr>
<tr>
<td>TizenRT</td>
<td>1.1_Public_Release</td>
<td><a href="https://github.com/Samsung/TizenRT">github.com/Samsung/TizenRT</a></td>
<td>Apache License 2.0</td>
</tr>
<tr>
<td>UC-7402.7408.7410.7420-LX Plus Source</td>
<td>20100210</td>
<td><a href="http://www.moxa.com/product/UC-7408.htm">moxa.com/product/UC-7408.htm</a></td>
<td>GNU General Public License v2.0 only</td>
</tr>
<tr>
<td>WinMerge</td>
<td>2.11.1.7</td>
<td><a href="https://winmerge.org/">winmerge.org/</a></td>
<td>Apache License 2.0</td>
</tr>
<tr>
<td>XAMPP</td>
<td>1.4.5, 1.6.4</td>
<td><a href="https://www.apachefriends.org/index.html">apachefriends.org/index.html</a></td>
<td>BSD 3-clause “New” or “Revised” License</td>
</tr>
<tr>
<td>XAMPP</td>
<td>1.6.4</td>
<td><a href="https://www.apachefriends.org/index.html">apachefriends.org/index.html</a></td>
<td>GNU General Public License v2.0 or later</td>
</tr>
<tr>
<td>XQilla</td>
<td>1.1.0</td>
<td><a href="http://xqilla.sourceforge.net">xqilla.sourceforge.net</a></td>
<td>BSD 3-clause “New” or “Revised” License</td>
</tr>
<tr>
<td>YaST</td>
<td>broken/svn/openSUSE-9_3</td>
<td><a href="http://opensuse.org/YaST">opensuse.org/YaST</a></td>
<td>MIT License</td>
</tr>
<tr>
<td>Zile (Zile is Lossy Emacs)</td>
<td>1.4, 1.5, 1.5.2, 1.5.3, 1.6, 1.6.1, 1.6.2</td>
<td><a href="http://zile.sourceforge.net">zile.sourceforge.net</a></td>
<td>GNU General Public License v2.0 or later</td>
</tr>
<tr>
<td>afwall</td>
<td>V2.6.0.1, v2.8.0, v2.9.0, v2.9.1, v2.9.4</td>
<td><a href="https://github.com/ukanth/afwall">ukanth/afwall</a></td>
<td>MIT License</td>
</tr>
<tr>
<td>alcatel</td>
<td>20</td>
<td><a href="http://www.alcatel-mobilephones.com/">alcatel-mobilephones.com/</a></td>
<td>Apache License 2.0</td>
</tr>
<tr>
<td>alcatel</td>
<td>4/18/2012, 20120601, 918</td>
<td><a href="http://www.alcatel-mobilephones.com/">alcatel-mobilephones.com/</a></td>
<td>GNU General Public License v2.0 or later</td>
</tr>
<tr>
<td>appweb</td>
<td>3.0B.0-0</td>
<td><a href="http://code.google.com/p/appweb">code.google.com/p/appweb</a></td>
<td>Apache License 2.0</td>
</tr>
<tr>
<td>asuswrt-merlin</td>
<td>376.48, 376.48, 380.62</td>
<td><a href="https://github.com/RMerl/asuswrt-merlin">github.com/RMerl/asuswrt-merlin</a></td>
<td>Artistic License 1.0</td>
</tr>
<tr>
<td>asuswrt-merlin</td>
<td>378.51, 380.62</td>
<td><a href="https://github.com/RMerl/asuswrt-merlin">github.com/RMerl/asuswrt-merlin</a></td>
<td>GNU General Public License v2.0 or later</td>
</tr>
<tr>
<td>avahi</td>
<td>v0.6</td>
<td><a href="http://avahi.org">avahi.org</a></td>
<td>GNU Lesser General Public License v2.1 or later</td>
</tr>
<tr>
<td>awokengazebo-lfi</td>
<td>lfi-20080723</td>
<td><a href="http://www.awokengazebo.com/software/lfi/">awokengazebo.com/software/lfi/</a></td>
<td>BSD 4-clause &quot;Original&quot; or &quot;Old&quot; License</td>
</tr>
<tr>
<td>beefproject</td>
<td>beef-0.4.3.1</td>
<td><a href="http://beefproject.com">beefproject.com</a></td>
<td>Apache License 2.0</td>
</tr>
<tr>
<td>bitswitcher</td>
<td>0.2.0, 0.3.0, 0.3.3</td>
<td><a href="http://sourceforge.net/projects/bitswitcher">sourceforge.net/projects/bitswitcher</a></td>
<td>GNU General Public License v2.0 or later</td>
</tr>
<tr>
<td>buildroot-kindle</td>
<td>master-20130206</td>
<td><a href="https://github.com/twobob/buildroot-kindle">github.com/twobob/buildroot-kindle</a></td>
<td>GNU General Public License v2.0 or later</td>
