NVIDIA UFM Cable Validation Tool v1.7.1

Table of contents

About This Document

This document describes NVIDIA® Unified Fabric Manager (UFM®) Cable Validation tool, connectivity and configuration options.

Technical Support

Customers who purchased NVIDIA products directly from NVIDIA are invited to contact us through the following methods:

- E-mail: enterprisesupport@nvidia.com
- Enterprise Support page: https://www.nvidia.com/en-us/support/enterprise

Customers who purchased NVIDIA M-1 Global Support Services, please see your contract for details regarding technical support.

Customers who purchased NVIDIA products through an NVIDIA-approved reseller should first seek assistance through their reseller.

Document Revision History

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

Release Notes

Changes and New Features in This Release

Feature	Description
Manage Agents from the UI	Added UI support for managing agents, refer to Agent Status Report and Services.
Download Topology Files from the UI	Added UI support for downloading topology files, refer to Services.
Control Using DNS to Resolve Device Names from the UI	Added UI support to enable or disable DNS when loading the topology file, refer to Load Topology.
Resource Filter	Filters data based on the specified resources, refer to Resource Filter.
Credential Profile	Support Credential Profile in Unified P2P topology file.

Unsupported Functionalities/Features

In the upcoming versions of the Cable Validation Tool, the Legacy Ethernet and InfiniBand P2P file formats will be deprecated. Moving forward, only the Unified Topology will be supported. For more information, refer to P2P File.

Installation Notes

Supported Fabric Types

The UFM Cable Validation Tool is compatible with the following fabric types:

Fabric Type	Description
InfiniBand	Supported on MLNX-OS Switches with NDR and HDR port speeds

Fabric Type	Description
XDR	Supported on NVOS Q3400-RA Switches with XDR port speed (Does not support link flap detection)
Ethernet	Supported on Cumulus Switches and Ethernet Hosts
NVLink	• Supported on NVOS Switches and compute nodes of GB200/GB300 rack types: 36x2, 36x1, and 72x1. • Supported on internal NVLink links in the rack • Hosts with external links as Ethernet or IB • The external InfiniBand links will report BER status, but neighboring information will need to be monitored from the neighboring switch side only.

Supported Devices

Type	Model
XDR Switches	• Q3400_RA
NDR Switches	• MQM9700
HDR Switches	• MQM8700
Ethernet Switches	• SN5600 • SN5610
NVLink Switches	• N5200_LD • N5201_LD • N5110_LD • N5112_LD • N5113_LD
Ethernet GPU Servers	• SYS-421GE-TNHR2-LC-TW008 (H100) • PowerEdge XE9680
NVLink GPU Servers	• GB200 NVL • DGX GB200 Compute Tray • Dell Server 9712a

ⓘ **Note**

Devices not included in this list will be supported by CVT but may not be displayed correctly in the rack view.

Pre-Requisites for Server/Hosts:

- IB servers should have `mft` and `ofed` packages installed.
- GB200 compute nodes should have `nvidia-container-toolkit` and `nvidia-smi` packages installed.

Bug Fixes in This Release

Ref #	Bug Fix
4568182	**Description**: /root/.npm/_cacache exists in plugin container
	Keywords: npm security
	Discovered in Release: 1.6.0
4583876	**Description**: Remove anomaly "Module Rx/Tx power was out of range but is good now (latched flag)
	Keywords: anomaly latched power
	Discovered in Release: 1.6.0
4510330	**Description**: In Login page no error message appears if the password or the username was incorrect
	Keywords: login
	Discovered in Release: 1.6.0
4575762	**Description**: In and Out Bytes on interfaces not being populated in CVT
	Keywords: rx tx bytes interface
	Discovered in Release: 1.6.0

Ref #	Bug Fix
4570374	**Description**: Option of having unmanaged nvswitches not working
	Keywords: unmanaged nvswitch
	Discovered in Release: 1.6.0
4372474	**Description**: Module read issue in sample error on many ports
	Keywords: transceiver module
	Discovered in Release: 1.5.0
4636690	**Description**: CVT plugin is not working on a non-root env
	Keywords: non-root env
	Discovered in Release: 1.6.0
4601395	**Description**: f nm port support for gb200 NVL72x1 switches
	Keywords: f nm port
	Discovered in Release: 1.6.0
4575765	**Description**: setting CV_DOT_IN_HOSTNAME on cvt_env.conf does not work as expected
	Keywords: hostname, CV_DOT_IN_HOSTNAME
	Discovered in Release: 1.6.0

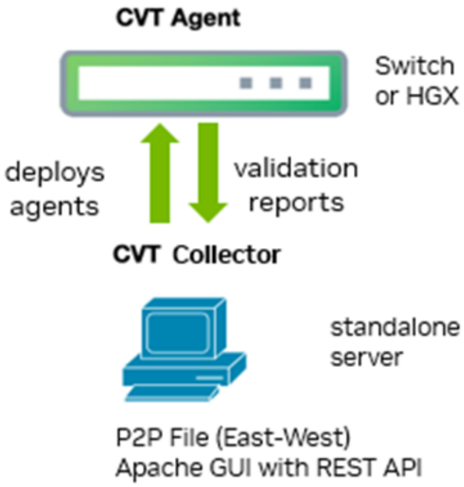
Known Issues in This Release

Reference Number	Issue
4707987	**Description**: the raw_ber/effective_ber for tests/ circuits should be in scientific format
	Keywords: raw ber; effective ber scientific format
	Discovered in Release: 1.7.0
4707986	**Description**: stop cvt collector will not stop tests if its running
	Keywords: test
	Discovered in Release: 1.7.0

Introduction

The **Cable Validation Tool (CVT)** is designed to ensure the accuracy and quality of network cluster wiring. Its primary purpose is to validate the connectivity of physical links within the cluster and verify high-quality communication between the network components. By maintaining the integrity of these connections.

The **Collector**, also referred to as the "bring-up server," serves as the central component of the system and operates as a Docker container. It can be deployed on any machine connected to the management network of the switches. This deployment enables seamless communication with the switches. For large-scale systems, the Collector relies on dedicated **agents** installed on each switch. These agents are responsible for verifying the connections between the switches.



Collector Responsibilities

The Collector performs the following critical tasks:

- **Deployment and Execution:** It is installed and executed on a server with network access to all nodes requiring validation.
- **Topology Validation:** Reads the **Topology Files** (P2P, topo or dot), which serves as the authoritative source for validating the physical link connections in the fabric topology.
- **Agent Management:**

    - Deploys agents on all nodes and uninstalls them
    - Monitors agents' health
    - Supports external (unmanaged) agent deployments, with the Collector only monitoring their health
- **Data Collection and Processing:** Collects and processes agents' reports from all validated nodes
- **User Interface and Reporting:**
    - Displays validation results through a web page, along with recommended remediation steps.
    - Provides data visualization options, including aggregation, sorting, and filtering, and supports downloading reports in CSV format. REST APIs are also available for integration with other systems.

Agent Responsibilities

Agents are installed on all switches and servers within the cluster. Their key functions include:

- **deployed on all Switches and Servers**
- **Real-Time Monitoring:**
    - Monitor node and link statuses every 10 seconds. Agents detect changes in link states and, when a change occurs, send an updated status report to the Collector.
    - If no changes are detected, agents send a periodic status report every 10 minutes.
    - Amberfile collection takes place upon state changes, which can take 40-50s.
- **Event-Driven Reporting:**
    - Upon receiving a "start_validation" message from the Collector, agents initiate status reporting.
    - Reports are triggered in two scenarios:
        - When a link status change is detected (ad-hoc report).

        - Every 10 minutes as part of routine reporting.

Further details on the Collector and Cable Agents, including their operational workflows, will be discussed in the subsequent chapters.

Symptoms and Remediation

The CVT tool supports the following symptoms/issues:

- **Validate *Physical* Connections (Cable, End points) / Miswiring**
    - **Wrong-neighbor** - the cable connects to a different device than Topo file (P2P, topo or dot) dictates
    - **Wrong-port** - the cable connects to the expected device but on the wrong port
    - **Unknown-neighbor** - the cable connects to a device not mentioned in the Topo file (P2P, topo or dot) or LLDP is not enabled/failing
    - **Extra Cable** - a cable was found to be connected but not part of the loaded Topo file (P2P, topo or dot)
    - **Media Unplugged** - a cable is missing in the port. A transceiver is not present in the port if optical cable is used.
    - **NIC name-mismatch** - The interface name of hosts in the ptp file does not match any actual interface names on the host
- **Validate *Layer1 Link* Integrity (Bit Error Rate, Lane powers, Temperature)**
- **Flapping link** - the link state has transitioned up->down->up on its own or due to external actions.
- **Link Down, No Signal** - fiber is not connected or broken
- **ErrDisable-Rx** - interface down events due to the Server NIC firmware bug issues (RX Disable)
- **Err-Disable-Flap** - Link Protection feature (5 flaps/10 sec.) due to excessive flaps
- **Anomalous-Port** - out-of-range parameters such as transceiver lane signal strength or transceiver temperature
- **Underperforming (BER) Bit-Error-Rate**

    - Effective BER errors should be 0 during the first 125 mins of the link being up
    - Raw BER should be ⩽ 1e-6
    - Effective BER should be ⩽ 1.5E-254 for ⩽ 6hrs measurements and ⩽ 1E-15 for ⩾ 6hrs measurements
- **Triage *Non-cable issues (Provisioning, CVT issues) requiring Escalations***
    - **AdminDown** - spectrum switch port is administratively shutdown
    - **Negotiation Fail** - detects interface issue due to speed and duplex mismatch between devices
    - **No-report** - agent communication is working but report not received
    - **unreachable-device** - agent not installed, not running, not reachable (e.g. port 8251 not open in switch configuration)

Syndrome	**Description**	**non-admin users**	**Supported Fabric**
Negotiation Fail	Negotiation of speed, fec or config issue	YES	IB, ETH, XDR
AdminDown	Link is disabled administratively	Yes	IB, ETH, XDR
Wrong-neighbor	Port is connected to the wrong peer node	YES	IB, ETH, XDR
Wrong-port	Port is connected to the wrong port in the correct peer node	YES	IB, ETH, XDR
Extra-cable	port is connected but neighbor is not in the P2P topology	YES	IB, ETH, XDR
Flapping-link	On switches: Carrier transitions are monitored every 10 seconds. If it increments by more than 2 in 125 sec interval, a link flap alarm is raised	YES	IB, ETH, XDR
Underperforming-link	High BER counters	YES	IB, ETH, XDR

Syndrome	**Description**	**non-admin users**	**Supported Fabric**
Anomalous-port (Signal, Temperature)	Some counters are not in range	YES	IB, ETH, XDR
Unreachable-device	Cannot ping it and/or Agent not deployed orAgent communication is failing	NO	IB, ETH, XDR
Media Unplugged	Port is down and no cable is plugged in or transceiver is not plugged in if its a optical cable	YES	IB, ETH, XDR
NIC Name-mismatch	Interface name of hosts in the ptp file does not match any actual interface names on the host	YES	ETH
Unknown-neighbor	Port is up, however no peer info found one known instance is when the far end is not reachable	NO	IB, ETH, XDR
Link Down, No signal	Port is down, **while transceivers are plugged in**	YES	IB, ETH, XDR
ErrDisable – Flap, Proto Down	Cumulus switch Port locally disabled by Link Protection (⩾5 flaps/10s), defensive mechanism enabled by default.	YES	IB, ETH, XDR
ErrDisable -Rx	Interface down events due to the Server NIC firmware bug issues (RX Disable)	YES	IB, ETH, XDR
no report	Node is reachable but no agent report was received yet	NO	IB, ETH, XDR

Features Supported in Different Fabrics

Feature	**Description**	**Supported Fabric**
Circuit View	Shows the links being monitored	IB, ETH, XDR, GB200
Port Status	Shows the port is up or down	IB, ETH, XDR, GB200

Feature	**Description**	**Supported Fabric**
Link Syndrome	Shows the cable/port issues for the link	IB, ETH, XDR, GB200
BER Stats	Shows the eff BER, raw BER, and grading based on BER	IB, ETH, XDR, GB200
Report Anomalies	Shows the anomalies detected	IB, ETH, XDR, GB200
Flapping Status	Shows the advanced flapping stats	IB, ETH
Rack View	Shows the switches and hosts on the rack	IB, ETH, XDR, GB200
System Admin	Allows to start the bringup service, load topology, set credentials and start validation. Shows overview of the validation session and reporting status of agents. Allows to manage GUI users and displays collector resource utilization.	IB, ETH, XDR, GB200
Golden BER Test	Creates a test to monitor Bit Error Rates (BER) and analyze the interface counters	IB, ETH
Amber Collection Test	Allows to collect amber files on demand	IB, ETH, XDR, GB200
Advanced Flapping Test	Creates a flapping test to analyze the metrics that could lead to a flapping event	IB, ETH

Topology Files

Topology files are the primary reference for validating physical link connections within a fabric topology. These files must be provided by the user, as the validation process cannot start without them.

The tool supports the following types of topology files:

- P2P File
- Topo File
- Dot File
- Optional Files

P2P File

Unified Topology Format

As of CVT v1.5.0. a n ew Unified Topology file format is introduced. This format aims to manage various cluster types, including InfiniBand (HDR/NDR/XDR) , Ethernet, and NVLink, with ease using an Excel-based structure for organized data representation. The proposed format (Excel file) with multiple sheets encapsulates the topology of modern data center requirements and acts as a one-stop solution. For downloading the Excel template, refer to Unified Topology Template.xlsx.

The Unified Topology file format consists of 4 sheets:

1. Nodes
2. Links
3. DC Floor Layout (optional sheet if not managing GB200/300 racks)
4. Server Profile (optional sheet if not managing Servers/Hosts)

Nodes

The Nodes sheet will focus on providing comprehensive information about the nodes within the cluster. The primary data points will include:

FabricID	Rack	Unit	TrayIndex	NodeName	NodeType	NodeOS	NodeModel	ServerPr

Column	Value Type	Description	Mandatory
NodeName	String	The identifier for the node within the network.	☐
NodeType	[Switch|Host]	The classification of the node (e.g., host, switch).	☐
NodeModel	String	The specific model of the node. **Mandatory for GB200/300**	☐
NodeOS	String	The OS running on the node. Supported OS are: mlnx-os, cumulus, nvos, and linux for hosts	☐
Rack	String	The physical rack where the node is located.	☐
Unit	Integer	The specific unit within the rack.	☐
TrayIndex	Integer	The tray index within the rack. Relevant and **mandatory for GB200/300**	☐
CoolingType	[Air|Liquid]	The node cooling type (e.g., Air, Liquid). For future use (not supported now)	☐
FabricID	String	The Site/Cluster/Fabric name/ID In case the topology includes multiple sites	☐
ServerProfile	String	Hosts are assigned server profiles to provide mapping between different interface naming conventions. It is a custom name you can create which will be referenced in the Server Profile Sheet.	☐
Managed	[yes|no]	Control if agent should be installed on this node. Set to 'yes' by default.	☐
CredentialProfile	String	A custom name assigned to a set of nodes with same credentials which is different from the default.	☐

Note

Notes:

- If Rack/Unit information is missing, some features—specifically the Rack View and filtering capabilities—will not function in the CVT interface.
- IB hosts need to be added to the nodes sheet, so that their corresponding links to switches can be detected. However, agent installation on them is not supported, hence their 'Managed' column value should be 'no'.
- Column names are case-insensitive, and the order of columns does not affect functionality.
- The user can dismiss optional columns as described in the Mandatory column.

Links

The Links sheet will detail the connectivity information across the nodes within the cluster. It will consist of the following columns:

A-Node	A-Port	Z-Node	Z-Port	Protocol	Shuffle-ID	A-Connector	A-MPO-Connector	Z-Connector	Z-MPO Connec

Column	Value Type	Description	Mandatory
A-Node	String	The source node for the link. Must exist in the Nodes sheet	▯
A-Port	String/Integer	The port on the source node.	▯
Z-Node	String	The destination node for the link. Must exist in the Nodes sheet	▯
Z-Port	String	The port on the destination node.	▯
Protocol	(ib/ethernet/nvlink)	The protocol used for the connection	▯

Column	Value Type	Description	Mandatory
Shuffle-ID	String/Integer	Shuffle Cable ID	▯
A-Connector	String/Integer	A side Connector of the shuffle cable	▯
A-MPO-Connector	String/Integer	A side MPO Connector of the shuffle cable	▯
Z-Connector	String/Integer	Z side Connector of the shuffle cable	▯
Z-MPO-Connector	String/Integer	Z side MPO Connector of the shuffle cable	▯
A-Module-PN	String/Integer	A side module part number	▯
Z-Module-PN	String/Integer	Z side module part number	▯
Source-Ref	Integer	Line number from a original reference file from which this PTP is constructed	▯

In case of Host/Server ports, the A/Z-port would be the custom NIC name if available or the default NIC name. This value can be found in the output of `ip address show` command.

For IB hosts it is mandatory to have a server profile for this interface so that there is a mapping available for corresponding RDMA name of the port. RDMA name can be found from output of command: `sudo mst status -v`.

Note

Notes:

- Column names are case-insensitive, and the order of columns does not affect functionality.
- The user can dismiss Optional Columns as explained in the Mandatory column.

Internal Links

Internal links (NVLink within the GB200/300 racks) will not be part of the Links sheet. CVT detects any internal links based on the RackType and uses the predefined JSON representation of these links to build the links. This approach ensures seamless integration and efficient configuration within the GB200/300 racks.

DC Floor Layout

The data center floor layout sheet is an optional sheet that is useful to provide a layout of the data center. This includes information about:

- Data Halls: The various halls within the data center.
- Scalable Units: Units designed to be scalable for future expansions. Add a default SU name for all if you are not using scalable unit concepts.
- Racks: Detailed information about the racks within the data center. This information is mandatory for GB200/300 racks.
-
    - Rack: Rack Name
-
    - Rack Type: Type of the rack. Can be GB200_72x1, GB200_36x2, or GB200_36x1 for GB200 racks (similar convention is used for GB300) or any other general rack type like ServerRack, NetworkRack for other general racks.
-
    - Rack Group: Used for grouping the racks, especially in GB200 racks. It is a unique number for each GB200 rack system. A GB200 72x1 system would have a unique group number for itself. In a GB200 36x2 system, the two racks will belong to the same group_number. (Group number is just a made-up integer to facilitate the identification of rack systems correctly).

DataHall	ScalableUnit	Rack	RackType	Rack Group

Server Profile

The Server Profile sheet will be used to specify the interface configurations on Hosts, it does not have any significance for switches. The ServerProfile name mentioned in the Nodes sheet should have a corresponding entry here. Server Profile sheet will include below information:

Fabric ID	CustomNICName	NICOSName	PhysicalPort	RDMAName	PCIAddress

Column	Value Type	Description	Mandatory
FabricID	String	For future use (not supported now)	▯
CustomNICName	String	Custom name given to the NIC if it has been renamed from its default value. This value can be found listed in the output of the command \`ip address show\`	▯
NICOSName	String	Default name of the interface. It can be found in the output of the command \`ip address show\`. If NIC is renamed with a custom name, the default name can be found under the altname tag.	▯
PhysicalPort	String	The OSFP port/slot in IB networks	▯
RDMAName	String	RDMA name of IB ports like mlx5_0, etc. Mandatory for IB ports. Can be found in the output of the command \`sudo mst status -v\`	▯
PCIAddress	String	For future use (not supported now)	▯

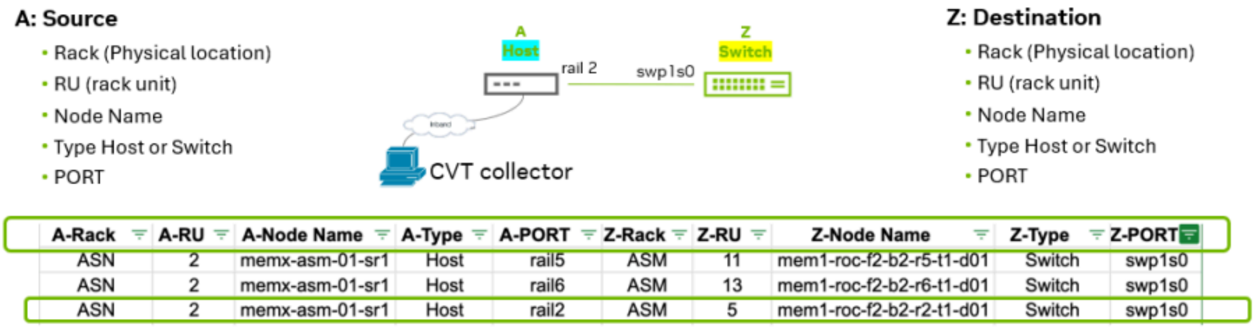
Note

DC Floor layout and Server Profile in Unified Topology are provided in the same Excel workbook and not passed as optional arguments.

Legacy P2P Format (will be deprecated)

The P2P file is an Excel file that details the physical link connections within the fabric. It may consist of multiple sheets, each containing the following columns:

- **A-Node Name**: Specify the name of the node on the "A" side of the connection.
- **A-Type**: Indicate the role of the "A" side node, either "Host" or "Switch" (applicable for Ethernet).
- **A-Port**: Provide the port name for the "A" side node.
- **Z-Node Name**: Specify the name of the node on the "Z" side of the connection.
- **Z-Type**: Indicate the role of the "Z" side node, either "Host" or "Switch" (applicable for Ethernet).
- **Z-Port**: Provide the port name for the "Z" side node.



A: Source

- Rack (Physical location)
- RU (rack unit)
- Node Name
- Type Host or Switch
- PORT

Z: Destination

- Rack (Physical location)
- RU (rack unit)
- Node Name
- Type Host or Switch
- PORT

A-Rack	A-RU	A-Node Name	A-Type	A-PORT	Z-Rack	Z-RU	Z-Node Name	Z-Type	Z-PORT
ASN	2	memx-asm-01-sr1	Host	rail5	ASM	11	mem1-roc-f2-b2-r5-t1-d01	Switch	swp1s0
ASN	2	memx-asm-01-sr1	Host	rail6	ASM	13	mem1-roc-f2-b2-r6-t1-d01	Switch	swp1s0
ASN	2	memx-asm-01-sr1	Host	rail2	ASM	5	mem1-roc-f2-b2-r2-t1-d01	Switch	swp1s0

P2P Examples

InfiniBand Example:

A sample sheet for InfiniBand connections in a P2P file:

Rack	U	Name	HCA/Port	Rack	Name	Name	Port
PXH	28	swx-proton03	1	PXX	30	sw-hdr-proton01	1/3
316	24	swx-proton04	3	PXX	27	sw-hdr-proton01	1/4

- The designated port can be a single number or a split port (e.g., 1/2).
- Mapping for HCA ports:
    - 1 → mlx5_0 P1

    - 2 → mlx5_1 P1
    - And so on.

The HCA mapping could be customized by the user, for more details see HCA Mapping File

XDR Example:

A sample sheet for NVOS connections in a P2P file:

Rack	U	Name	HCA/Port	Rack	Name	Name	Port
316	22	clx-abc-073	1	R113	22	bm-abc-t4	sw1p1
316	24	clx-abc-074	1	R113	22	bm-abc-t5	sw1p2

- Designated ports are represented as `sw<port_number>p<split_number>` for switches.
- Mapping for HCA ports: Same as InfiniBand (see above).

Ethernet Example:

A-Rack	A-U	A-Node Name	A-Type	A-Port	Z-Rack	Z-RU	Z-Node Name	Z-Type	Z-Port
ASN	2	memx-asm-01-sr1	Host	rail5	ASM	11	mem1-roc-f2-b2-r5-t1-d01	Switch	swp1s0
ASN	2	memx-asm-01-sr1	Host	rail6	ASM	13	mem1-roc-f2-b2-r6-t1-d01	Switch	swp1s1

Designated ports are represented as `swp<port_number>s<split_number>` for switches.

Please note that in all fabrics the tool relies on the header names to extract the information it needs. So user must have these names exactly as they appear above. If you make a syntax error then it will fail.

Legend Sheet

it is mandatory for the PTP file to incorporate a "Legend" sheet, which contains vital details regarding switch and host patterns. The below is an example:

Example:

Name	Model	Switch/HCA	Speed	Rate
`c-csi-mqm*`	MQM9700	Switch	4x 100G	NDR
`c-csi-0*`	HCA_2	HCA	4x 100G	NDR

Topo File

This file is specifically supported for InfiniBand and NVOS fabrics. It is generated using the `ibdiagnet` tool with a specific command.

The following is example for topo file:

```
MQM8700 sw-ufm-hdr01
   P2 -4x-50G-> HCA_mlx5_3 swx-ufm3-04 mlx5_0/P1
   P3 -4x-50G-> HCA_mlx5_3 swx-ufm3-04 mlx5_1/P1
   P40 -4x-50G-> MQM9700 sw-ufm-ndr05 P62
   P5 -4x-50G-> HCA_mlx5_3 swx-ufm3-05 mlx5_2/P1
MQM9700 sw-ufm-ndr05
   P61 -4x-50G-> MQM8700 sw-ufm-hdr01 P39
   P62 -4x-50G-> MQM8700 sw-ufm-hdr01 P40
```

Dot File

This file is specifically supported for Ethernet fabric

```
graph network {
```

```
"mose-gbnvl-l1" [ label="mose-gbnvl-l1" role="leaf" os="cumulus"]
"mose-gbnvl-l2" [ label="mose-gbnvl-l2" role="leaf" os="cumulus" ]
"mose-gbnvl-s1" [ label="mose-gbnvl-s1" role="spine" os="cumulus" ]
"mose-gbnvl-s2" [ label="mose-gbnvl-s2" role="spine" os="cumulus"]
"mose-gbnvl-c1" [ label="mose-gbnvl-c1" role="super_spine" os="cumulus"]
"mose-gbnvl-c2" [ label="mose-gbnvl-c2" role="super_spine" os="cumulus"]

"mose-gbnvl-l1":"swp1s0" -- "gb-nvl-yok-04-compute01":"enp3s0np0"
"mose-gbnvl-l1":"swp1s1" -- "gb-nvl-yok-04-compute02":"enp3s0np0"
"mose-gbnvl-l1":"swp2s0" -- "gb-nvl-yok-04-compute03":"enp3s0np0"
"mose-gbnvl-l1":"swp2s1" -- "gb-nvl-yok-04-compute04":"enp3s0np0"
"mose-gbnvl-l1":"swp3s0" -- "gb-nvl-yok-04-compute05":"enp3s0np0"
"mose-gbnvl-l1":"swp3s1" -- "gb-nvl-yok-04-compute06":"enp3s0np0"
"mose-gbnvl-l1":"swp4s0" -- "gb-nvl-yok-04-compute07":"enp3s0np0"
"mose-gbnvl-l1":"swp4s1" -- "gb-nvl-yok-04-compute08":"enp3s0np0"
"mose-gbnvl-l1":"swp5s0" -- "gb-nvl-yok-04-compute09":"enp3s0np0"
"mose-gbnvl-l1":"swp5s1" -- "gb-nvl-yok-03-compute01":"enp3s0np0"
}
```

Optional Files

The tool also supports optional files that helps with customizing the topology file.

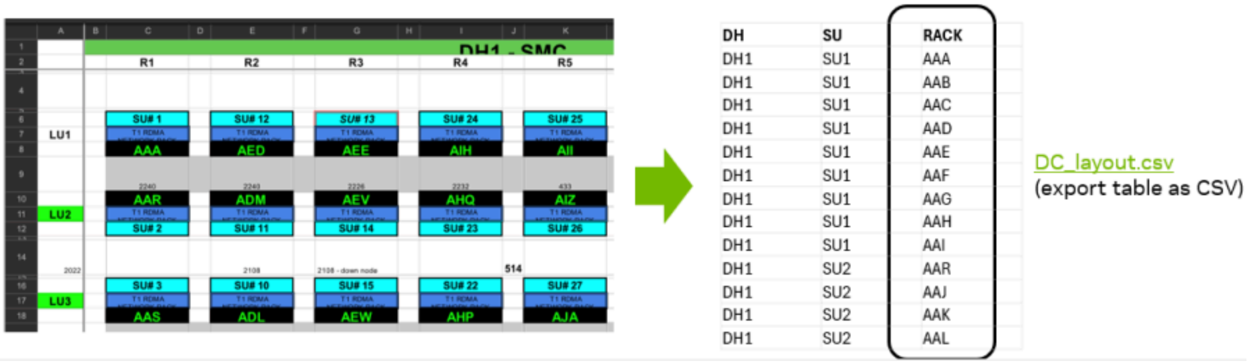
DC Floor Layout File

Supported for **Ethernet** fabrics, this CSV file allows users to aggregate CVT results based on their Data Center (DC) layout hierarchy, such as:

- **Data Hall # or Pod #**
- **Scalable Unit #**
- **Row #**
- **Line Unit #**

It defines the mapping of **racks** to a set of desired layout categories (column headers).

Example: below racks are mapped to SU's, LU's, R's, DH



DH	SU	RACK
DH1	SU1	AAA
DH1	SU1	AAB
DH1	SU1	AAC
DH1	SU1	AAD
DH1	SU1	AAE
DH1	SU1	AAF
DH1	SU1	AAG
DH1	SU1	AAH
DH1	SU1	AAI
DH1	SU2	AAR
DH1	SU2	AAJ
DH1	SU2	AAK
DH1	SU2	AAL

DC_layout.csv
(export table as CSV)

HCA Mapping File

Supported for **InfiniBand** and **NVOS** fabrics, this CSV file defines the relationship between port numbers and HCA names in the following format:

port,hca-name

- `port`: Interface number on the host
- `hca-name`: RDMA interface name, which could be mlx5_x, HCA-X, ibp*, or any other naming convention.

By default, if the user didn't provide the `hca_mapping` the tool will use the following assumption:

```
port,hca-name
1, mlx5_0
2, mlx5_1
3, mlx5_2
4, mlx5_3
5, mlx5_4
6, mlx5_5
7, mlx5_6
8, mlx5_7
```

The below is an example of the hca_mapping file:

```
port,hca-name
1,mlx5_1
2,mlx5_0
3,mlx5_2
4,mlx5_4
```

Collector

The collector is the main module that should be deployed and run on a host with management network access.

- Deploying the Module
- Bringup CLI
- Update Certificate
- CVT as a Service
- CVT Configuration
- High Availability (HA) Mode Support for CVT Plugin
- Agent Deployment and SSH Configuration
- Cluster Sizing Guide

Deploying the Module

Server Resource Requirements per Cluster Size

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

Fabric Size	CPU Requirements*	Memory Requirements	Disk Space Requirements
Above 10000 nodes	Contact NVIDIA Support		

The Cable Validation tool can be deployed in two methods:

- Deploying the Module as Standalone
- Deploying the Module as a UFM Enterprise Plugin

Deploying the Module as Standalone

Deploy the `cables_bringup` container on a host as described below:

1. `docker load -i <image_path>/cables_bringup_<version>.tar.gz`
2. `docker run --name cables_bringup -itd --network=host cables_bringup`
3. `docker exec -it cables_bringup /bin/bash`

Setting Docker Environment

There are three ways to set environment variables to help customize some of the settings on CVT.

1. Set the values in CVT environment variable configuration file [Least Priority]
2. Setting Environment variables when starting the docker container
3. Exporting environment variables manually inside the container [Highest Priority]

Environment Variable Configuration File

To enhance flexibility and usability, CVT supports environment variable management through a dedicated file. All variables listed below can be set in the configuration file, which includes default values for easy customization.

If an environment variable is defined in both the Docker environment and the configuration file, the Docker environment value takes precedence.

Step 1: Configuration File

The `cvt_env.conf` file is installed with CVT and comes preloaded with default values.

You can modify this file to match your environment requirements.

Step 2: Updating Variables

To update an environment variable:

1. Edit the `<cable_bringup_root>/config/cvt_env.conf` file.
2. Save your changes.
3. Restart the CVT collector for the changes to take effect.

   This can be done in the following ways:

   1. supervisorctl restart cvt-service
   2. supervisorctl stop cvt-service bringupcli -k

ⓘ **Note**

Note: A Docker container restart is not required—only the CVT collector needs to be restarted.

```
[Version]
# DO NOT EDIT THIS SECTION
# Developer note: when adding/removing/changing a variable, you
must increment the version number.
# Version of the cvt_env.conf file
# This version is used to check if the cvt_env.conf file is
compatible with the current version of the CVT
```

```
# If the version is not compatible, the original cvt_env.conf file will be saved as cvt_env.conf.save
# and the new cvt_env.conf file will be created with the current version
# The new cvt_env.conf file will be used to start the CVT

CVT_ENV_VERSION=1.0.0

# Variable names are case-sensitive, and should be unique among sections.

# Network Configuration
[network]
# IP addresses used by the agents:
# if no Environment Variable is set, the IP address of the default interface will be used.
# if AGENTS_COLLECTOR_NAT_IP is set
#   - the agents (switch and host) will use this IP address
# otherwise
#   - if DEFAULT_AGENTS_INTERFACE_NAME is set, switch and host agents will use the IP address of the interface
#       specified by DEFAULT_AGENTS_INTERFACE_NAME
#   - if HOST_SPECIFIC_INTERFACE_NAME is set, host agents will use the IP address of the interface
#       specified by HOST_SPECIFIC_INTERFACE_NAME

# Collector External (NAT) IP address; define if there is a NAT between the collector and the agents
# this IP address is used by the agents to communicate the collector
# fetch images and send data/reports to the collector
# Leave empty if there is no NAT between the collector and the agents
AGENTS_COLLECTOR_NAT_IP=
# Interface name of the collector over which all the agents (switch and host) will communicate
# The IP address of this interface will be used by all agents (switch and host) to communicate to the
# collector to fetch images and send data/reports
DEFAULT_AGENTS_INTERFACE_NAME=

# Use the following variable if you want to use a different interface of the collector for the host agents
# Define if the interface to connect with hosts is different from the one used for switch agents
# specified by DEFAULT_AGENTS_INTERFACE_NAME
# The IP address of this interface will be used by the host agents to communicate to the
# collector to fetch images and send data/reports
# Leave empty if you want to use the same interface as DEFAULT_AGENTS_INTERFACE_NAME
HOST_SPECIFIC_INTERFACE_NAME=

# Agent Configuration (settings used by the agents themselves)
[agent]
# set `true` if the switch hostname contains a dot (other than the domain part)
CV_DOT_IN_HOSTNAME=
# Time after which a full report is forced to be published.
# Value to be provided in minutes. Default is 720 minutes (12 hours).
# Interval less than 10 mins is not supported.
FULL_REPORT_PUBLISH_INTERVAL_MINUTES=720
# Set to `true` to publish amber data on each agent iteration, regardless of changes.
# Default is `false` - amber is only published when there are changes or during forced full reports.
AMBER_PUBLISH_EACH_ITERATION=false
# Agent data collection interval in seconds. Default is 600 seconds (10 minutes).
# This controls how often the agent collects and processes
port/link data.
AGENT_COLLECT_INTERVAL=600

# Collector Configuration (settings used by the collector to
manage agents)
[collector]
# Max time to wait for an agent to become inactive in minutes
MAX_INACTIVE_INTERVAL=15
# Interval to check for new switches in minutes
CHECK_NEW_SWITCHES_INTERVAL=15
# Time to wait for an agent to become active after start
validation (minutes)
START_VALIDATION_TIMEOUT=5
# Time to wait for an agent to become inactive in minutes
WAIT_TIME_INACTIVE_AGENTS=1

# Worker Concurrency Settings
# Max number of workers to run in parallel for general operations
(validation, connectivity, DNS)
CVT_MAX_WORKERS=30
# Max number of workers for agent deployment (limited due to 384MB
image transfers)
CVT_DEPLOYMENT_MAX_WORKERS=30

# Timeout Settings
# Quick timeout for unreachable devices (seconds) - reduces wait
time for failed connections
CVT_QUICK_TIMEOUT=3
# Agent communication timeout (seconds) - timeout for individual
HTTP requests to agents
AGENT_COMM_TIMEOUT=30

# Batch Processing Settings
# Batching threshold - only use batching for deployments larger
than this (reduces overhead)
```

```
CVT_BATCHING_THRESHOLD=5000
# Batch size when batching is used (devices per batch)
CVT_BATCH_SIZE=1000

# DNS Resolution Settings
# DNS resolver options for fast timeouts to avoid long waits on
unresolvable hostnames
# This configures the system resolver behavior when load_topo
performs parallel DNS resolution
# Format: timeout:X attempts:Y single-request
# - timeout:X = seconds to wait per DNS query (default: 1)
# - attempts:Y = number of retry attempts (default: 1, no retries)
# - single-request = send A and AAAA queries separately (improves
performance)
CVT_DNS_RES_OPTIONS=timeout:1 attempts:1 single-request

# SSH Configuration
[ssh]
# SSH private key file path for passwordless authentication to HOST
devices only
# NOTE: SSH keys are NOT used for switch devices (switches use
password authentication)
# Used by both SSH commands and SFTP file transfers during agent
deployment to hosts
#
# IMPORTANT: Path must be accessible inside the collector
container, not the host system
# If using Docker volumes, ensure the key file is mounted into
the container
#
# Leave empty to use standard SSH key discovery (recommended)
# When empty, SSH will automatically try default container locations
like:
# - ~/.ssh/id_rsa, ~/.ssh/id_dsa, ~/.ssh/id_ecdsa,
~/.ssh/id_ed25519
#
```

```
# Set to specific path only if you need to use a non-standard key location
# Examples:
#   CV_SSH_KEY_FILE=/opt/collector/keys/host_key     (container path)
#   CV_SSH_KEY_FILE=/home/collector/.ssh/custom_key  (container path)
CV_SSH_KEY_FILE=

# SSH connection timeout in seconds
# Applied to both SSH command execution and SFTP file transfers
# Increase for slow networks, decrease for faster failure detection
SSH_CONN_TIMEOUT=20

# Enable automatic SSH key discovery from SSH agent and default locations
# NOTE: Only applies to HOST devices, not switches
# When enabled, the system will try to use keys from (inside container):
# - SSH agent (if running and accessible in container)
# - Default container locations (~/.ssh/id_rsa, ~/.ssh/id_dsa, etc.)
# Only used for host devices when no password is provided
SSH_LOOK_FOR_KEYS=true

[application]
# Topology loading at startup - specify what to load:
#   none  - do not load any topology file (default)
#   last  - load the last loaded topology from history
#   <path> - load a specific topology file (supports .topo, .dot, .xlsx, .json)
# Examples:
#   STARTUP_TOPOLOGY=none
#   STARTUP_TOPOLOGY=last
#   STARTUP_TOPOLOGY=./topologies/production.topo
#   STARTUP_TOPOLOGY=/absolute/path/to/topology.xlsx
```

```
STARTUP_TOPOLOGY=none

# automatically start validation if a topology file is loaded
AUTO_START_VALIDATION=false

[data management]
# set `true` if you want to poll for stats from CVT collector
ENABLE_STATS_POLLING=false
```

Configuration Parameters

Section	Parameter	Description
network	AGENTS_COLLECTOR_NAT_IP	NAT IP address between collector and agents. Leave empty if no NAT.
	DEFAULT_AGENTS_INTERFACE_NAME	Interface for all agents to communicate with the collector.
	HOST_SPECIFIC_INTERFACE_NAME	Interface for host agents if different from default.
agent	CV_DOT_IN_HOSTNAME	Set `true` if hostname contains extra dots (not in domain).
	FULL_REPORT_PUBLISH_INTERVAL_MINUTES	Minutes until forced full report; min 10 mins.
	AMBER_PUBLISH_EACH_ITERATION	Set True if amber file is to be published every iteration, regardless of changes.
	AGENT_COLLECT_INTERVAL	Agent data collection interval in seconds.
collector	MAX_INACTIVE_INTERVAL	Minutes before inactive agent is considered down.