</tr>
<tr>
<td>Component name</td>
<td>Version</td>
<td>Home Page</td>
<td>License</td>
</tr>
<tr>
<td>-----------------------</td>
<td>----------------------------------</td>
<td>------------------------------------------------</td>
<td>----------------------------------------------</td>
</tr>
<tr>
<td>busybox</td>
<td>1.10.0, 1.12.0, 1.2.0, 1.4.0,</td>
<td><a href="https://github.com/mirror/busybox">https://github.com/mirror/busybox</a></td>
<td>GNU General Public License v2.0 only</td>
</tr>
<tr>
<td></td>
<td>1.5.0, 1.8.0, 1.11.0, 1.13.0,</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.14.1, 1.16.0, 1.17.1, 1.17.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2, 1.18.0, 1.18.2, 1.19.0,</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.19.1, 1.19.4, 1.20.2, 1.21.0,</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.24.0, 1.29.0, 1.3.0, 1.7.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>busybox</td>
<td>1.14.0, 1.15.0, 1.17.0, 1.19.2,</td>
<td><a href="https://github.com/mirror/busybox">https://github.com/mirror/busybox</a></td>
<td>GNU General Public License v2.0 or later</td>
</tr>
<tr>
<td></td>
<td>1.19.3, 1.20.0, 1.20.1,</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.28.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>catboost/catboost</td>
<td>v0.2</td>
<td><a href="https://catboost.ai">https://catboost.ai</a></td>
<td>Apache License 2.0</td>
</tr>
<tr>
<td>curl</td>
<td>7.16.0</td>
<td><a href="https://curl.se/">https://curl.se/</a></td>
<td>curl License</td>
</tr>
<tr>
<td>decorator-ko</td>
<td>26, 28</td>
<td><a href="http://jinself.tistory.com/372">http://jinself.tistory.com/372</a></td>
<td>Public Domain</td>
</tr>
<tr>
<td>file</td>
<td>5.22</td>
<td><a href="http://www.darwinsys.com/file/">http://www.darwinsys.com/file/</a></td>
<td>Fine Free File Command License</td>
</tr>
<tr>
<td>fluxcapacitor</td>
<td>0</td>
<td><a href="https://github.com/majek/fluxcapacitor">https://github.com/majek/fluxcapacitor</a></td>
<td>MIT License</td>
</tr>
<tr>
<td>fvpats : fvpat Webdev Server</td>
<td>fvpats v0.1.4</td>
<td><a href="http://sourceforge.net/projects/fvpats">http://sourceforge.net/projects/fvpats</a></td>
<td>Apache License 2.0</td>
</tr>
<tr>
<td>generator-minxing</td>
<td>1.0.2</td>
<td><a href="https://github.com/yeoman/generator-minxing#readme">https://github.com/yeoman/generator-minxing#readme</a></td>
<td>Apache License 2.0</td>
</tr>
<tr>
<td>geonkick</td>
<td>2.3.6</td>
<td><a href="https://github.com/iurie-sw/geonkick">https://github.com/iurie-sw/geonkick</a></td>
<td>GNU General Public License v3.0 or later</td>
</tr>
<tr>
<td>hostap-ct</td>
<td>If-5.1.7, If-5.3.3, If-5.3.3b,</td>
<td><a href="https://github.com/greearb/hostap-ct">https://github.com/greearb/hostap-ct</a></td>
<td>BSD 3-clause “New” or “Revised” License</td>
</tr>
<tr>
<td></td>
<td>If-5.3.4, If-5.3.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>hostapd</td>
<td>hostap_0_5_2, hostap_0_5_3,</td>
<td><a href="http://w1.fi/hostapd/">http://w1.fi/hostapd/</a></td>
<td>GNU General Public License v2.0 or later</td>
</tr>
<tr>
<td></td>
<td>hostap_0_5_6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>howl</td>
<td>0.9.4, 0.9.6, 0.9.7, 0.9.9,</td>
<td><a href="https://howl.io">https://howl.io</a></td>
<td>BSD 3-clause “New” or “Revised” License</td>
</tr>
<tr>
<td></td>
<td>1.0.0, 0.9.3, 0.9.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>illumos-joyent</td>