Section	Parameter	Description
	CHECK_NEW_SWITCHES_INTERVAL	Minutes between checks for new switches.
	START_VALIDATION_TIMEOUT	Minutes allowed for agent to become active after start.
	WAIT_TIME_INACTIVE_AGENTS	Minutes to wait before confirming inactivity.
	CVT_MAX_WORKERS	Max parallel workers.
	CVT_DEPLOYMENT_MAX_WORKERS	Max number of workers for agent deployment.
	CVT_QUICK_TIMEOUT	Quick timeout in seconds for unreachable devices.
	AGENT_COMM_TIMEOUT	Agent communication timeout in seconds - timeout for individual HTTP requests to agents.
	CVT_BATCHING_THRESHOLD	Batching threshold - Use batching for deployments larger than this value.
	CVT_BATCH_SIZE	Batch size when batching is used (devices per batch).
	CVT_DNS_RES_OPTIONS	This configures the system resolver behavior when load_topo performs parallel DNS resolution.
ssh	CV_SSH_KEY_FILE	Path to SSH private key file.
	SSH_CONN_TIMEOUT	SSH connection timeout in seconds.
	SSH_LOOK_FOR_KEYS	When enabled, the system will try to use keys from inside container. Only used for

Section	Parameter	Description
		host devices when no password is provided.
application	STARTUP_TOPOLOGY	Topology loading at startup - specify what to load.
	AUTO_START_VALIDATION	Automatically start validation if a topology file is loaded.
data management	ENABLE_STATS_POLLING	Set `true` if you want to poll for stats from CVT collector.

Setting Environmental Variables with Docker Run

Specifying the Network Interface

If the host system is equipped with multiple network interfaces and the switches are connected to the host through an interface that differs from the default management interface, the user can designate this particular interface by utilizing a specific environment variable, namely **AGENTS_IFC_NAME**. To illustrate, assuming the hypothetical interface name is **eno3**:

```
docker run --name cables_bringup -itd --network=host --env
AGENTS_IFC_NAME=eno3
```

Adding Hostnames

If the switches are not configured in the DNS server, you may add hostnames; the user may use the **--add-host** option when running the container. For example (assuming the switch name is **switch-3245fa** and its IP is **192.168.1.1**):

```
docker run --name cables_bringup -itd --network=host --add-
```

```
host=switch-3245fa:192.168.1.1 cables_bringup
```

Using Volumes

Volumes can be used for data persistence or easier file transfer to the **cables_bringup** container. The volume must be mapped to **/cable_bringup_root** in the container for data persistence. This volume can also be used for loading topology files. Example:

```
docker run --name cables_bringup -itd --network=host -v
/opt/bringup_data:/cable_bringup_root cables_bringup
```

Overriding Apache Configuration

In the event that a host machine is running another Apache instance and utilizing the default ssh ports **443**, an alternative port may be designated for the bringup server by the user, these ports should be available and free. To accomplish this, the **APACHE_HTTPS_PORT** environment variables can be employed. Consider the following example:

```
docker run --name cables_bringup -itd --network=host --env
APACHE_HTTPS_PORT=9443 cables_bringup
```

Deploying the Module as a UFM Enterprise Plugin

⚠ **Warning**

Warning: Please note that Running Cable Validation as plugin is not supported on UFM Gen2.0.

Deploy the module as a UFM Enterprise plugin as follows:

1.
```
docker load -i /<image_path>/ufm-plugin-cablevalidation-<version>.tar.gz
```
2. `./manage_ufm_plugins.sh add -p cablevalidation -t <version>`
3. `./manage_ufm_plugins.sh start -p cablevalidation`
4. `docker exec -it ufm-plugin-cablevalidation bash`

Copy Files to the Plugin

Users have two methods for copying files, such as topology files, to the Cable Validation plugin:

1. Copy the files to the plugin's data volume **/opt/ufm/ufm_plugins_data/cablevalidation** which is mapped to **/data/** inside the plugin container.
2. Use **docker cp** to copy the needed files to the container.

Overriding the Apache Configuration

When using Cable Validation as a plugin, the default ports **443** are already in use by UFM Enterprise. Therefore, port **8633** will be used for HTTPS by default. Users can opt to use different ports for the bring-up server, provided that these ports are available and free.

The plugin **config.cfg** file can be modified to update **APACHE_HTTPS_PORT** variables for that purpose. To make this adjustment, follow these steps:

1. Execute **/opt/ufm/scripts/manage_ufm_plugins.sh add -p cablevalidation** to add the Cable Validation plugin.

2. Stop the plugin using **/opt/ufm/scripts/manage_ufm_plugins.sh stop -p cablevalidation**
3. Use **vim /opt/ufm/files/conf/plugins/cablevalidation/config.cfg** to modify the **'APACHE_HTTPS_PORT'** variable.
4. Update and save the file.
5. Start the plugin again with **/opt/ufm/scripts/manage_ufm_plugins.sh start -p cablevalidation**.

With these changes, the new configuration will take effect, and Apache will run with the updated ports.

Bringup CLI

Running Bringup CLI

Running the bringup CLI can be done in two ways:

1. Direct Execution in the Container:

   ```
   docker exec -it cables_bringup bringupcli
   ```

2. Alternatively, it is possible to run exec bash in the container and run the appropriate CLI command based on the fabric type from anywhere within the container:

-
    - Start a bash session in the container:

      ```
      docker exec -it cables_bringup
      ```

    - Execute the desired CLI command:

```
# stop cvt service
supervisorct stop cvt-service
# run bringupcli
bringupcli -k
```

Bringupcli Usage

bringupcli may have command line arguments, see usage below for more details:

Note

root@r-ufm65:/# bringupcli -h

usage: bringupcli \[-h\] \[-V\] \[-k\] \[-d\]

Optional Arguments

Argument	Description
`-h, --help`	Show this help message and exit
`-v, --version`	Show program version number and exit
`-k, --kill-other-sessions`	Kill other CLI sessions if existent
`-d, --daemon`	Run as daemon

Bringupcli Commands

Load Topology Commands

load_topo

Description: Loads topo file if the fabric is InfiniBand, and dot file if the fabric is Ethernet.

```
load_topo <filename> dns=<true|false> cluster=<cluster name >
```

Parameters:

- `filename`: absolute path for the topo/dot file
- `dns`(Optional): assumes that DNS is active, and you can access the switches by hostnames by default dns=true.
- `cluster`(Optional): cluster name, if cluster name is provided it will be set to the provided value, else it will be set to 'default'.

load_ptp

Description: Loads PTP topology file (Excel file).

```
 load_ptp <filename> format=<legacy_ib|legacy_eth|unified_topo>
dns=<true|false> cluster=<cluster name> sheets=<comma separated
sheets> dc_layout=<file path> hca_mapping=<file path>
```

Parameters:

- `filename`: The absolute path for the P2P file.
- `format`: The format of ptp is of legacy or unified topology:
  - legacy_ib: allows to load legacy ib ptp file
  - legacy_eth: allows to load legacy eth ptp file
  - unified_topo: allows to load the unified ptp file which supports IB, eth, xdr and nvlink protocols
- `dns`(Optional): Assumes that DNS is active, and you can access the switches by hostnames by default `dns=true`.

- `cluster`(Optional): Cluster name, *if cluster name is provided it will be set to the provided value, else it will be set to 'default'.*
- `sheets`(Optional): comma separated sheets name. If provided, only the specified sheets from the Excel file are loaded. If not provided, all sheets in the file are loaded.
- `dc_layout`(Optional, Ethernet Fabric Only): A CSV file that describes the data center layout, for more information please find DC Floor Layout File.
- `hca_mapping`(Optional, InfiniBand and NVOS Fabrics Only): A CSV file that defines the relationship between port numbers and HCA names, for more information please find HCA Mapping File.

load_ip

Description: Loads switch IP addresses, can be used if DNS is inactive.

```
load_ip <filename> cluster=<cluster name >
```

Loads the IP/switch-name mapping, to allow reaching the switch via REST API to retrieve local topology, GUID, etc. The file format is pairs of IP addresses and hostname. This file will be used in association with a 'topo' file in case DNS is unavailable.

An IP file example:

```
A comment
10.0.30 switch1
10.0.0.31 switch2
```

Parameters:

- `filename`: The absolute path for the IP file.
- `cluster`(Optional): The cluster name, *if cluster name is provided it will be set to the provided value, else it will be set to 'default'.*

load

Description: Loads both IP addresses and topo files.

```
load <topo filename> <ip filename> cluster=<cluster name>
```

Loads the .topo and .ip files.

> **Note**
>
> Note: if you have multiple files describing a topology, use the commands:
>
> `load_ip file.ip`
>
> `load_topo file1.topo file2.topo file3.topo`

Parameters:

- `topo filename`: The absolute path for the topology file directory
- `ip filename`: The absolute path for the IP file directory
- `cluster`(Optional): The cluster name, *if cluster name is provided it will be set to the provided value, else it will be set to 'default'.*

load_clusters

> **Note**

Note: will be deprecated in coming release

Clusters file should have the following format, where topo file should be in xlsx format and the IP file is optional.

```
load_clusters <filename> dns=true|false
```

```
cluster_name, topo_file, ip_file
CLUSTER1, cluster1_topo.xlsx, cluster1.ip
CLUSTER2, cluster2_topo.xlsx,
```

Parameters**:**

- `filename`: The absolute path for the cluster file
- `dns`(optional): Assumes that DNS is active, and you can access the switches by hostnames by default dns=true.

Validations Commands

show_clusters

Description: Show list of loaded clusters as loaded from the clusters file.

```
show_clusters
```

show_switches

Description**:** Show list of loaded switches as loaded from the topology file

```
show_switches cluster=<cluster_name>
```

Parameters:

- `cluster` (Optional): cluster name, If the cluster name is provided, show the switch in the given cluster only.

Output Example:

```
MQM8700 sw-hdr-proton01
-----------------------
MQM8700 sw-hdr-proton01 P3 --> swx-proton03 mlx5_0 P1
MQM8700 sw-hdr-proton01 P4 --> swx-proton04 mlx5_2 P1
MQM8700 ufm-sw-hdr01
--------------------
MQM8700 ufm-sw-hdr01 P1 --> ufm-sw-hdr02 P1
MQM8700 ufm-sw-hdr02
--------------------
MQM8700 ufm-sw-hdr02 P1 --> ufm-sw-hdr01 P1
```

check_switch_status

Description: Check switch connectivity status (Ping/JSON-API/Agent)

```
check_switch_status cluster=<cluster name >
```

Parameters:

- `cluster` (Optional): cluster name, If the cluster is provided, the check will be done for the switches in the provided cluster only.

Output Example:

```
Host IP ping JSONAPI Agent
----------------------------- ------------- ---- ---- -----
sw-hdr-proton01.mtr.labs.mlnx 209.44.74 True True True
ufm-sw-hdr01.mtr.labs.mlnx 10.209.36.113 True True True
ufm-sw-hdr02.mtr.labs.mlnx 10.209.36.122 True True True
```

start_validation

Description: Push topology to switches and get validation reports.

```
start_validation timeout=<n> cluster=<cluster_name>
```

Parameters:

- `cluster` (Optional): cluster name, If the cluster is provided, the validation will be started in the switches in the provided cluster only.
- `timeout` (Optional): timeout in which validation stops, n is in seconds (s), minutes (m), hours (h) or days (d). For example timeout=20m or timeout=2h.

If timeout is not provided, use the `stop_validation` command to stop it.

stop_validation

Description: Stops validation routine. Unsubscribe from getting switches updates.

```
stop_validation
```

Troubleshooting

	Description	Example
deploy_single_agent	Deploys agent on a specific node will be deprecated soon. use `deploy_agents` instead.	`deploy_single_agent <switch-ip/host-ip>`
deploy_all_agents	Deploys agents on loaded nodes that have no agents. will be deprecated soon. use `deploy_agents` instead.	`deploy_all_agents`

deploy_agents	Deploys agents on all or specific nodes.	`deploy_agents all`
		`deploy_agents node_ip1 node_ip2`

remove_all_agents	Removes agents from loaded nodes that have agents. will be deprecated soon. use `remove_agents` instead.	`remove_all_agents`
remove_single_agent	Removes an agent from a specific node. will be deprecated soon. use `remove_agents` instead.	`remove_single_agent <switch-ip/host-ip>`

remove_agents	Remove agents on from all or specific nodes.	`remove_agents all`
		`remove_agents node_ip1 node_ip2`

Set Credentials Commands

show_creds

Description:Display the credentials for all nodes

```
show_creds [format=json|report]
```

These credentials are used for communication with switches and hosts.

Parameters:

- `format`: the format for displayed data .

set_default_creds

Description: Sets the default switch/host credentials to override the built-in default credentials.

```
set_default_creds user=<user> pwd=<pwd> type=<switch|host> save=
<true|false>
```

These credentials are used for communication with any switch that does not have specific credentials.

Parameters:

- `user`: user name.
- `pwd`: password.
- `type`(Optional): the default value is switch
- `save`(Optional): If save it set to true (default: true), credentials will be saved encrypted to a file

set_node_creds

Description: Sets the credentials for a specific switch/host, it can be used when the switch credentials are different than the defaults.

```
set_node_creds <switch> user=<user> pwd=<pwd> save=true|false
```

Parameters:

- `switch`: switch name
- `user`: user name.
- `pwd`: password.
- `save`(Optional): If save it set to true (default: true), credentials will be saved encrypted to a file

remove_node_creds

Description: remove the credentials for a specific switch/host.

```
remove_node_creds <switch>
```

Parameters:

- `switch`: switch name

set_credential_profile

Description: Sets the credentials for a credential profile name specified in unified topology ptp file.

```
set_credential_profile <credential_profile_name> user=<user> pwd=
<pwd> [save=true|false]
```

Parameters:

- `credential_profile_name`: Credential Profile name.
- `user`: user name.
- `pwd`: password.
- `save`(Optional): If save it set to true (default: true), credentials will be saved encrypted to a file.

remove_credential_profile

Description: remove the credentials associated to a credential profile.

```
remove_credential_profile <credential_profile_name>
```

Parameters:

- `credential_profile_name`: Credential Profile name

Web User Commands

add_web_login

Description: Add new users to login to the web gui apart from the default 'admin' login.

```
add_web_login user=<user> pwd=<pwd> account_type=<account_type>
```

Parameters:

- `user`: username

- `pwd`: password
- `account_type`: Account_type can be cabler, admin, nvidia, or developer

delete_web_user

Description: Delete a web user account.

```
delete_web_user user=<user>
```

Parameters:

- `user`: username.

show_web_users

Description: Shows the web users added along with their account types.

```
show_web_users
```

Example output:

```
admin: admin
nUser: nvidia
dUser: developer
```

update_web_user

Description: Update the specified web user, allowing changes to the password, account type, or both.

```
update_web_user user=<user> pwd=<pwd> account_type=<account_type>
```

Parameters:

- `user`: username
- `pwd`: password
- `account_type`: Account_type can be cabler, admin, nvidia, or developer

Other Commands

show_switch_history

Description: Lists data files collected from switches in the last days

```
show_switch_history <switch names> past=<n> start_time=<n>
end_time=<n> prev=<n>
```

Parameters:

- `switch_names`(Optional): a space delimited switch names, if no switches are provided, it will bring data of all switches
- `past` (Optional): Past argument can be used to specify the history interval, by default it is set to one week past=1w.
- `start_time`, `end_time`(Optional): time period to look for data, if no time is provided, data of the last week will be provided
- `prev`(Optional): retrieve data during the previous period from specified time to now.

'prev' formats: num[day|week]

amber_show_latest

Description: Shows latest collected amber data from switches

```
amber_show_latest
```

show_incoming_reports

Description: show or hide incoming reports

```
show_incoming_reports show=true/false
```

Parameters:

- `show` (Optional): the default value is true

show_statistics

Description: Shows a summary for all issues

```
show_statistics
2024-11-21 12:58:46.726351 - show statistics for default cluster:
Status Number of occurrences Nodes affected
------------------------------------ --------------------- ------
--------
No Transceiver 0 0
Link Down, No signal 0 0
Admin Down 0 0
```

```
ErrDisable - Flap 0 0
ErrDisable - Rx 0 0
Negotiation fail 0 0
Wrong-neighbor 0 0
Wrong-port 0 0
Unknown-neighbor 0 0
Extra-cable 0 0
Underperforming-link (BER) 0 0
Flapping-link 0 0
Anomalous-port (Signal, Temperature) 0 0
Unreachable-device 0 0
Correctly-wired 0
No report
```

show_switches

Description: Shows the current loaded switches

```
show_switches
```

Example output:

```
MQM8700 sw-hdr-proton01, rack: PXX, unit: 30
--------------------------------------------
MQM8700 sw-hdr-proton01 P3 --> swx-proton03 mlx5_0 P1
MQM8700 sw-hdr-proton01 P4 --> swx-proton04 mlx5_2 P1
```

add_certificate

Description: Updates the SSL certificate file used by Apache for secure connections

```
add_certificate <crt file> <key_file>
```

The provided file should be a valid SSL certificate file in crt format. The old certificate file will be backed up before replacing it with the new one.

version

Description: Shows application version.

exit

Description: Exits the application.

help

Description: Shows a list of commands. For help on a specific command, run help <command>

Tool Initializing Using Bringup CLI

To initialize the tool, perform the following:

1. Run bringupcli
2. Open bringup GUI
3. Load the fabric topology file using one of the Load Topology Commands or from the bringup GUI
4. Set the credentials for the switches.
5. Deploy the agent on all switches.

Note

`wget` or `curl` should be installed on the switches and hosts as it is used to download the agent image from the collector

6. Start validation using GUI or using the Validation Command.

Running bringup GUI

1. Open the following URL in the browser:
   https://<bringup_machine_ip>/cables_validation

2. Enter credentials in the login page.

3. You may change the default self signed certificate located by default in the container at:

```
SSLCertificateFile ${BRINGUP_CONF_APACHE_PATH}/certs/cv-
cert.crt
SSLCertificateKeyFile ${BRINGUP_CONF_APACHE_PATH}/private/cv-
cert.key
```

Update Certificate

To update a certificate, run the following command:

```
 add_certificate <crt file> <key_file>: update the ssl
certificate
```

CVT as a Service

Overview

The Cable Validation Tool (CVT) can be deployed and managed as a service, providing automated cable validation capabilities with persistent operation, automatic startup, and web-based management. This document covers the service architecture, configuration, deployment, and management of CVT in service mode.

Architecture

The CVT service architecture consists of multiple components managed by `supervisord.`

```
              CVT Container                <p>
</p>                                            <p></p>
supervisord                  <p></p>
                                         <p></p>              CVT
Service                   <p></p>
                                      <p></p>            BringupCLI
Engine               <p></p>         • Topology Management
    <p></p>         • Agent Deployment                 <p></p>
   • Validation Orchestration         <p></p>
                                      <p></p>
                                      <p></p>            Web Service
    <p></p>         • REST API                         <p></p>
   • Web UI                           <p></p>         •
Authentication                   <p></p>
                                      <p></p>
                                         <p></p>
                                         <p></p>            Apache
Web Server               <p></p>      • HTTPS/SSL Support
   <p></p>      • Reverse Proxy                       <p></p>      •
Static Content Serving              <p></p>
                                         <p></p>
                                         <p></p>           CV
Controller                    <p></p>      • Service Management API
   <p></p>      • System Administration               <p></p>
                                         <p>
</p>
```

Service Components

CVT Service (cvt-service)

- **Purpose**: Main CVT application running in daemon mode
- **Command**: `/usr/local/bin/cvt-service --daemon`
- **Features**:
    - Automatic topology loading
    - Validation orchestration
    - Agent management
    - Signal handling (SIGTERM, SIGINT)

Apache Web Server (apache)

- **Purpose**: HTTPS frontend and reverse proxy
- **Command**: `/usr/local/bin/apache2_wrapper.sh`
- **Features**:
    - SSL/TLS termination
    - Web UI serving
    - API reverse proxy
    - Authentication handling

CV Controller (cv_controller)

- **Purpose**: Service management and system administration
- **Command**: `/usr/local/bin/run_cv_controller.sh`

- **Features**:
  - Service start/stop operations
  - System status monitoring
  - Configuration management

Cron Service (cron)

- **Purpose**: Scheduled tasks and maintenance
- **Command**: `cron -f`
- **Features**:
  - Log rotation
  - Periodic cleanup
  - Scheduled reports

Installation and Deployment

Docker Deployment for CVT Service

The CVT service is deployed using a pre-built Docker image (bringup service image). This allows for a consistent and isolated runtime environment.