<td>20121101</td>
<td><a href="http://www.illumos.org/projects/illumos-gate">http://www.illumos.org/projects/illumos-gate</a></td>
<td>Common Development and Distribution License 1.0</td>
</tr>
<tr>
<td>krb5/krb5</td>
<td>1.0-alpha3, 1.0-beta2, 1.0-beta5</td>
<td><a href="https://github.com/krb5/krb5">https://github.com/krb5/krb5</a></td>
<td>Krb5-MIT License</td>
</tr>
<tr>
<td>libevent - an event notification library</td>
<td>0.1, 1.0d, 1.0e, 1.4.1-beta</td>
<td><a href="http://libevent.org/">http://libevent.org/</a></td>
<td>BSD 3-clause “New” or “Revised” License</td>
</tr>
<tr>
<td>libexpat</td>
<td>1.95.0, 1.95.1, 1.95.2, 2.0.0,</td>
<td><a href="http://www.libexpat.org/">http://www.libexpat.org/</a></td>
<td>Expat License</td>
</tr>
<tr>
<td></td>
<td>v19991013</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Component name</td>
<td>Version</td>
<td>Home Page</td>
<td>License</td>
</tr>
<tr>
<td>------------------</td>
<td>-----------------</td>
<td>------------------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>libexpat</td>
<td>V19991013</td>
<td><a href="http://www.libexpat.org/">http://www.libexpat.org/</a></td>
<td>Mozilla Public License 1.1</td>
</tr>
<tr>
<td>linux-yocto-dev</td>
<td>v2.6.12</td>
<td><a href="http://git.yoctoproject.org/cgit/">http://git.yoctoproject.org/cgit/</a></td>
<td>GNU General Public License v2.0 with Linux Syscall Note</td>
</tr>
<tr>
<td></td>
<td></td>
<td>cgit.cgi/linux-yocto-dev/</td>
<td></td>
</tr>
<tr>
<td>littlekernel-m900-ecclair</td>
<td>master-20110326</td>
<td><a href="http://github.com/LouZiffer/">http://github.com/LouZiffer/</a></td>
<td>GNU General Public License v2.0 only</td>
</tr>
<tr>
<td></td>
<td></td>
<td>littlekernel-m900-ecclair</td>
<td></td>
</tr>
<tr>
<td>lmdb</td>
<td>0.9.18</td>
<td><a href="http://symas.com/mdb/">http://symas.com/mdb/</a></td>
<td>Open LDAP Public License</td>
</tr>
<tr>
<td>math-linux</td>
<td>0.0.1</td>
<td><a href="http://sourceforge.net/projects/math-linux">http://sourceforge.net/projects/math-linux</a></td>
<td>Gnu General Public License v3.0 or later</td>
</tr>
<tr>
<td>mod_dup</td>
<td>2.5.0</td>
<td><a href="http://github.com/Orange-OpenSource/mod_dup/">http://github.com/Orange-OpenSource/mod_dup/</a></td>
<td>Apache License 2.0</td>
</tr>
<tr>
<td>ngx_pagespeed</td>
<td>1.9.32.4-dbg-ssl-crash</td>
<td><a href="https://github.com/pagespeed/ngx_pagespeed">https://github.com/pagespeed/ngx_pagespeed</a></td>
<td>Apache License 2.0</td>
</tr>
<tr>
<td>nss_ldap</td>
<td>253</td>
<td><a href="https://github.com/PADL/nss_ldap">https://github.com/PADL/nss_ldap</a></td>
<td>GNU Library General Public License v2 or later</td>
</tr>
<tr>
<td>opensm</td>
<td>3.3.17</td>
<td><a href="http://www.openfabrics.org/">http://www.openfabrics.org/</a></td>
<td>BSD 2-clause “Simplified” License</td>
</tr>
<tr>
<td>pGina</td>
<td>Plugin Bundle 05-11-2006</td>
<td><a href="http://pgina.org/">http://pgina.org/</a></td>
<td>MIT License</td>
</tr>
<tr>
<td>pam_radius</td>
<td>release_2_0_0</td>
<td><a href="http://freeradius.org/pam_radius_auth/">http://freeradius.org/pam_radius_auth/</a></td>
<td>GNU General Public License v2.0 only</td>
</tr>
<tr>
<td>protovis</td>
<td>3.3.1</td>
<td><a href="http://mbostock.github.io/protovis/">http://mbostock.github.io/protovis/</a></td>
<td>BSD 3-clause “New” or “Revised” License</td>
</tr>
<tr>
<td>root-project</td>
<td>5-13-04e</td>
<td><a href="https://root.cern">https://root.cern</a></td>
<td>(GNU Lesser General Public License v2.1 or later AND MIT License AND GNU General Public License v2.0 or later)</td>
</tr>
<tr>
<td>rsyslog</td>
<td>sysklogd-141-import</td>
<td><a href="https://www.rsyslog.com/">https://www.rsyslog.com/</a></td>
<td>GNU General Public License v2.0 or later</td>
</tr>
<tr>
<td>rtems-libbsd</td>
<td>5.1</td>
<td><a href="http://git.rtems.org/rtems-libbsd.git/">http://git.rtems.org/rtems-libbsd.git/</a></td>
<td>Apache License 2.0</td>
</tr>
<tr>
<td>rtl8186 - toolchain</td>
<td>0.5.5_src</td>
<td><a href="http://rtl8186.sourceforge.net">http://rtl8186.sourceforge.net</a></td>
<td>GNU General Public License v2.0 or later</td>
</tr>
<tr>
<td>snake-os</td>
<td>0.9</td>
<td><a href="http://code.google.com/p/snake-os/">http://code.google.com/p/snake-os/</a></td>
<td>GNU General Public License v2.0 or later</td>
</tr>
<tr>
<td>Component name</td>
<td>Version</td>
<td>Home Page</td>
<td>License</td>
</tr>
<tr>
<td>----------------</td>
<td>---------</td>
<td>-----------</td>
<td>---------</td>
</tr>
<tr>
<td>ssmtp</td>
<td>2.61</td>
<td><a href="http://packages.qa.debian.org/s/sssmtp.html">http://packages.qa.debian.org/s/sssmtp.html</a></td>
<td>GNU General Public License v2.0 or later</td>
</tr>
<tr>
<td>util-linux</td>
<td>2.11q, 2.11w, 2.12a, 2.13-pre1</td>
<td><a href="https://en.wikipedia.org/wiki/Util-linux">https://en.wikipedia.org/wiki/Util-linux</a></td>
<td>GNU General Public License v2.0 or later</td>
</tr>
<tr>
<td>videolan/vlc</td>
<td>0.5.0</td>
<td><a href="https://github.com/videolan/vlc">https://github.com/videolan/vlc</a></td>
<td>(GNU Lesser General Public License v2.1 or later AND GNU General Public License v2.0 or later)</td>
</tr>
<tr>
<td>wakame-vdc</td>
<td>v13.06.0</td>
<td><a href="http://wakame.axsh.jp/">http://wakame.axsh.jp/</a></td>
<td>Unknown License</td>
</tr>
<tr>
<td>wpa_supplicant - IEEE 802.1X, WPA, WPA2, RSN, IEEE 802.11i</td>
<td>0.5.0, 0.5.3, 0.5.5, 0.5.6, 0.5.8, 0.6.0, 0.6.10, 0.6.2, 0.6.3, 0.6.4, 0.6.8, 0.7.0, 0.7.1, 0.7.2, 0.7.3, 1, 2, 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.7+git20190108+11ce7a1, 2.7-git20180504+60a5737, 2.7-git20180606+b915f2c, 2.7-git20180706+420b5dd</td>
<td><a href="http://w1.fi/wpa_supplicant/">http://w1.fi/wpa_supplicant/</a></td>
<td>BSD 3-clause “New” or “Revised” License</td>
</tr>
<tr>
<td>xorp.ct</td>
<td>1.5, xorp-1-7</td>
<td><a href="http://www.candelatech.com/xorp.ct">http://www.candelatech.com/xorp.ct</a></td>
<td>MIT License</td>
</tr>
<tr>
<td>zeroconf</td>
<td>0.9</td>
<td><a href="https://files.pythonhosted.org/packages/20/d7/418ff6c684ace0f5855ec56c66cfa99ec50443c41693b91e9abccfa096c/zeroconf-0.20.0.tar.gz">https://files.pythonhosted.org/packages/20/d7/418ff6c684ace0f5855ec56c66cfa99ec50443c41693b91e9abccfa096c/zeroconf-0.20.0.tar.gz</a></td>
<td>GNU General Public License v2.0 or later</td>
</tr>
</tbody>
</table>
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. Neither NVIDIA Corporation nor any of its direct or indirect subsidiaries and affiliates (collectively: “NVIDIA”) make any 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 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.

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, the NVIDIA logo, and Mellanox are trademarks and/or registered trademarks of NVIDIA Corporation and/or