1. Load the CVT Collector Service Image.

   If you have a local Docker image file (`.tar.gz`), load it using:

   ```
   docker load -i /path/to/cables_bringup_{version}.tar.gz
   ```

2. [Optional] Pull the Image from a Registry

If the image is available in a Docker registry, you can pull it directly:

```
docker pull mellanox/cables_bringup:{version}
```

3. Run the CVT Service Container: Use the following command to start the CVT service:

```
docker run -itd \
  --name cables_bringup \
  --network=host \
  {override/add env variables} \
  -v /path/to/data:/cable_bringup_root \
  mellanox/cables_bringup:{version}
```

Options:

Option	**Description**
-itd	Run container in interactive mode, allocate a pseudo-TTY, and run in detached mode
--name cables_bringup	Assigns a name to the container for easier management.
--network=host	Shares the host's network stack for the container.
{override/add env variables}	Specify any environment variables needed by the service.
-v /path/to/data:/cable_bringup_root	Mounts a host directory into the container for persistent data.
mellanox/cables_bringup:{version}	The Docker image and version tag to run.

Configuration

Environment Variables

The CVT service behavior is controlled through environment variables and configuration files:

Configuration File (/etc/cablevalidation/cvt_env.conf)

Run the following command to edit the file and restart cvt-service:

```
supervisorctl restart cvt-service
```

For more information, refer to CVT Configuration.

Environment Variable Configuration

Run the following command to configure environment variables:

```
docker run
```

Web Service Configuration

Web service configuration can be done only in the `docker run` command.

```
# Web service interface and ports
export CVT_WEB_INTERFACE=127.0.0.1
export CVT_WEB_SERVICE_PORT=8251
export APACHE_HTTPS_PORT=443

# Controller service port
export CV_CONTROLLER_PORT=8252
```

UFM Plugin Mode

Install the CVT UFM plugin using the UFM CLI or Web UI, like any other plugin.

NetQ Plugin Mode

TBD

Supervisord Configuration

The service is managed by `supervisord` with the following configuration:

Directive	Description
`command`	Command to start the CVT service in daemon mode.
`autostart`	Automatically starts the service when Supervisor starts.
`autorestart`	Restarts the service automatically if it exits unexpectedly.
`startretries`	Number of retry attempts if the service fails to start.
`startsecs`	Number of seconds the service must stay running to be considered started.
`stderr_logfile`	Path to the log file for standard error output.
`stdout_logfile`	Path to the log file for standard output.
`stdout_logfile_maxbytes` / `stderr_logfile_maxbytes`	Maximum size of each log file before rotation.
`stdout_logfile_backups` / `stderr_logfile_backups`	Number of rotated log backups to keep.
`user`	The user under which the service runs.
`killasgroup`	Terminates all processes in the service’s process group on stop.

Directive	Description
`stopasgroup`	Sends stop signals to the entire process group.
`environment`	Sets environment variables for the service (`PYTHONUNBUFFERED=1` ensures unbuffered Python output).

Service Management

CVT Service Management with Supervisor

The CVT service is managed through **Supervisor**, which allows you to start, stop, restart, and monitor services on your appliance. Below are the common commands:

Service	Description	Command
Start CVT Service	Starts the CVT service if it is not already running.	`supervisorctl start cvt-service`
Stop CVT Service	Stops the CVT service gracefully.	`supervisorctl stop cvt-service`
Restart CVT Service	Stops and then starts the CVT service. Useful for applying configuration changes.	`supervisorctl restart cvt-service`
Check Service Status	Displays the current status of the CVT service (e.g., RUNNING, STOPPED, FATAL).	`supervisorctl status cvt-service`
View Service Logs	Displays real-time logs from the CVT service. Useful for troubleshooting.	`supervisorctl tail cvt-service`
Start All Services	Starts all services managed by Supervisor.	`supervisorctl start all`

Service	Description	Command
Reload Configuration	- **reread**: Reloads Supervisor configuration files to detect new or removed programs. - **update**: Applies the configuration changes, starting or stopping services as needed.	supervisorctl reread && supervisorctl update

CVT Service Commands

Command	Description
`supervisorctl start cvt-service`	Starts the CVT service if it is not already running.
`supervisorctl stop cvt-service`	Stops the CVT service gracefully.
`supervisorctl restart cvt-service`	Restarts the CVT service. Useful for applying configuration changes.
`supervisorctl status cvt-service`	Displays the current status of the CVT service (RUNNING, STOPPED, FATAL).
`supervisorctl tail cvt-service`	Displays real-time logs from the CVT service for troubleshooting.
`supervisorctl start all`	Starts all services managed by Supervisor.
`supervisorctl reread && supervisorctl update`	Reloads configuration

Using CVT Controller API

The CVT Controller provides programmatic service management:

Command	Description
`curl -X POST http://localhost:8252/start`	Starts the CVT service via the controller.
`curl -X POST http://localhost:8252/stop`	Stops the CVT service via the controller.
`curl -X GET http://localhost:8252/status`	Retrieves the CVT service status via the controller. **Note:** This feature is not supported yet.

Legacy Mode Operation

For troubleshooting, development, or manual operations, you can stop the CVT service and run the traditional interactive CLI:

Stop Service and Run Legacy CLI

1. Stop the CVT Service

```
supervisorctl stop cvt-service
```

2. Run the Legacy Bringup CLI:

```
bringupcli
```

Legacy CLI Features

When running in legacy mode, you get:

- **Interactive Command Shell**: Full command-line interface with tab completion
- **Manual Control**: Step-by-step topology loading and validation control

- **Real-time Feedback**: Immediate command output and status updates
- **Debugging**: Easier debugging with direct command execution

Common Legacy Mode Workflow

1. Stop the CVT service:

```
supervisorctl stop cvt-service
```

2. Start interactive CLI:

```
bringupcli
```

3. Perform manual operations:

```
Cable Bringup: load_topo /path/to/topology.topo
Cable Bringup: deploy_all_agents
Cable Bringup: start_validation
Cable Bringup: show_switches
Cable Bringup: exit
```

4. Restart service when done:

```
upervisorctl start cvt-service
```

When to Use Legacy Mode

- **Initial Setup**: First-time configuration and testing
- **Troubleshooting**: Debugging connectivity or configuration issues
- **Manual Operations**: One-time tasks that don't require automation
- **Development**: Testing new topologies or validation scenarios
- **Training**: Learning CVT commands and workflows

Switching Back to Service Mode

1. Exit the legacy CLI:

```
Cable Bringup: exit
```

2. Restart the CVT service:

```
supervisorctl start cvt-service
```

3. Verify that the service is running:

```
supervisorctl status cvt-service
```

Note

While in legacy mode, automatic features (e.g., topology loading and validation) are disabled. The web UI remains accessible. Restarting the service restores full automatic functionality.

Command Line Options

The CVT service supports several command line options:

Command	Description
`cvt-service --daemon`	Runs the CVT service as a background daemon.
`cvt-service --kill-other-sessions`	Terminates any other running CVT sessions before starting a new one.
`cvt-service --version`	Displays the current version of the CVT service.
`cvt-service`	Starts the CVT service in interactive mode, useful for debugging.

Automatic Features

Topology Loading

The CVT service can automatically load topology files on startup based on the `STARTUP_TOPOLOGY` configuration:

Load Last Topology

```
STARTUP_TOPOLOGY=last
```

- Loads the most recently used topology from history
- Supports all topology formats (.topo, .dot, .xlsx, .json)

Load Specific File

```
STARTUP_TOPOLOGY=/path/to/topology.topo
```

- Loads a specific topology file on startup
- Supports absolute and relative paths
- Auto-detects file format based on extension

Supported Formats

- `.topo`: Native topology format
- `.dot`: Graphviz DOT format
- `.xlsx`: Unified topology Excel format
- `.json`: JSON topology format

Automatic Validation

Enable automatic validation start after topology loading:

```
AUTO_START_VALIDATION=true
```

Behavior:

- Only starts if topology was successfully loaded
- Uses validation settings from history
- Logs all activities for monitoring

Signal Handling

The service handles system signals gracefully:

- **SIGTERM**: Graceful shutdown with cleanup
- **SIGINT**:
    - Daemon mode: Graceful shutdown
    - Interactive mode: Cancel current operation (double Ctrl-C to exit)

Monitoring and Logging

Log Files

All service logs are centrally managed:

```
/cable_bringup_root/log/<p></p>    cvt-service.log              #
Main service logs<p></p>    cv_controller_service.log    #
Controller logs<p></p>    apache2/<p></p>        access.log
# Web access logs<p></p>        error.log              # Apache
error logs<p></p>    supervisord.log             # Supervisor logs
```

Log Rotation

Logs are automatically rotated with the following settings:

- **Max size**: 10MB per log file
- **Backups**: 7 historical files
- **Format**: Timestamped entries with log levels

Health Monitoring

Monitor service health using:

Command	Description
`supervisorctl status`	Checks the status of all services managed by Supervisor.
`supervisorctl tail -f cvt-service`	Monitors CVT service logs in real-time.
`systemctl status apache2`	Displays the current status of the Apache web server.
`htop`	Interactive tool to monitor system processes and resource usage.
`iostat -x 1`	Displays detailed I/O statistics for devices every 1

Command	Description
	second.

Troubleshooting

Common Issues

Issue	Troubleshooting
Service Won't Start	1. **Check Supervisord Status**: supervisorctl statussystemctl status supervisor 2. **Check Log Files**: tail -f /cable_bringup_root/log/cvt-service.logtail -f /var/log/supervisor/supervisord.log 3. **Verify Configuration**: supervisorctl rereadsupervisorctl update
Port Conflicts	1. **Check Port Usage**: netstat -tlnp | grep -E ':(443|8251|8252)'lsof -i :443 2. **Update Port Configuration**: # Edit environment variablesexport CVT_WEB_SERVICE_PORT=8253export CV_CONTROLLER_PORT=8254
Permission Issues	1. **Check File Permissions**: ls -la /cable_bringup_root/ls -la /etc/cablevalidation/ 2. **Fix Ownership**: chown -R root:root /cable_bringup_root/chmod -R 755 /cable_bringup_root/
Memory/Performance Issues	1. **Monitor Resource Usage**: htopfree -hdf -h 2. **Adjust Worker Limits**:

Issue	Troubleshooting
	`# In cvt_env.confCVT_MAX_WORKERS=15  # Reduce for limited memory`

Topology Loading Failures	1. **Check File Paths**:
	`ls -la /path/to/topology/file`
	2. **Validate File Format**:
	`file topology.topohead -n 10 topology.topo`
	3. **Check History**:
	`# View topology loading history in logsgrep "Loading topology" /cable_bringup_root/log/cvt-service.log`

Debug Mode

For detailed debugging, run CVT in interactive mode:

Command	Description
`supervisorctl stop cvt-service`	Stops the CVT service managed by Supervisor.
`/usr/local/bin/cvt-service`	Runs the CVT service in interactive mode.
`PYTHONUNBUFFERED=1 /usr/local/bin/cvt-service`	Runs the CVT service interactively with debug logging enabled.

Log Analysis

Analyze logs for common patterns:

Command	Description
`grep -i error /cable_bringup_root/log/cvt-service.log`	Searches the CVT service log for error messages.
`grep -i "starting|daemon" /cable_bringup_root/log/cvt-service.log`	Monitors the startup sequence of the CVT service.

Command	Description
`grep -i "topology|load" /cable_bringup_root/log/cvt-service.log`	Checks log entries related to topology loading.
`grep -i "validation|start|stop" /cable_bringup_root/log/cvt-service.log`	Monitors validation activities and their start/stop events.

Best Practices

	Flow
Security	1. **Use HTTPS**: Always use HTTPS for production deployments 2. **Strong Authentication**: Configure strong passwords for web users 3. **SSH Keys**: Use SSH key-based authentication for agent deployment 4. **Firewall**: Restrict access to necessary ports only 5. **Regular Updates**: Keep CVT and system packages updated
Performance	1. **Resource Allocation**: Ensure adequate CPU and memory resources 2. **Worker Tuning**: Adjust `CVT_MAX_WORKERS` based on system capacity 3. **Log Management**: Implement log rotation and archival 4. **Monitoring**: Set up comprehensive monitoring and alerting
Reliability	1. **Backup Configuration**: Regularly backup configuration files and data 2. **Health Checks**: Implement automated health monitoring 3. **Graceful Shutdown**: Always use proper shutdown procedures 4. **Recovery Procedures**: Document and test disaster recovery procedures
Maintenance	1. **Regular Monitoring**: Check service status and logs regularly 2. **Capacity Planning**: Monitor resource usage and plan for growth

	Flow
	3. **Update Procedures**: Establish procedures for updates and patches 4. **Documentation**: Keep configuration and operational documentation current
Development and Testing	1. **Staging Environment**: Use a staging environment for testing changes 2. **Configuration Management**: Use version control for configuration files 3. **Automated Testing**: Implement automated testing for service functionality 4. **Rollback Procedures**: Have rollback procedures ready for failed deployments

CVT Configuration

CVT Environment Configuration (cvt_env.conf)

This document outlines the configuration options available in the `cvt_env.conf` file, which controls various aspects of the Cable Validation Tool (CVT) collector and agent behavior.

Configuring CVT

CVT can be configured through two primary methods:

1. **Environment Variables:** Setting system-level environment variables.

2. **Configuration File (`cvt_env.conf`):** Defining variables within the `cvt_env.conf` file.

Precedence

When a configuration option is defined in both the environment variables and the `cvt_env.conf` file, **environment variables take precedence** over the values specified in the configuration file.

For example, if `AGENT_COLLECT_INTERVAL` is set to `300` in your shell's environment variables and to `600` in `cvt_env.conf`, CVT will use `300` as the collection interval.

Changing Configuration

- **Via Configuration File:** To apply changes made to the `cvt_env.conf` file, you must edit the file and then **restart the CVT service:**
  ```
  supervisorctl restart cvt-service
  ```
- **Via Environment Variables:** To apply changes using environment variables, you need to **set the environment variables in the Docker container run command** when launching the CVT service.
  - Example:
    ```
    docker run --name cables_bringup -itd --env VAR1=value1 --env VAR2=value2 --network=host mellanox/cables_bringup:<version>
    ```

Network Configuration

Variable	Description	Default
`AGENTS_COLLECTOR_NAT_IP`	Collector External (NAT) IP address. Define this if there is a NAT between the collector and the agents. This IP is used by agents to fetch images and send data/reports.	Empty (r
`DEFAULT_AGENTS_INTERFACE_NAME`	Interface name of the collector over which all agents (switch and host) communicate. The IP of this interface is used by agents to fetch images and send data/reports.	Empty (I
`HOST_SPECIFIC_INTERFACE_NAME`	Use if a different interface is desired for host agents, separate from `DEFAULT_AGENTS_INTERFACE_NAME`. The IP of this interface is used by	Empty (u `DEFAUL`)

Variable	Description	Default
	host agents to communicate with the collector.	

Agent Configuration

Variable	Description	Value / Unit	Default	Const
`CV_DOT_IN_HOSTNAME`	Set to true if the switch hostname contains a dot (other than the domain part).	Boolean	Empty (implies false)	—
`FULL_REPORT_PUBLISH_INTERVAL_MINUTES`	Time after which a full report is forced to be published.	Minutes	720 (12 hours)	Interva than 1 minut not suppo
`AMBER_PUBLISH_EACH_ITERATION`	Set to true to publish amber data on each agent iteration, regardless of changes.	Boolean	false	—
`AGENT_COLLECT_INTERVAL`	Agent data collection interval. Controls how often the agent collects and processes	Seconds	600 (10 minutes)	—

Variable	Description	Value / Unit	Default	Const
	port/link data.			

Collector Configuration

Variable	Description	Value / Unit	Default	Co No
`MAX_INACTIVE_INTERVAL`	Maximum time to wait for an agent to become inactive.	Minutes	15	—
`CHECK_NEW_SWITCHES_INTERVAL`	Interval to check for new switches.	Minutes	15	—
`START_VALIDATION_TIMEOUT`	Time to wait for an agent to become active after starting validation.	Minutes	5	—
`WAIT_TIME_INACTIVE_AGENTS`	Time to wait for an agent to become inactive.	Minutes	1	—
`CVT_MAX_WORKERS`	Maximum number of workers to run in parallel for general operations (validation, connectivity, DNS).	—	30	—
`CVT_DEPLOYMENT_MAX_WORKERS`	Maximum number of workers for agent deployment	—	30	—

Variable	Description	Value / Unit	Default	Co No
	(limited due to image transfers).			
`CVT_QUICK_TIMEOUT`	Quick timeout for unreachable devices, reducing wait time for failed connections.	Seconds	3	—
`AGENT_COMM_TIMEOUT`	Agent communication timeout for individual HTTP requests to agents.	Seconds	30	—
`CVT_BATCHING_THRESHOLD`	Batching threshold; batching is only used for deployments larger than this value to reduce overhead.	—	5000	—
`CVT_BATCH_SIZE`	Batch size when batching is used (devices per batch).	—	1000	—
`CVT_DNS_RES_OPTIONS`	DNS resolver options for fast timeouts to avoid long waits on unresolvable hostnames. Configures system resolver behavior when	Format: `timeout:X attempts:Y single-request`	`timeout:1 attempts:1 single-request`	`ti` Se wa qu `at` : N ret att `si re`

Variable	Description	Value / Unit	Default	Co No
	`load_topo` performs parallel DNS resolution.			: Se AA sep imp per

SSH Configuration

Variable	Description	Value / Unit	Default
`CV_SSH_KEY_FILE`	SSH private key file path for passwordless authentication to **HOST** devices only. SSH keys are **not** used for switch devices. Must be accessible inside the collector container. Leave empty to use standard SSH key discovery.	Path	Empty (uses standard SSH key discovery)
`SSH_CONN_TIMEOUT`	SSH connection timeout for both SSH command execution and SFTP file transfers. Increase for slow networks, decrease for faster failure detection.	Seconds	20
`SSH_LOOK_FOR_KEYS`	Enable automatic SSH key discovery from SSH agent and default locations. Applies only to HOST devices when no password is provided.	Boolean	true

Application Configuration

Variable	Description	Options / Value	Default
`STARTUP_TOPOLOGY`	Topology loading at startup. Specifies which topology file to load.	`none`: Do not load any file `last`: Load the last loaded topology from history `<path>`: Load a specific topology file (	none

Variable	Description	Options / Value	Default
		`.topo`, `.dot`, `.xlsx`, `.json`)	
`AUTO_START_VALIDATION`	Automatically start validation if a topology file is loaded.	Boolean	false

Data Management

Note

Note: for future use.

Variable	Description	Default
`ENABLE_STATS_POLLING`	Set to true to enable polling for stats from the CVT collector.	false

High Availability (HA) Mode Support for CVT Plugin

Overview

The CVT (Cable Validation Tool) plugin supports High Availability (HA) mode to ensure continuous operation and service resilience in production environments. In an HA deployment, CVT operates in an active-passive configuration where the collector runs on only one node at a time. When a failover event occurs, the CVT collector on the active node (node1) is stopped, and the CVT collector on the standby node (node2) automatically starts up. This architecture ensures that cable validation monitoring continues with minimal interruption, which is critical for maintaining network health and detecting issues in real-time.

HA Architecture

The HA implementation for CVT follows a shared-storage model where both nodes have access to common configuration and data files. This shared storage ensures that when the standby node takes over, it has immediate access to the same topology data, validation history, and configuration that was being used by the primary node. The shared resources include:

- **Configuration files** - All CVT configuration settings, including `cvt_env.conf`
- **Topology data** - Current and historical topology files
- **Database and persistent storage** - All collected metrics and historical information

Note on Validation State: The runtime validation state (agent status, validation results, and operational data) is maintained in memory only and is not persisted to shared storage. When a failover occurs, the standby node will reload the topology from shared storage and restart validation. The validation state will be rebuilt as agents are redeployed and begin reporting data. This means there will be a brief period during failover where the validation state is being re-established, but the topology configuration and historical data remain intact.

When a failover occurs—whether due to planned maintenance, system failures, or network issues—the standby node can quickly resume operations by loading the same topology and restarting validation, minimizing the gap in validation coverage.

Failover Configuration

To enable seamless automatic failover in HA mode, the CVT plugin requires specific configuration settings that allow the standby collector to automatically resume validation operations immediately upon startup. Without proper configuration, manual intervention would be required after each failover to reload the topology and restart validation, which would defeat the purpose of having an HA setup. The automated recovery mechanism ensures that network monitoring remains consistent and continuous across failover events.

Required Configuration Settings

For proper HA failover support, users must configure two critical parameters in the `cvt_env.conf` file under the `[application]` section:

Setting	Description
STARTUP_TOPOLOGY=last	This setting instructs the collector to automatically load the last successfully loaded topology from the history when starting up. In a failover scenario, this ensures that the standby collector immediately restores the exact topology state that was active on the primary collector before the failover. Since both nodes share the same data files, the "last" topology reference points to the same topology file that was in use on the primary node. Without this setting, the collector would start with no topology loaded (the default `none` value), requiring manual topology loading after every failover and causing a gap in validation coverage.
AUTO_START_VALIDATION=true	This setting enables automatic validation startup once a topology is successfully loaded. When combined with `STARTUP_TOPOLOGY=last`, this creates a fully automated recovery process: upon startup, the standby collector loads the last topology from shared storage and immediately begins validation operations without requiring any user intervention. This is essential for maintaining continuous validation during failover events, as it eliminates the manual steps that would otherwise be needed to resume monitoring. The validation process will automatically deploy agents to network devices and resume cable validation exactly as it was running on the primary node.

Complete HA Configuration Example

To configure CVT for HA mode with automatic failover support, update the `[application]` section in `cvt_env.conf`:

```
[application]

# Topology loading at startup - specify what to load:

#   none  - do not load any topology file (default)

#   last  - load the last loaded topology from history

#   <path> - load a specific topology file (supports .topo, .dot, .xlsx, .json)
# For HA mode: set to 'last' to enable automatic topology restore on failover

STARTUP_TOPOLOGY=last

# Automatically start validation if a topology file is loaded

# For HA mode: set to 'true' to enable automatic validation resume on failover

AUTO_START_VALIDATION=true
```

Failover Behavior

With the above configuration, the failover sequence operates as follows:

1. **Primary node (node1) failure** - The CVT collector on node1 stops due to hardware failure, network issue, or planned maintenance
2. **HA cluster failover** - The cluster management system (e.g., Pacemaker, Kubernetes, or other HA solution) detects the failure and initiates failover to node2
3. **Standby node (node2) startup** - The CVT collector starts on node2
4. **Automatic topology restore** - CVT reads the `STARTUP_TOPOLOGY=last` setting and automatically loads the last topology from the shared storage
5. **Automatic validation start** - CVT reads the `AUTO_START_VALIDATION=true` setting and immediately begins validation operations
6. **Validation state rebuild** - Agents are redeployed to network devices and begin reporting data. The runtime validation state (agent status, current validation results) is rebuilt in memory as agents come online and start collecting cable data
7. **Service resumed** - Cable validation monitoring continues with minimal interruption

The entire process is fully automated, requiring no manual intervention to restore service after a failover event. While the topology configuration and historical data are immediately available from shared storage, the runtime validation state will be progressively rebuilt as agents reconnect and report their status.

Important Considerations

- **Shared Storage**: Ensure that the shared storage is reliable and accessible from both nodes with low latency

- **Network Configuration**: Both nodes should have similar network configurations to ensure agents can communicate with the collector regardless of which node is active
- **Cluster Management**: Use a proper HA cluster management solution to handle failover detection and node transitions
- **Testing**: Regularly test failover scenarios to ensure the configuration is working as expected
- **Monitoring**: Implement monitoring to detect failover events and verify that validation resumes successfully on the standby node

Summary

Together, the `STARTUP_TOPOLOGY=last` and `AUTO_START_VALIDATION=true` configuration settings create a robust failover mechanism where the standby collector can automatically take over validation operations in an active-passive HA deployment. This configuration ensures minimal disruption to network monitoring and allows HA deployments to provide the high reliability that production environments demand, without requiring manual intervention during failover events.

SSH Configuration and Usage in Agent Deployment/Uninstall

The Cable Validation Tool (CVT) uses SSH for deploying and managing agents on Linux-based devices (hosts and switches). This document provides verification and QA teams with comprehensive guidance on SSH configuration, testing procedures, troubleshooting, and validation criteria for agent deployment and uninstall operations.

SSH Architecture

System Components

The CVT system uses multiple SSH components to handle different types of device connections:

1. **SSH Connection Management**
    - Base SSH client for establishing secure connections

    - Linux-specific SSH client for command execution on hosts and Linux switches
    - SFTP client for secure file transfers during deployment
    - Specialized client for MLNX-OS switch communication

2. **Agent Deployment System**
    - Linux agent deployment handler for hosts and Linux switches
    - MLNX-OS agent deployment handler for Mellanox switches

Device Support Matrix

Device Type	OS Type	SSH Usage	Authentication
Host	Linux	SSH + SFTP	Password or SSH Keys
Switch	Linux (Cumulus, NVOS)	SSH + SFTP	Password (required)
Switch	MLNX-OS	JSON API (not SSH)	Password only

Important Notes:

- SSH is NOT used for MLNX-OS switches - they use JSON API over HTTP/HTTPS
- For switches (including Linux switches), password authentication is **required** as the agent uses these credentials to communicate with the switch for port information retrieval
- Supported Linux switch operating systems: Cumulus Linux, NVOS (for NVLink and XDR switches)
- SONiC is not currently supported

SSH Configuration

Environment Variables

Configure SSH behavior using these environment variables in `/etc/cablevalidation/cvt_env.conf`:

```
[ssh]

# SSH private key file path for HOST devices only

# NOTE: SSH keys are NOT used for switch devices (switches require password authentication)

# Switch passwords are mandatory as agents use them to communicate with switches for port information

# Path must be accessible inside the collector container

CV_SSH_KEY_FILE=

# SSH connection timeout in seconds (default: 20)

# Applied to both SSH commands and SFTP transfers

SSH_CONN_TIMEOUT=20

# Enable automatic SSH key discovery (default: true)

# Only applies to HOST devices when no password is provided

# Searches: SSH agent, ~/.ssh/id_rsa, ~/.ssh/id_dsa, ~/.ssh/id_ecdsa, ~/.ssh/id_ed25519

SSH_LOOK_FOR_KEYS=true
```

Key Configuration Details

1. **CV_SSH_KEY_FILE**
    - Only used for HOST devices (switches always require passwords)
    - Must be a container-accessible path
    - Leave empty for automatic key discovery
    - Switch devices cannot use SSH keys due to agent communication requirements
2. **SSH_CONN_TIMEOUT**

    - Connection timeout in seconds
    - Applies to both SSH commands and SFTP transfers
    - Increase for slow networks, decrease for faster failure detection

3. **SSH_LOOK_FOR_KEYS**
    - Only affects HOST devices (not applicable to switches)
    - When enabled, searches standard SSH key locations
    - Only used when no password is provided for hosts

Authentication Methods

1. Password Authentication

- **Supported**: All Linux devices (hosts and switches)
- **Configuration**: Set credentials using CVT credential management
- **Usage**: **MANDATORY for all switches** (required for agent communication with switch for port information)
- **Usage**: Optional for hosts (can use SSH keys instead)

2. SSH Key Authentication

- **Supported**: HOST devices only (NOT supported for switches)
- **Configuration**: Set `CV_SSH_KEY_FILE` or enable `SSH_LOOK_FOR_KEYS`
- **Usage**: Available for hosts only
- **Switch Limitation**: Switches cannot use SSH keys because deployed agents need password credentials to communicate with the switch OS for retrieving port information

Authentication Priority (for hosts only)

1. SSH key authentication (if key available and no password set)
2. Password authentication (if password provided)
3. Automatic key discovery (if `SSH_LOOK_FOR_KEYS=true` and no password)

Switch Authentication Requirements

- **All switch types require password authentication**
- **SSH keys are not supported for switches**
- **Passwords are used by agents for ongoing switch communication**
- **Supported switch OS types**: Cumulus Linux, NVOS (NVLink/XDR switches)
- **Not supported**: SONiC (not currently supported)

Agent Deployment Process

Linux Devices Deployment Flow

1. **Preparation Phase**
    - System generates deployment script from template
    - Script customized with environment-specific values (image URLs, checksums, configuration)
    - Temporary deployment file created for transfer
2. **File Transfer Phase (SFTP)**
    - Secure connection established to target device

    - Deployment script uploaded to `/tmp` directory on target device
    - Connection closed after successful transfer

3. **Execution Phase (SSH)**
    - SSH connection established for command execution
    - Deployment script executed with elevated privileges
    - Cleanup commands remove temporary files
    - Connection closed after completion
4. **Validation Phase**
    - Deployment results logged and validated
    - Temporary files removed from both systems
    - Success/failure status reported

Deployment Script Features

The `install_agent.sh` script handles:

- Architecture detection (x86_64, aarch64)
- Docker prerequisite checks
- Image download with checksum verification
- Container deployment with appropriate parameters
- GPU support detection (for specific hardware)
- LLDP socket mounting (for ethernet monitoring)
- Comprehensive logging to `/var/log/cvt_deployment.log`

Agent Uninstall Process

Linux Devices Uninstall Flow

1. **Preparation**
    - Generate uninstall script from template (`uninstall_agent.sh`)
    - Create temporary uninstall file
2. **File Transfer and Execution**
    - Same SFTP upload process as deployment
    - Execute uninstall script with sudo privileges
3. **Uninstall Operations**
    - Container shutdown and removal
    - Docker image cleanup
    - System resource cleanup

Credential Management

Setting Credentials

The CVT system supports multiple levels of credential configuration:

Default Credentials

- Configure default username/password for all switches (password required)
- Configure default username/password for all hosts (password can be empty if using SSH keys)
- Applied when no specific credentials are found

Node-Specific Credentials

- Set unique credentials for individual devices
- Override default credentials for specific IP addresses
- For hosts: password can be empty when using SSH key authentication
- For switches: password is always required
- Highest priority in credential resolution

Credential Profiles

- Group devices with common credentials
- Assign profile names to device groups
- Manage credentials for multiple devices centrally
- Same password rules apply: switches require passwords, hosts can use empty passwords with SSH keys

Credential Priority

1. Node-specific credentials
2. Credential profile credentials (if assigned)
3. Default credentials for device type

Troubleshooting

Common SSH Issues

1. **Authentication Failures**

   ```
   SSH Authentication failure: please check device credentials
   ```

   - Verify credentials in CVT credential management

   - Check if SSH keys are properly configured for hosts
   - Ensure SSH service is running on target device

2. **Connection Timeouts**

   ```
   Failed to execute commands on node: <IP>: Connection timeout
   ```

   - Increase `SSH_CONN_TIMEOUT` value
   - Check network connectivity to target device
   - Verify firewall rules allow SSH (port 22)

3. **Permission Denied**

   ```
   Failed to execute commands on node: <IP>: Permission denied
   ```

   - Verify sudo access for the user account
   - Check if password is required for sudo
   - Ensure user has Docker access permissions

4. **File Transfer Failures**

   ```
   Failed to upload deployment file
   ```

   - Check SFTP connectivity
   - Verify write permissions to `/tmp` directory
   - Ensure sufficient disk space on target device

SSH Key Issues

1. **Key File Not Found**
   - Verify `CV_SSH_KEY_FILE` path is accessible in container
   - Check file permissions (should be 600 or 400)

  - Ensure key file is mounted into container if using Docker volumes

2. **Key Format Issues**
   - Ensure key is in OpenSSH format (not PuTTY or other formats)
   - Verify key format compatibility with paramiko SSH library
   - Check for proper key file structure and encoding

3. **Key Permission Problems**
   - Verify SSH key file permissions are restrictive (600 or 400)
   - Ensure correct ownership of key files
   - Check that key files are readable by the CVT process

Deployment Script Issues

1. **Docker Not Available**

   ```
   docker is not installed, it is required for running the agent
   ```

   - Install Docker on target device
   - Ensure Docker service is running
   - Add user to docker group if needed

2. **Image Download Failures**

   ```
   Failed to fetch the image from server
   ```

   - Check network connectivity to image server
   - Verify image URL is accessible
   - Check firewall rules for HTTP/HTTPS traffic

3. **Checksum Verification Failures**

`Checksum verification failed!`

  - Image may be corrupted during download
  - Network issues during transfer
  - Script will automatically retry download

Debugging Steps

1. **Enable Debug Logging**
    - Check deployment logs: `/var/log/cvt_deployment.log` on target device
    - Review CVT collector logs for SSH connection details
2. **Manual SSH Testing**
    - Test SSH connectivity with specified timeout values
    - Verify SFTP connectivity for file transfer operations
    - Test with specific SSH keys when configured
    - Validate authentication methods work as expected
3. **Network Connectivity Testing**
    - Verify basic network connectivity to target devices
    - Test SSH port accessibility (default port 22)
    - Check for firewall or network restrictions
    - Validate network latency and timeout settings

Best Practices

Security

1. **SSH Key Management**
    - Use dedicated SSH keys for CVT operations
    - Rotate keys regularly
    - Restrict key access with proper file permissions
    - Consider using SSH agent forwarding in containers
2. **Credential Security**
    - Use strong passwords
    - Implement credential rotation policies
    - Use credential profiles for device groups
    - Store credentials securely (CVT encrypts stored credentials)
3. **Network Security**
    - Use SSH key authentication when possible
    - Implement network segmentation
    - Configure firewall rules appropriately
    - Consider using SSH jump hosts for isolated networks

Performance

1. **Connection Management**
    - Adjust `SSH_CONN_TIMEOUT` based on network conditions
    - Use parallel deployment for multiple devices
    - Monitor deployment worker limits
2. **Resource Management**

  - Ensure sufficient bandwidth for image transfers
  - Monitor disk space on target devices
  - Clean up temporary files after deployment

Operational

1. **Monitoring**
   - Monitor deployment success rates
   - Track authentication failures
   - Review deployment logs regularly
2. **Documentation**
   - Maintain inventory of SSH keys and their usage
   - Document credential profiles and their assignments
   - Keep network topology documentation updated

Container Considerations

When running CVT in containers:

1. **SSH Key Access**
   - Mount SSH keys into container using volume mapping
   - Configure `CV_SSH_KEY_FILE` to point to container-accessible path
   - Verify key file permissions and ownership within container
2. **Network Access**
   - Ensure container can reach target devices

    - Configure network settings for direct device access
    - Verify container networking doesn't block SSH connections
3. **SSH Agent**
    - Forward SSH agent for key-based authentication
    - Configure agent socket mounting for container access
    - Verify agent accessibility within container environment

Cluster Sizing Guide

Overview

This sizing guide provides hardware and network recommendations for Cable Validation deployments based on cluster size. Recommendations are based on performance analysis of enterprise deployments and optimal resource utilization patterns.

Important Note: Cable Validation Tool (CVT) handles both **switches and hosts** in modern deployments. The legacy naming in the codebase (e.g., "SwitchAgentMgr", "switch_ip") reflects historical origins when CVT only handled switches, but now applies to all managed devices (switches, hosts, HCAs, etc.).

Sizing Methodology

Key Factors:

- **Device Overload Threshold**: Individual devices (switches/hosts) can handle ~5-10 concurrent REST API calls
- **Network Bandwidth**: 10G MGMT interface provides ~800-900 MB/s practical throughput
- **CPU Utilization**: Target 15-25 load average for optimal performance
- **Memory Requirements**: ~50-100 MB per 1000 devices for topology and batch processing
- **Batch Processing**: Optimal batch sizes scale with worker count

- **Mixed Workloads**: Switches and hosts may have different response characteristics

Quick Configuration Reference

▯ Simple 3-Variable Configuration

CVT performance can be optimized with just three environment variables:

```
# 1. Agent Deployment (~200MB image + local container operations)

export CVT_DEPLOYMENT_MAX_WORKERS=60

# 2. Everything Else (validation, connectivity, DNS, etc.)

export CVT_MAX_WORKERS=150

# 3. Batching Control (when to split large deployments)

export CVT_BATCHING_THRESHOLD=10000
```

Note: Agent deployment includes multiple phases: image fetch (~200MB), save to disk, load image, and container creation. Higher worker counts are possible because the process isn't purely bandwidth-limited.

See the Simple Tuning Guide for detailed configuration guidance.

Cluster Sizing Table

Cluster Size	Recommended CPUs	Recommended Memory	Recommended MAX_WORKERS	DEPLOYMENT_MAX_WO
Small Clusters (1-1,000 devices)				
100 devices	4-8 cores	4-8 GB	30-50	20
500 devices	8-16 cores	8-16 GB	50-75	20-40

Cluster Size	**Recommended CPUs**	**Recommended Memory**	**Recommended MAX_WORKERS**	**DEPLOYMENT_MAX_WO**
1,000 devices	16-32 cores	16-32 GB	50-100	20-40
Medium Clusters (1,000-10,000 devices)				
2,500 devices	32-64 cores	32-64 GB	75-100	40
5,000 devices	64-128 cores	64-128 GB	100-150	40-60
7,500 devices	96-192 cores	96-192 GB	125-175	60
10,000 devices	128-256 cores	128-256 GB	150-200	60
Large Clusters (10,000-25,000 devices)				
15,000 devices	192-384 cores	192-384 GB	175-225	60-80
20,000 devices	256-512 cores	256-512 GB	200-250	80
25,000 devices	320-640 cores	320-640 GB	225-275	80-100
Hyperscale Clusters (25,000+ devices)				
30,000 devices	384-768 cores	384-768 GB	250-300	100-120
35,000 devices	448-896 cores	448-896 GB	275-325	120-140

Cluster Size	Recommended CPUs	Recommended Memory	Recommended MAX_WORKERS	DEPLOYMENT_MAX_WO
40,000 devices	512-1024 cores	512 GB-1TB	300-350	140-160

Detailed Recommendations by Scale

Small Clusters (1-1,000 devices)

Characteristics:

- Single server deployment
- Basic network infrastructure
- Development/test environments
- **Device Mix**: Primarily switches, some hosts/HCAs

Sizing Logic:

- **CPU**: 1 core per 25-50 devices
- **Memory**: 10-20 MB per device for topology data
- **Workers**: Conservative scaling to avoid device overload
- **Network**: 1G sufficient for small clusters

Medium Clusters (1,000-10,000 devices)

Characteristics:

- Production deployments
- 10G management networks
- Regional or multi-site deployments
- **Device Mix**: Mixed switches and hosts, HCAs in compute clusters

Sizing Logic:

- **CPU**: 1 core per 40-80 devices (better efficiency at scale)
- **Memory**: 8-15 MB per device (shared topology data)
- **Workers**: Balanced scaling considering device capacity
- **Network**: 10G required for concurrent processing
- **Host Considerations**: Hosts may respond differently than switches

Large Clusters (10,000-25,000 devices)

Characteristics:

- Enterprise-scale deployments
- High-performance requirements
- 25G+ management networks
- **Device Mix**: Large numbers of compute hosts + infrastructure switches

Sizing Logic:

- **CPU**: 1 core per 60-100 devices (enterprise efficiency)
- **Memory**: 5-12 MB per device (optimized topology handling)
- **Workers**: Approaching device overload thresholds
- **Network**: 25G+ to handle concurrent load
- **Mixed Response**: Account for different device response characteristics

Hyperscale Clusters (25,000+ devices)

Characteristics:

- Massive datacenter deployments

- Enterprise-grade hardware (like customer's 448-core server)
- 40G+ management networks
- **Device Mix**: Thousands of compute hosts + infrastructure switches

Sizing Logic:

- **CPU**: 1 core per 80-120 devices (maximum efficiency)
- **Memory**: 3-10 MB per device (highly optimized)
- **Workers**: At or near device overload limits
- **Network**: 40G+ essential for performance
- **Device Diversity**: Must handle switches, hosts, HCAs, storage devices

Network Bandwidth Analysis

Bandwidth Requirements by Cluster Size

Cluster Size	Concurrent Workers	Peak Bandwidth Required	Network Recommendation	Device Types
1,000	50 workers	~50-100 MB/s	1G (sufficient)	Switches + some hosts
5,000	125 workers	~200-400 MB/s	1G (tight) / 10G (recommended)	Mixed switches/hosts
10,000	175 workers	~400-700 MB/s	10G (required)	Balanced switches/hosts
25,000	250 workers	~600-900 MB/s	10G (tight) / 25G (recommended)	Majority hosts + switches
40,000	350 workers	~800-1200 MB/s	25G (minimum) / 40G (optimal)	Large compute + storage

Bandwidth Calculation Logic:

- **Per Worker**: ~2-4 MB/s during active validation startup
- **Peak Usage**: During initial topology push to all devices
- **Sustained Usage**: Much lower during normal validation operation
- **Burst Patterns**: High bandwidth during startup, lower during monitoring

Key Insights:

- **Bandwidth is BURSTY**: High during startup, low during validation
- **10G Limit**: Starts getting tight around 10,000 devices
- **25G Sweet Spot**: Good performance for 25,000-40,000 devices
- **40G Future-Proof**: Optimal for large hyperscale deployments

Memory Usage Patterns

Memory Breakdown by Component

Component	Memory per 1000 Devices	Notes
Topology Data	20-40 MB	Device definitions, links, mixed switches/hosts
Batch Processing	15-30 MB	Temporary data during processing
Connection Pools	5-10 MB	HTTP session management
Results Storage	10-20 MB	Validation results and reports
Device Metadata	5-15 MB	Host-specific data, HCA mappings
Total	**55-115 MB**	**Per 1000 devices (switches + hosts)**

Performance Optimization Guidelines

Quick Performance Tuning

For detailed tuning instructions and troubleshooting, see the Simple Tuning Guide which provides:

- Easy-to-follow configuration decisions
- Monitoring guidance and success criteria
- Troubleshooting common issues
- System tuning for large deployments

CPU Optimization

- **Target Load**: 15-25 average load during processing
- **NUMA Awareness**: Use dual-socket servers for 20,000+ devices
- **Worker Scaling**: Adjust `CVT_MAX_WORKERS` based on CPU cores and observed load
- **Device Mix**: Account for different CPU requirements of switches vs hosts
- **Monitoring**: If load stays low, increase workers; if timeout errors increase, reduce workers

Memory Optimization

- **Batching**: Use `CVT_BATCHING_THRESHOLD` to control memory usage on large deployments
- **Connection Pooling**: Automatically scales with worker count
- **Garbage Collection**: Monitor for large deployments (20,000+ devices)
- **Device Metadata**: Additional memory for host-specific data (HCA mappings, etc.)

Network Optimization

- **Bandwidth Planning**: Set `CVT_DEPLOYMENT_MAX_WORKERS` based on management network capacity
- **Connection Reuse**: Essential for large deployments (handled automatically)
- **Bandwidth Monitoring**: Watch for saturation at scale
- **Device Response Variance**: Hosts may respond differently than switches
- **Burst Patterns**: High bandwidth during startup, lower during validation operation

Device Type Considerations

Switches vs Hosts Performance Characteristics

Device Type	Typical Response Time	Concurrent Call Limit	Special Considerations
Network Switches	1-3 seconds	5-10 concurrent	REST API on switch OS
Compute Hosts	2-5 seconds	3-8 concurrent	Agent on host OS, may be busier
Storage Devices	1-4 seconds	5-12 concurrent	Usually dedicated management
HCA Devices	1-2 seconds	8-15 concurrent	Lightweight agent

Example Configurations

Small Deployment (<1,000 devices)

```
# Network: 1G management interface

# Server: 8-32 cores

export CVT_DEPLOYMENT_MAX_WORKERS=20
export CVT_MAX_WORKERS=50

# CVT_BATCHING_THRESHOLD=10000 (default, no need to change)
```

Medium Deployment (1,000-10,000 devices)

```
# Network: 10G management interface

# Server: 64-128 cores

export CVT_DEPLOYMENT_MAX_WORKERS=40

export CVT_MAX_WORKERS=100

# CVT_BATCHING_THRESHOLD=10000 (default, no need to change)
```

Large Deployment (10,000-30,000 devices)

```
# Network: 25G+ management interface

# Server: 192-384 cores

export CVT_DEPLOYMENT_MAX_WORKERS=60

export CVT_MAX_WORKERS=150

export CVT_BATCHING_THRESHOLD=5000
```

Hyperscale Deployment (30,000+ devices)

```
# Network: 40G+ management interface

# Server: 448+ cores

export CVT_DEPLOYMENT_MAX_WORKERS=80

export CVT_MAX_WORKERS=200

export CVT_BATCHING_THRESHOLD=3000
```

Configuration Guidelines

CVT_DEPLOYMENT_MAX_WORKERS (Agent Deployment):

- Based on network bandwidth and local container operations (~200MB per device image)
- Deployment includes: image fetch, save to disk, load image, container creation
- 1G network: 20 workers (~4GB concurrent + local ops)
- 10G network: 60 workers (~12GB concurrent + local ops)
- 25G+ network: 100-160 workers (higher bandwidth + parallel local operations)

CVT_MAX_WORKERS (Validation Operations):

- Based on server CPU cores and device capacity
- 8-32 cores: 30-50 workers
- 32-128 cores: 75-150 workers
- 128+ cores: 150-300 workers
- Watch for device timeout errors and reduce if needed

CVT_BATCHING_THRESHOLD (Batch Processing):

- <5,000 devices: 10000 (default, single batch)
- 5,000-20,000 devices: 5000 (light batching)
- 20,000+ devices: 3000 (aggressive batching)

Scaling Considerations

Vertical Scaling Limits

- **Single Server**: Effective up to ~40,000 devices (switches + hosts)

- **CPU Bound**: Beyond 40,000 devices, consider distributed processing
- **Memory Bound**: Rarely an issue with modern servers (hosts require slightly more memory)
- **Network Bound**: Primary constraint for large deployments
- **Device Mix**: Higher host percentage may require more resources

Horizontal Scaling Options

- **Multiple Collectors**: Split clusters across multiple servers
- **Geographic Distribution**: Regional collectors for global deployments
- **Load Balancing**: Distribute devices across multiple validation instances

Device Mix Impact on Sizing

Typical Cluster Compositions

Cluster Type	Switches %	Hosts %	Notes	Sizing Impact
Infrastructure-Heavy	80%	20%	Network-focused deployment	Lower memory, higher network load
Compute-Heavy	30%	70%	HPC/AI clusters	Higher memory, variable response times
Balanced	50%	50%	Mixed enterprise deployment	Standard sizing applies
Storage-Heavy	40%	60%	Storage clusters with many storage hosts	Higher memory, faster responses

Sizing Adjustments by Device Mix

Infrastructure-Heavy Clusters (80% switches):

- **CPU**: Use lower end of range
- **Memory**: Use lower end of range
- **Workers**: Can be more aggressive
- **Network**: Higher bandwidth needs per device

Compute-Heavy Clusters (70% hosts):

- **CPU**: Use higher end of range
- **Memory**: Use higher end of range (HCA mappings, host metadata)
- **Workers**: More conservative (hosts may be busier)
- **Network**: Variable load patterns

Balanced Clusters (50/50 mix):

- **CPU**: Use middle of range
- **Memory**: Use middle of range
- **Workers**: Standard recommendations apply
- **Network**: Standard bandwidth planning

Monitoring and Alerting

Key Metrics to Monitor

1. **Server Load**: Target 15-25 during processing
2. **Memory Usage**: Should stay well below allocated
3. **Network Utilization**: Watch for bandwidth saturation (especially during agent deployment)
4. **Device Response Times**: Primary performance indicator

5. **Error Rates**: Timeout and connection errors
6. **Validation Completion Time**: Compare against expected times in sizing table

Success Indicators

Good Signs (can increase `CVT_MAX_WORKERS`):

- ▯ Server load increases during validation (better CPU utilization)
- ▯ Validation completes faster than baseline
- ▯ No significant increase in timeout errors
- ▯ Network bandwidth stays below 80%

Warning Signs (reduce `CVT_MAX_WORKERS`):

- ▯ Many "Timeout while trying to start validation" errors
- ▯ "Connection refused" errors from devices
- ▯ Server load stays low (underutilization)
- ▯ Network bandwidth hits 90%+

Scaling Triggers

- **Scale Up Workers**: Load < 10, low error rates, fast completion
- **Scale Down Workers**: High error rates, device timeouts, network saturation
- **Increase Deployment Workers**: Network utilization < 50% during agent deployment
- **Decrease Deployment Workers**: Network bandwidth > 80% during agent deployment
- **Adjust Batching**: Memory usage > 80% (reduce `CVT_BATCHING_THRESHOLD`)

Performance Expectations by Cluster Size

Cluster Size	Expected Validation Time	Expected Load Average	Target Worker Count
1,000 devices	1-3 minutes	8-12	50-100
5,000 devices	3-5 minutes	12-18	100-150
10,000 devices	5-8 minutes	15-22	150-200
25,000 devices	10-15 minutes	18-25	225-275
40,000 devices	18-25 minutes	20-28	300-350

Document Relationship

This **Cluster Sizing Guide** provides comprehensive hardware and infrastructure planning:

- CPU, memory, and network capacity planning
- Expected performance at different scales
- Device type considerations (switches, hosts, HCAs)
- Detailed sizing methodology

For **day-to-day performance tuning**, refer to the Simple Tuning Guide:

- Simple 3-variable configuration
- Quick-start configurations by deployment size
- Troubleshooting and monitoring guidance
- Practical tuning adjustments

Simple CVT Performance Tuning Guide

Easy Configuration - Just 3 Variables

Based on extensive testing and real-world deployments, CVT performance can be optimized with just these simple settings:

Core Configuration:

```
# 1. Agent Deployment (~200MB image + local container operations)

export CVT_DEPLOYMENT_MAX_WORKERS=60

# 2. Everything Else (validation, connectivity, DNS, etc.)

export CVT_MAX_WORKERS=150

# 3. Batching Control (when to split large deployments)

export CVT_BATCHING_THRESHOLD=10000
```

How to Decide Values

CVT_DEPLOYMENT_MAX_WORKERS (Agent Deployment)

Question: How much bandwidth can your management network handle?

Note: Deployment includes multiple phases: image fetch (~200MB), save to disk, load image, and container creation. The process is not purely bandwidth-limited, so we can use higher worker counts.

Network Speed	Recommended Value	Reasoning
1G	20	20 × 200MB = ~4GB concurrent + local ops
10G	60	60 × 200MB = ~12GB concurrent + local ops
25G+	100-160	Higher bandwidth + parallel local operations

CVT_MAX_WORKERS (Everything Else)

Question: How many CPU cores does your server have?

Server CPU Cores	Recommended Value	Reasoning
8-32 cores	30-50	Conservative scaling
32-128 cores	75-150	Balanced scaling
128+ cores	150-300	Aggressive scaling

But watch for switch overload! If you see many timeout errors, reduce this value.

CVT_BATCHING_THRESHOLD (When to Batch)

Question: How many switches do you have?

Switch Count	Recommended Value	What Happens
<5,000	10000 (default)	Single batch (faster)
5,000-20,000	5000	Light batching
20,000+	3000	More aggressive batching

Quick Start by Deployment Size:

Small Deployment (<1,000 switches)

```
export CVT_DEPLOYMENT_MAX_WORKERS=20

export CVT_MAX_WORKERS=50

# No need to change batching threshold
```

Medium Deployment (1,000-10,000 switches)

```
export CVT_DEPLOYMENT_MAX_WORKERS=40

export CVT_MAX_WORKERS=100

# No need to change batching threshold
```

Large Deployment (10,000-30,000 switches)

```
export CVT_DEPLOYMENT_MAX_WORKERS=60

export CVT_MAX_WORKERS=150

export CVT_BATCHING_THRESHOLD=5000
```

Hyperscale Deployment (30,000+ switches)

```
export CVT_DEPLOYMENT_MAX_WORKERS=80

export CVT_MAX_WORKERS=200

export CVT_BATCHING_THRESHOLD=3000
```

How to Know If Your Settings Are Good

Good Signs:

- Server load increases during validation (better CPU utilization)
- Validation completes faster than before
- No significant increase in timeout errors
- Network bandwidth stays below 80%

Warning Signs:

- □ Many "Timeout while trying to start validation" errors
- □ "Connection refused" errors from switches
- □ Server load stays low (underutilization)
- □ Network bandwidth hits 90%+

Adjustment Strategy:

1. Too many timeouts: Reduce `CVT_MAX_WORKERS` by 25-50
2. Server underutilized: Increase `CVT_MAX_WORKERS` by 25-50
3. Deployment too slow: Increase `CVT_DEPLOYMENT_MAX_WORKERS` (if network allows)
4. Memory issues: Reduce `CVT_BATCHING_THRESHOLD`

Expected Performance

Your Customer's 32,149 Switches:

Current Configuration:

```
export CVT_DEPLOYMENT_MAX_WORKERS=60    # Optimal for 10G network with 200MB image

export CVT_MAX_WORKERS=150              # Good for 448-core server

export CVT_BATCHING_THRESHOLD=10000     # Will use batching (32K > 10K)
```

Expected Results:

- Current: 7m 13s
- Optimized: 2-3 minutes (60-75% improvement)
- Server Load: Should increase from 5-7 to 15-20

Monitoring and Troubleshooting

Critical Monitoring Points:

Watch for Device Overload:

1. Timeout Errors: Increase in "Timeout while trying to start validation" messages
2. Connection Refused: "Connection error" messages from devices
3. Response Times: Slower device response times
4. Failure Rate: Higher percentage of failed device connections

Server Utilization Monitoring:

1. Load Average: Should increase from baseline to target ranges
2. CPU Usage: Better utilization of available cores
3. Memory: Watch for any memory pressure during large deployments
4. Network: Monitor bandwidth utilization on management interface

Success Criteria by Worker Count

Worker Count	Expected Load Average	Expected Time Improvement	Risk Level
50-75	8-12	30-50% faster	Low
100-150	12-18	50-70% faster	Medium
200+	18-25	70%+ faster	High

Red Flags (Scale Back If You See)

- □ Significant increase in timeout errors
- □ "Connection refused" errors from devices

- ▯ Device response times getting slower
- ▯ Higher failure rates than baseline

Green Lights (Scale Up If You See)

- ▯ Stable or improved device response times
- ▯ No increase in connection errors
- ▯ Server load well below target range
- ▯ Good success rate maintained

System Tuning for Large Deployments

File Descriptor Limits

```
# Increase file descriptor limits for high concurrency
ulimit -n 65536
# Make permanent by adding to /etc/security/limits.conf:
echo "* soft nofile 65536" >> /etc/security/limits.conf
echo "* hard nofile 65536" >> /etc/security/limits.conf
```

Network Optimization

```
# Optimize network settings for high concurrency
echo 65536 > /proc/sys/net/core/somaxconn
echo 1 > /proc/sys/net/ipv4/tcp_tw_reuse
echo 1 > /proc/sys/net/ipv4/tcp_tw_recycle
```

Memory Settings

# For very large deployments (20,000+ devices)

echo 1 > /proc/sys/vm/overcommit_memory

echo 80 > /proc/sys/vm/overcommit_ratio

Agent Deployment Time Estimates

Deployment Duration by Cluster Size:

Devices	Workers	Network	Estimated Time
1,000	20	1G	1-2 hours
4,000	60	10G	2-4 hours
10,000	80	25G	4-8 hours
25,000+	100-160	40G+	10-20 hours

Note: Agent deployment includes image fetch (~200MB per device), local save, image load, and container creation. With the reduced image size and parallel local operations, deployment is significantly faster than with larger images.

Bottom Line

Most customers only need to set 2 variables:

1. `CVT_DEPLOYMENT_MAX_WORKERS` (based on network bandwidth)
2. `CVT_MAX_WORKERS` (based on server CPU and device tolerance)

The third variable (`CVT_BATCHING_THRESHOLD`) usually works fine at default!

Cables Agent

The Cables agent is implemented as a docker container that executes on the switch to gather data on neighboring switches and link quality. The agent operates a web service capable of providing information on ports and links through user queries. Moreover, the agent transmits validation reports to the bringup server.

- Open Agent Port
- Check if Cable Agent is Running
- Collecting Amber File

Open Agent Port

The agent establishes communication with the CVT collector using **port 8251**. To ensure proper functionality, this port must be open on the switch's firewall. The process of enabling this port differs based on the type of fabric being used:

1. **InfiniBand Fabric:**

   In the InfiniBand fabric, the tool automatically opens port 8251 without requiring any manual intervention from the user. No additional configuration is necessary.

2. **Ethernet Fabric:**

   For ethernet fabric, port 8251 must be opened manually or using ZTP tools. This can be accomplished by executing the following command directly on the switch:

   ```
   nv set acl acl-default-whitelist rule 200 match ip tcp dest-port 8251
   nv set acl acl-default-whitelist type ipv4
   nv config apply
   nv config save
   ```

3. **NVOS Fabric:**

   For NVOS fabric (XDR switches), port 8251 must be opened manually. This can be accomplished by executing the following commands directly on the switch:

   ```
   nv set acl AAA type ipv4
   nv set acl AAA rule 1 match ip tcp dest-port 8251
   nv set acl AAA rule 1 action permit
   nv set interface eth0-1 acl AAA inbound control-plane
   nv config apply
   nv config save
   ```

Check if Cable Agent is Running

On InfiniBand

Check if cable agent is running on the switch:

1. Run:

   ```
   ssh admin@<switch-ip-or-name>
   ```

2. Enable
3. Show docker images
4. Exit

If cables agent is running on the switch, the following output is prompted.

```
-----------------------------------------------------------------
-----------
```

```
Image                                  Version      Created
Size
------------------------------------------------------------------
-----------
cables_agent                           latest       13 hours ago
788MB
```

On Ethernet and NVOS

Check if cable agent is running on the switch:

1. Run:

   ```
   ssh admin@<switch-ip-or-name>
   ```

2. docker ps

If cables agent is running on the switch, the following output is prompted.

```

------------------------------------------------------------------
-----------
Image                                  Version      Created
Size
------------------------------------------------------------------
-----------
cables_agent                           latest       13 hours ago
788MB
```

Collecting Amber File

Ethernet Switches, InfiniBand Switches(NDR, HDR) and Hosts

Ethernet and InfiniBand(NDR/HDR) switches have one MST device for which Amber file is collected.

Hosts have one MST device per NIC, hence Amber file is collected per mst device.

Following command can be used to collect amber file:

1. Find the MST device by running the command:

```
# sudo mst status -v
```

2. Run mlxlink command to collect the amber:

```
# mlxlink -d <mst_device_name> --amber_collect
<path_to_save_the_amber_file>
```

   Example:

```
mlxlink –d /dev/mst/mt54000_pciconf0 –amber_collect
/tmp/amber.csv
```

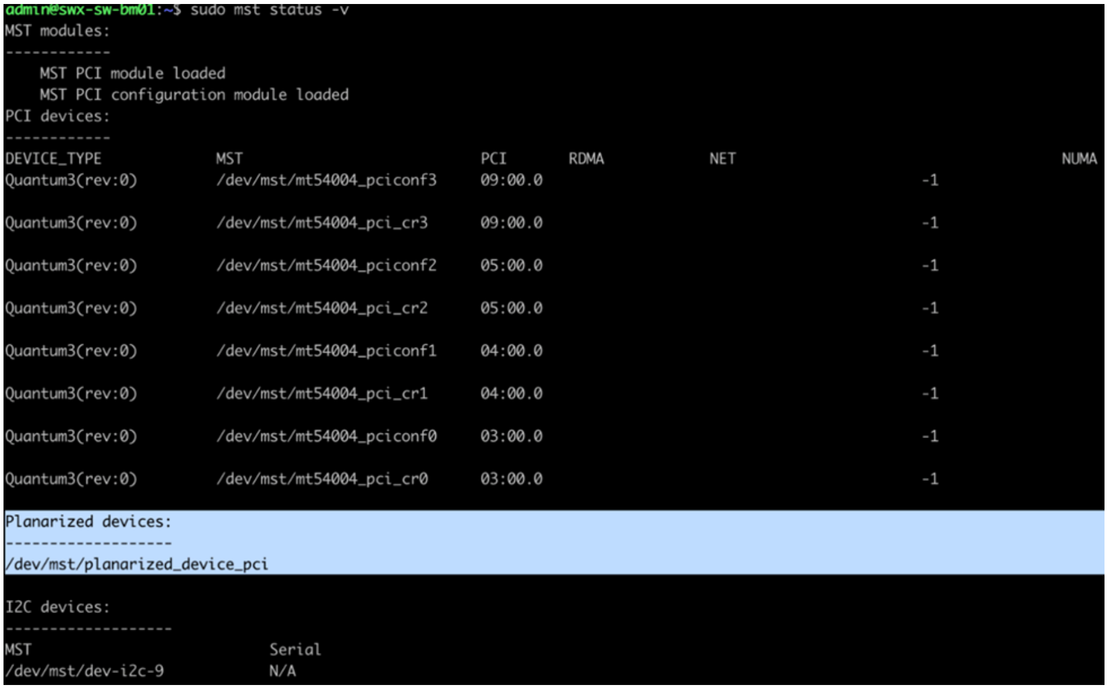
XDR Switches

On **XDR switches** there are four devices - one per ASIC.

1. To get the amber for all the ASICs, create a planarized device. To do so, run the following command:

```
# sudo mst start --planarized_device_pci
```

Expected output:



```
admin@swx-sw-bm01:~$ sudo mst status -v
MST modules:
------------
    MST PCI module loaded
    MST PCI configuration module loaded
PCI devices:
------------
DEVICE_TYPE             MST                           PCI       RDMA            NET                                     NUMA
Quantum3(rev:0)         /dev/mst/mt54004_pciconf3     09:00.0                                           -1

Quantum3(rev:0)         /dev/mst/mt54004_pci_cr3      09:00.0                                           -1

Quantum3(rev:0)         /dev/mst/mt54004_pciconf2     05:00.0                                           -1

Quantum3(rev:0)         /dev/mst/mt54004_pci_cr2      05:00.0                                           -1

Quantum3(rev:0)         /dev/mst/mt54004_pciconf1     04:00.0                                           -1

Quantum3(rev:0)         /dev/mst/mt54004_pci_cr1      04:00.0                                           -1

Quantum3(rev:0)         /dev/mst/mt54004_pciconf0     03:00.0                                           -1

Quantum3(rev:0)         /dev/mst/mt54004_pci_cr0      03:00.0                                           -1

Planarized devices:
-------------------
/dev/mst/planarized_device_pci

I2C devices:
-------------------
MST                           Serial
/dev/mst/dev-i2c-9            N/A
```

2. Run mlxlink command for the planarized device and it will generate 4 files automatically - one per asic:

```
sudo mlxlink -d /dev/mst/planarized_device_pci --
amber_collect <path_to_save_the_amber_file>
```

Example:

```
sudo mlxlink -d /dev/mst/planarized_device_pci --
amber_collect /tmp/amber.csv
```

This command creates four amber files – and labels the file name as: `amber_ASIC_0.csv, amber_ASIC_1.csv, amber_ASIC_2.csv, amber_ASIC_3.csv`

Each amber file will have information for a particular split port.

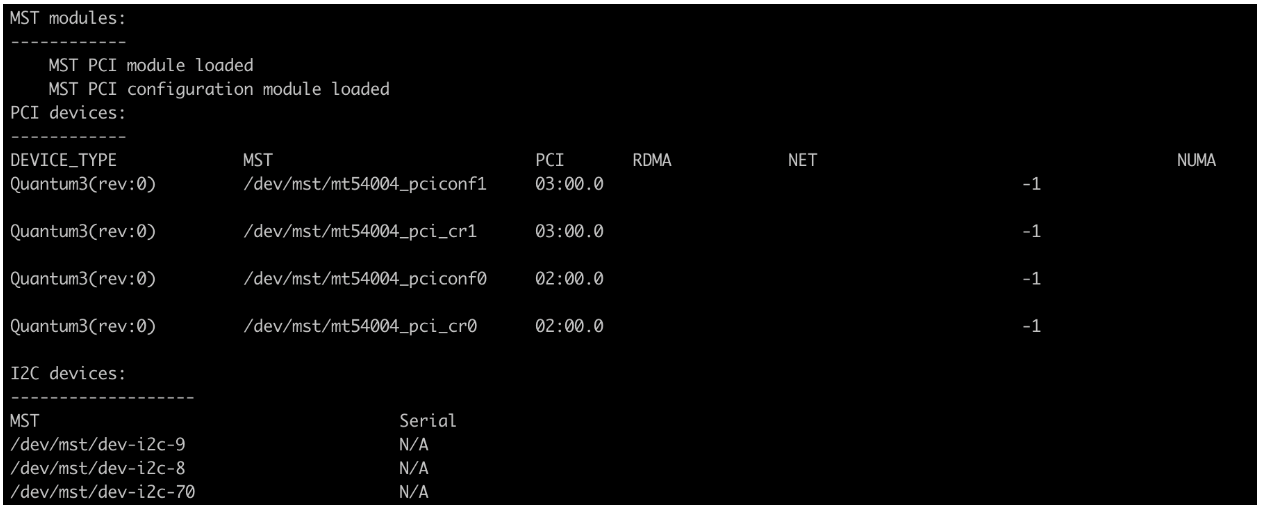
NVswitches

On NVlink Switches there are 2 mst devices - one per ASIC.

1. Find the MST devices by running the command:

```
# sudo mst status -v
```

Example:



```
MST modules:
------------
    MST PCI module loaded
    MST PCI configuration module loaded
PCI devices:
------------
DEVICE_TYPE             MST                           PCI       RDMA            NET                                     NUMA
Quantum3(rev:0)         /dev/mst/mt54004_pciconf1     03:00.0                                           -1

Quantum3(rev:0)         /dev/mst/mt54004_pci_cr1      03:00.0                                           -1

Quantum3(rev:0)         /dev/mst/mt54004_pciconf0     02:00.0                                           -1

Quantum3(rev:0)         /dev/mst/mt54004_pci_cr0      02:00.0                                           -1

I2C devices:
-------------------
MST                                     Serial
/dev/mst/dev-i2c-9                      N/A
/dev/mst/dev-i2c-8                      N/A
/dev/mst/dev-i2c-70                     N/A
```

2. Run mlxlink command to collect the amber:

```
# mlxlink -d <mst_device_name> --amber_collect
<path_to_save_the_amber_file>
```

Example:

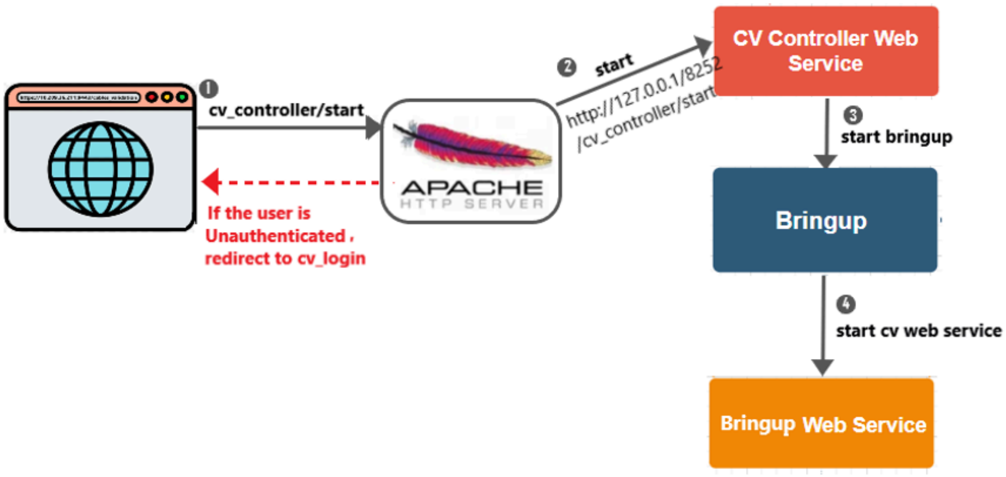
```
mlxlink –d /dev/mst/mt54004_pci_cr0 –amber_collect
/tmp/amber_asic1.csv
mlxlink -d /dev/mst/mt54004_pci_cr1 -amber_collect
/tmp/amber_asic2.csv
```

Cable Validation Controller

The CV Controller is a web service that is automatically started when the CVT container is launched via **supervisord**. It provides REST APIs to initiate and terminate the bring-up process, enhancing usability and eliminating the need for manual steps.

The CV Controller uses **Apache** to handle incoming REST API requests. Apache routes any request that includes the `cv_controller` prefix to the CV Controller service.

Authentication is handled through **session-based authentication**, meaning users must be logged in to access the service.



❶ cv_controller/start
If the user is Unauthenticated , redirect to cv_login
APACHE HTTP SERVER
❷ start http://127.0.0.1/8252 /cv_controller/start
CV Controller Web Service
❸ start bringup
Bringup
❹ start cv web service
Bringup Web Service

The CV controller service output will be saved in a new log file called **cv_controller_collector.log** under **/cable_bringup_root/log** directory.

Overriding CV Controller Port

The CV controller uses port **8252** by default. Users can modify this port by updating the **CV_CONTROLLER_PORT** variable in the **config.cfg** file.

To make this adjustment, follow these steps:

As Plugin

1. Execute **/opt/ufm/scripts/manage_ufm_plugins.sh add -p cablevalidation** to add the Cable Validation plugin.

2. Stop the plugin using **/opt/ufm/scripts/manage_ufm_plugins.sh stop -p cablevalidation**
3. Use **vim /opt/ufm/files/conf/plugins/cablevalidation/config.cfg** to modify the **' CV_CONTROLLER_PORT '** variable.
4. Update and save the file.
5. Start the plugin again with **/opt/ufm/scripts/manage_ufm_plugins.sh start -p cablevalidation**.

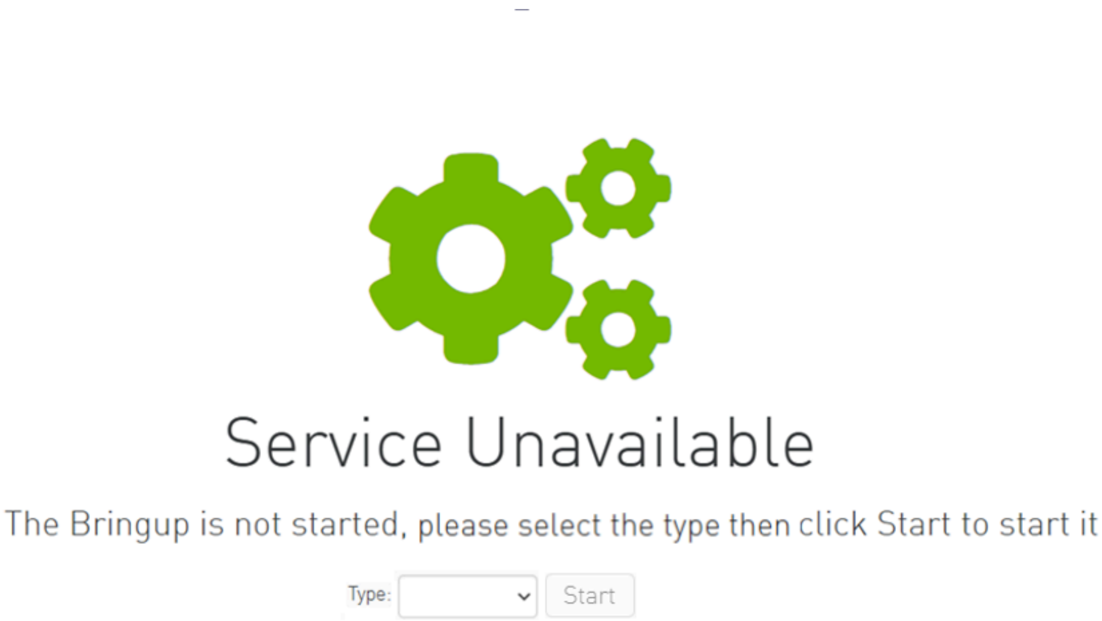
As Standalone

1. Pass the new port as Env by executing **docker run --name cables_bringup -itd --network=host --env CV_CONTROLLER_PORT=<new-port> cables_bringup**

With these changes, the new configuration will take effect, and Apache will run with the updated ports.

CV Controller GUI

When a user accesses the web page of the Cable Validation Tool, the following screen will appear **if the bring-up service has not yet started**:

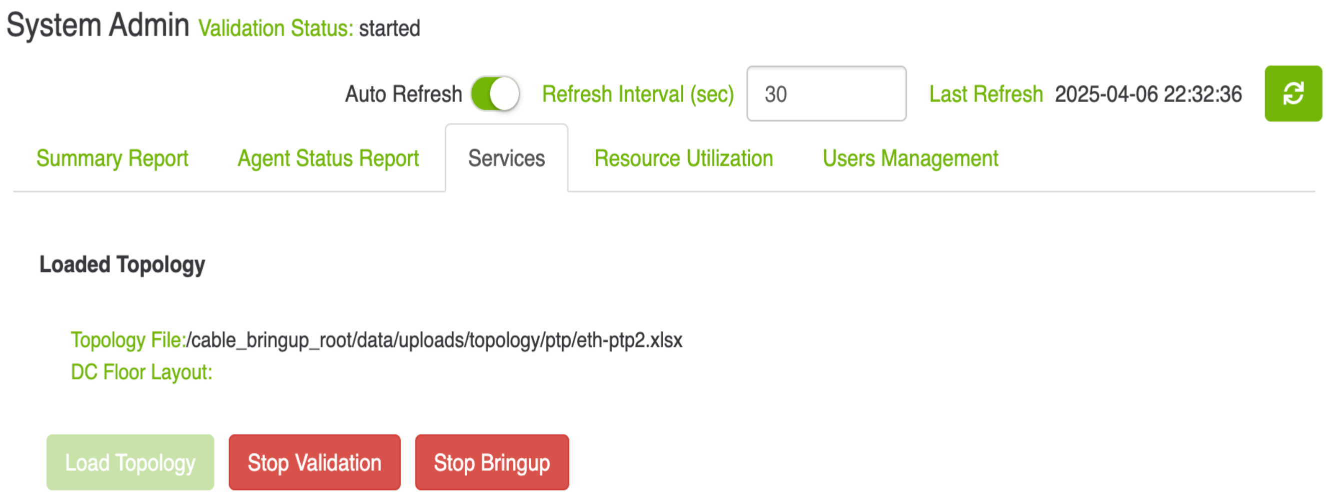


Service Unavailable

The Bringup is not started, please select the type then click Start to start it

The page displays a button to start the bring-up process, along with a drop-down menu to select the bring-up type (e.g., InfiniBand, XDR, Ethernet).

After starting the bringup service, Users can also stop the bring-up service from the **Service** tab on the **System Admin** page.



Validation Status: started

Refresh Interval (sec) 30

Last Refresh 2025-04-06 22:32:36

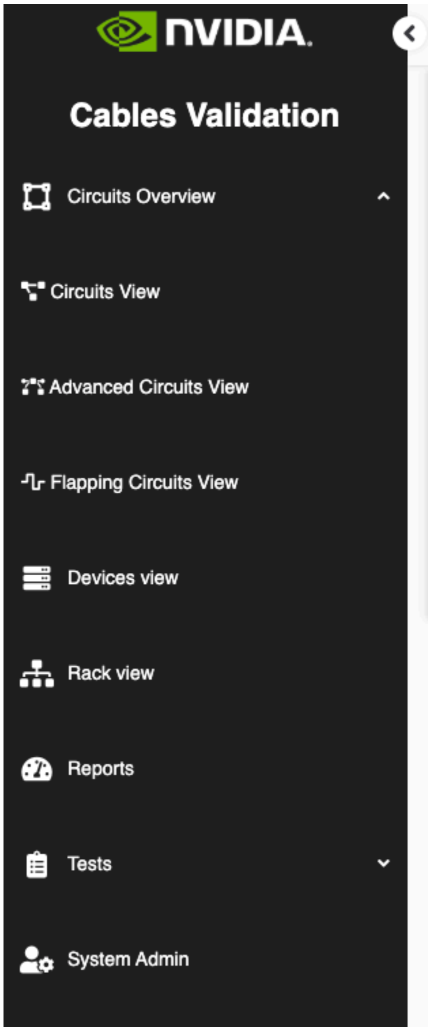
Topology File:/cable_bringup_root/data/uploads/topology/ptp/eth-ptp2.xlsx

DC Floor Layout:

Bringup GUI

The Cable Validation Tool views facilitate rapid issue identification and correlation, enabling efficient remediation.

The bring-up GUI includes the following pages:"



- **Circuits View:** Combines link data reported by both ends of a circuit (correlation).
- **Flapping Circuits View:** Flapping circuits with 24h flap history.
- **Rack View:** Issues displayed by physical location.
- **Reports:** Summary (total), list of cable issues.

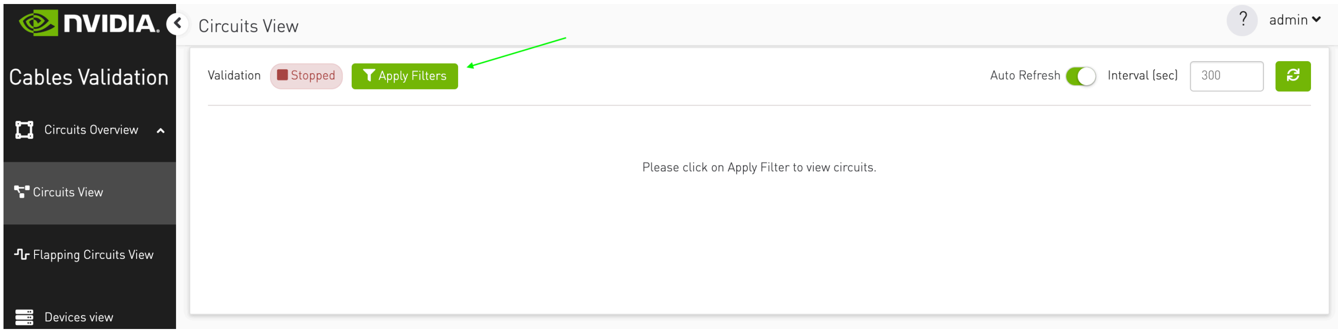
- **Golden BER Tests:** Golden amBER test (clear counters, validate compliance).
- **Amber Collection Test:** Collect Amber files.
- **Advanced Flapping Monitoring**: Monitors circuit flapping events along with BER (Raw/Eff), temperature, and Tx/Rx.
- **System Admin:** CVT tool monitoring and debugging (report delays, load topology, manage device access credentials, start/stop validation, etc.).

Circuits Overview

Circuits View

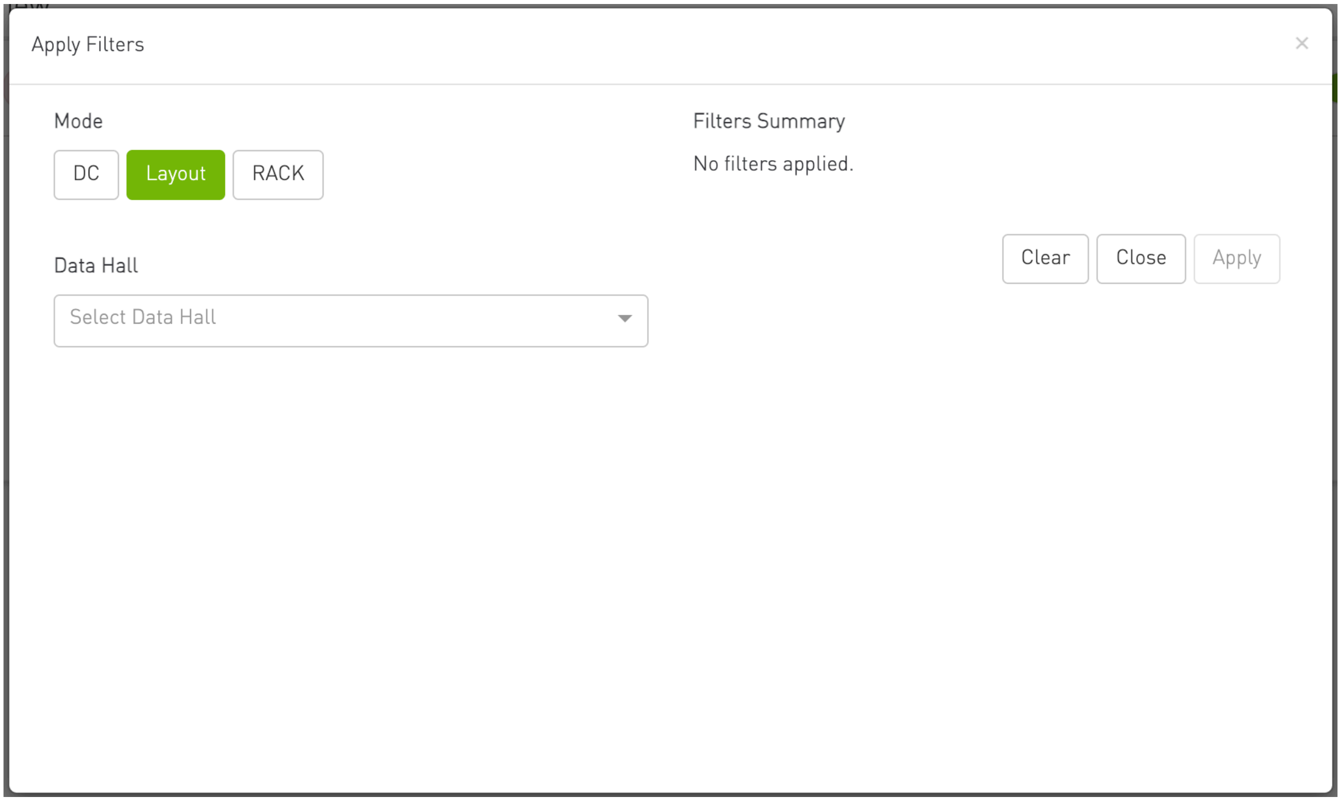
The **Circuits View** helps you explore the connection details between two endpoints, labeled **A** and **Z**.

When you open the page, you'll see a message asking you to apply a filter to view the circuits.

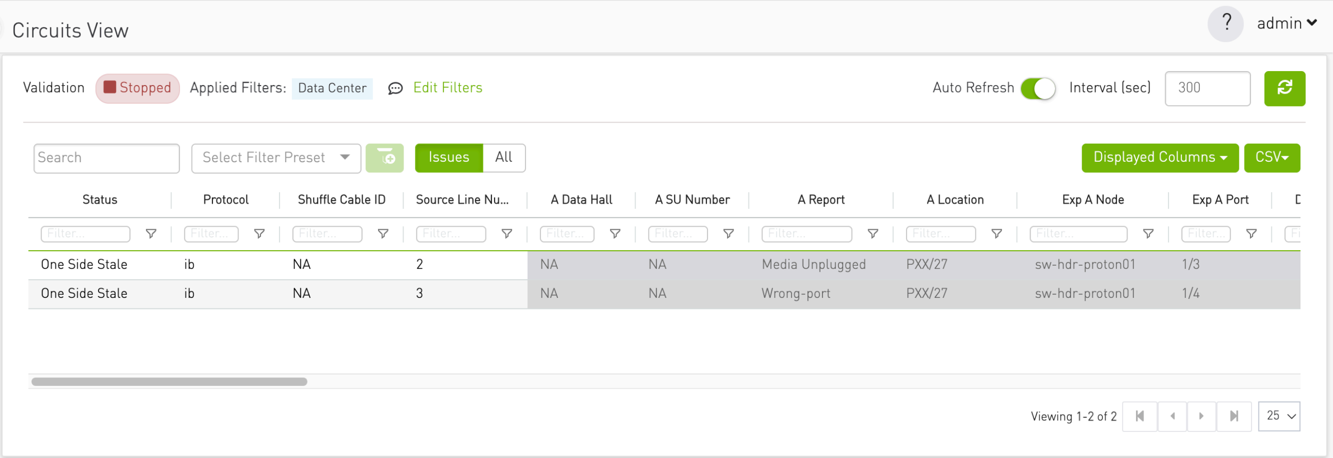


Please click on Apply Filter to view circuits.

Click **"Apply Filters"** to open the filter window, choose your preferred filter, and apply it. For more guidance, see the **Resource Filter** page.



Once your filter is applied, all matching circuits will appear in a table.



Status	Protocol	Shuffle Cable ID	Source Line Nu...	A Data Hall	A SU Number	A Report	A Location	Exp A Node	Exp A Port
One Side Stale	ib	NA	2	NA	NA	Media Unplugged	PXX/27	sw-hdr-proton01	1/3
One Side Stale	ib	NA	3	NA	NA	Wrong-port	PXX/27	sw-hdr-proton01	1/4

Each circuit includes the following information:

1. Status

The status reflects the overall health of the circuit and can have one of the following values:

- **Fail**: Indicates that there are issues with one or both endpoints.

- **Pass**: Indicates that there are no issues with either endpoint.
- **Incomplete**: Indicates no issues with the endpoints, but the port status is not yet ready.
- **One Side Stale**: One of the endpoints is stale.
- **Both Sides Stale**: Both endpoints are stale.

2. **Protocol**: circuit protocol type ib, ethernet or nvlink.

3. Shuffle Cable ID

4. Source Line Number

5. Endpoint Information

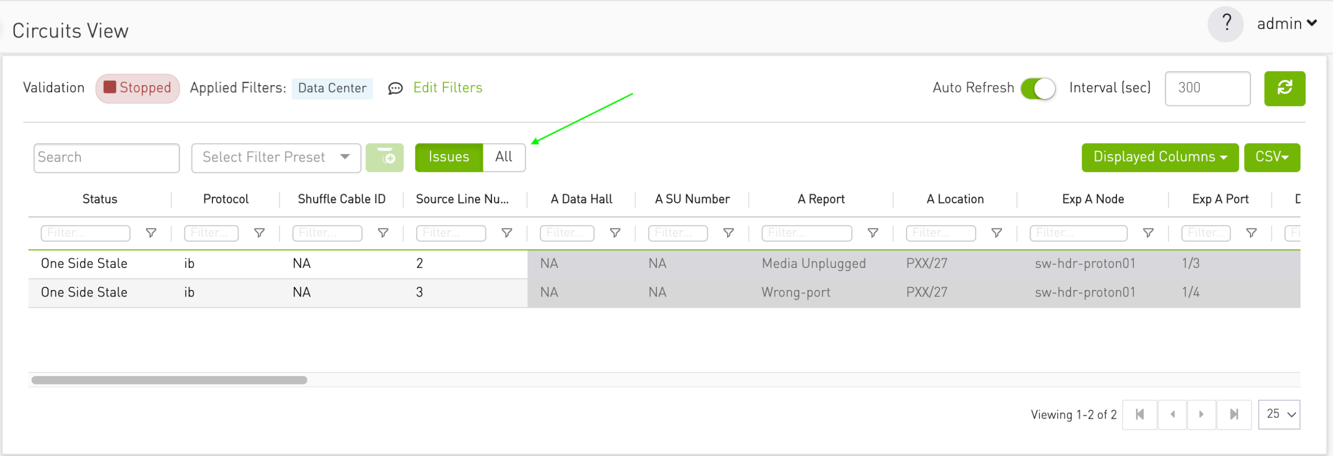
Detailed information about each endpoint includes:

- **Report**: the reported syndrome
- **Location**: Specifies the endpoint's location, Rack, and Unit.
- **Node and Port Details**: information about the nodes and the associated ports.
- **Transceiver Information**: Includes details such as:
    - Transceiver part number, hidden by default.
    - Transceiver serial number, hidden by default.
    - Firmware version, hidden by default.
    - Reinsert count, hidden by default.
    - Swap count, hidden by default.
    - Connection status: Displays **false** if one of the endpoints has a "No Transceiver" issue.
- **Port Status**: Indicates the current state of the port:
    - **Down**: If any of the following conditions are present:
        - "No Transceiver"

    - "Link Down, No Signal"
    - "ErrDisable - Flap"
    - "Admin Down"
    - "ErrDisable - Rx"
    - "Negotiation Failure"
  - **Up**: When none of the above conditions apply.
- **Signal Status**: Shows transmission and reception power levels (Tx/Rx power lanes).
- **BER Counters**: Provides bit error rate (BER) statistics, such as raw BER and effective BER., These statistics are only available to Admin users and are hidden by default.
- **Traffic Counters**: Includes traffic statistics, such as errors, drops, and byte counts for incoming and outgoing traffic.
- **Physical Port**
- **Custom NIC Name**
- **NIC OS Name**
- **RDMA Name:** hidden by default.
- **PCI Address:** hidden by default.
- **Shuffle Cable Information**:
  - A Shuffle Cable Connector
  - A Shuffle Cable MPO Connector
  - A Module Part Number
- **Recommended Action**: Suggests actions to resolve any identified issues.
- **Time Since Last Clear:** Indicates the time since the counters were last cleared.
- **Report Status**: Indicates the timeliness of the report, with the following values:

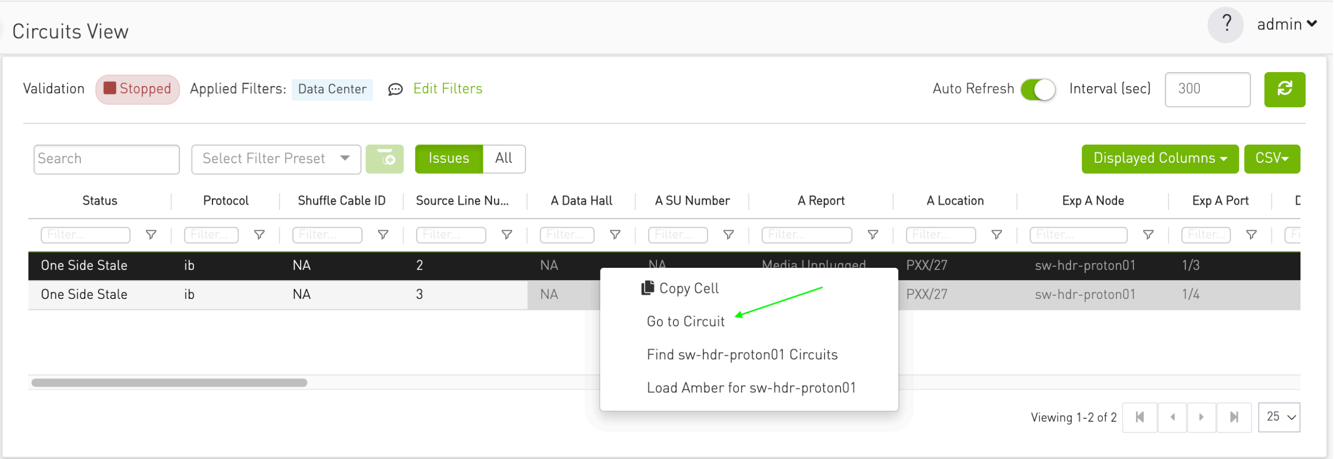
- **No Report**: The report has not been updated yet(the associated endpoint will be grayed out).
- **Stale Report**: The report has not been updated within the last 15 minutes (the associated endpoint will be grayed out).
- **Latest Report**: The report was updated within the last 15 minutes.

By default, only circuits with a **Fail** status (unhealthy circuits) are displayed. However, users can view all circuits by selecting the **All** option.



Status	Protocol	Shuffle Cable ID	Source Line Nu...	A Data Hall	A SU Number	A Report	A Location	Exp A Node	Exp A Port
One Side Stale	ib	NA	2	NA	NA	Media Unplugged	PXX/27	sw-hdr-proton01	1/3
One Side Stale	ib	NA	3	NA	NA	Wrong-port	PXX/27	sw-hdr-proton01	1/4

To select the **peer-port cable issue** user can right-click on the specified circuit and select **Go To Circuit** to see the combined circuit **details** and **remediation actions.**



Status	Protocol	Shuffle Cable ID	Source Line Nu...	A Data Hall	A SU Number	A Report	A Location	Exp A Node	Exp A Port
One Side Stale	ib	NA	2	NA	NA	Media Unplugged	PXX/27	sw-hdr-proton01	1/3
One Side Stale	ib	NA	3	NA			PXX/27	sw-hdr-proton01	1/4

Rx/Tx Power Lane

In the realm of data transmission, understanding the power values for each lane in the port is crucial for maintaining optimal performance and ensuring efficient signal transmission.

Rx/Tx Power Values for Each Lane

The Rx and Tx power values are essential metrics that provide insight into the performance of each lane in a port. These values are especially significant in high-speed data transmission environments, where maintaining the correct power levels is critical for data integrity and system reliability.

NDR Switches

For NDR switches, the power values for each lane in the port are provided, with each cage containing two ports. The relevant lanes for each port can be identified using the _Lanes_Used' counter. This counter plays a vital role in determining which lanes are active and relevant for data transmission.

The 'Module_Lanes_Used' counter is a binary indicator that specifies the active lanes in a particular module. For instance:

- If the value is '1_1_1_1_0_0_0_0', lanes 4-7 should be considered.
- If the value is '0_0_0_0_1_1_1_1', lanes 0-3 should be taken into account.

The Rx and Tx power values for these lanes are denoted as rx_power_lane_n and tx_power_lane_n, where 'n' can range from 0 to 7, depending on the active lanes.

InfiniBand Technology

When it comes to InfiniBand technology, the approach to handling Rx and Tx power values differs slightly. InfiniBand typically provides power values only for lanes 0-3. As a result, the 'Module_Lanes_Used' mask value is disregarded in this context.

Testing and Verification

Step 1: Identifying Active Lanes

Using the 'Module_Lanes_Used' counter, identify the active lanes for each port. For NDR switches, deter 'Module mine whether lanes 0-3 or 4-7 are active based on the counter's value.

Step 2: Recording Power Values

Record the Rx and Tx power values for the active lanes. Ensure that the values are accurately captured and correspond to the designated lanes.

Step 3: Cross-Verification

Cross-verify the recorded power values with the expected values for the active lanes. This step is crucial for identifying any discrepancies or anomalies in the power readings.

Step 4: Ignoring Irrelevant Lanes

For InfiniBand, ignore the 'Module_Lanes_Used' mask value and focus solely on lanes 0-3. Verify that the power values for these lanes are accurate and within the acceptable range.

Amber Simulation

Amber simulation supports the manual injection of amBER files into the Circuits View. These CSV files, collected by NVIDIA's *mlxlink* tool from PCI devices, contain port statistics and counters.

Key Counters Included:

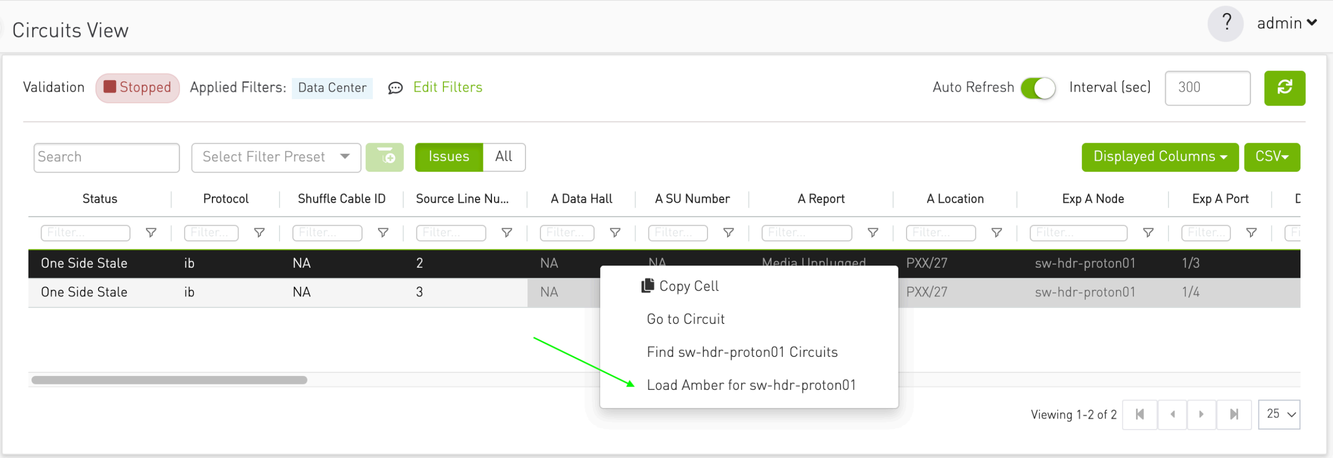
1. **BER Counters** – Indicate bit errors on the link.
2. **RX/TX Power Levels** – Provide power metrics to assess link activity.
3. **Module Temperature**
4. **Module Status** – Indicates whether the module is plugged in or unplugged.
5. **And more...**

This functionality allows users to load amBER files for one or both sides of a circuit, providing visual indicators and filtering options for improved data analysis and management.

It is particularly useful in the following scenarios:

- **No Agent Installed or No Connectivity**: When the node lacks an installed agent or has no active connection.
- **Historical Analysis**: Users can analyze past events that are no longer present in live data by loading an amBER file captured during the event.
- **Debugging and Simulation**: amBER files can be edited to simulate and debug corner-case scenarios.

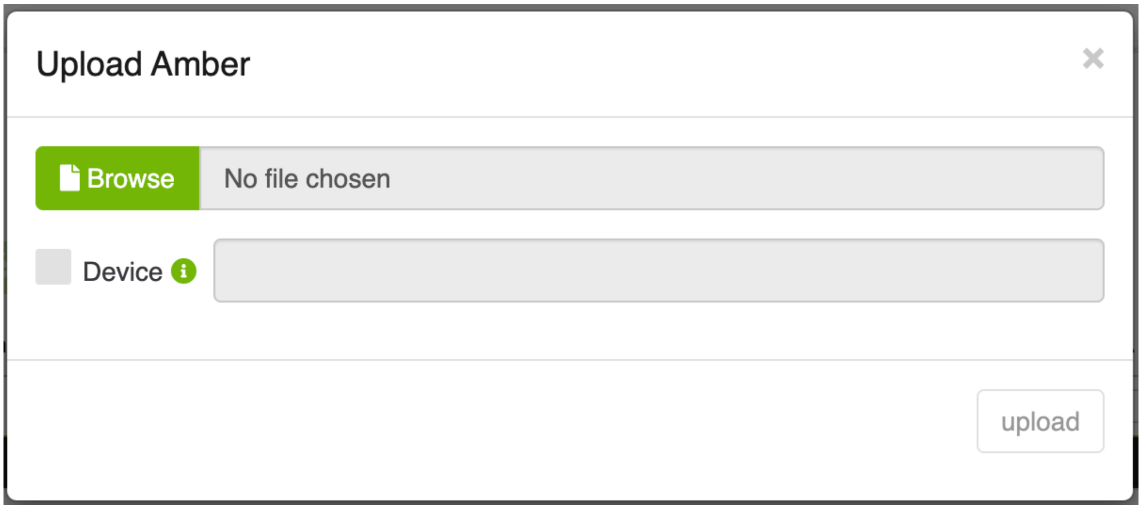
To simulate any endpoint, the user can right-click on it and select **'Load Amber for <endpoint name>'**. This action is only available to non-cabler users.



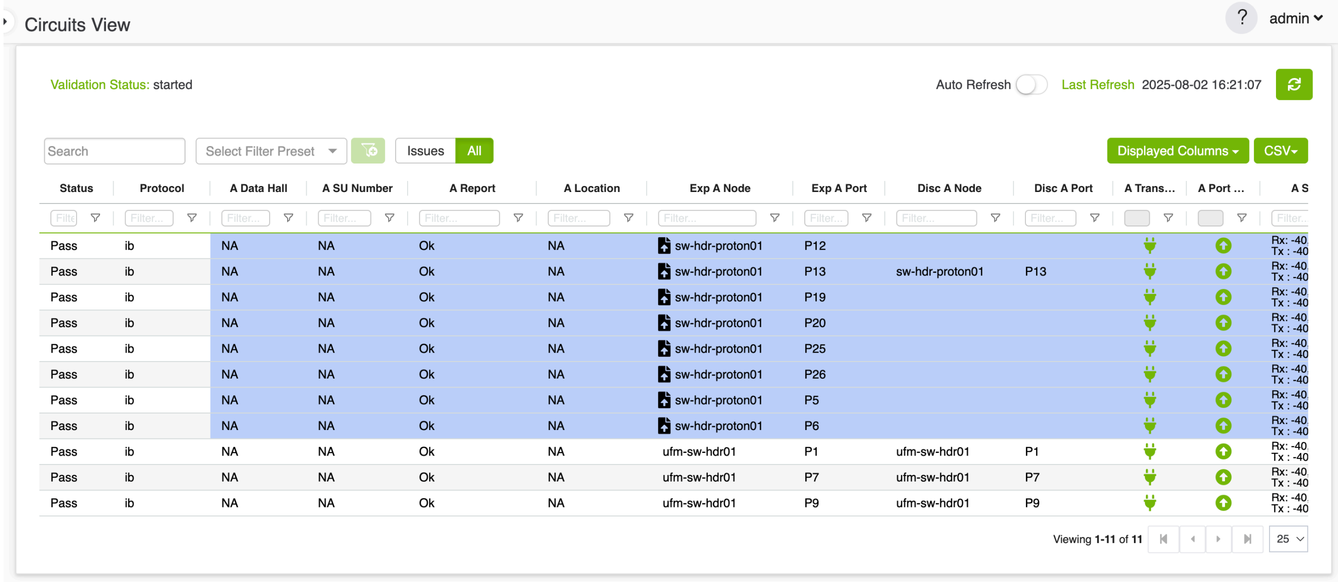
Status	Protocol	Shuffle Cable ID	Source Line Nu...	A Data Hall	A SU Number	A Report	A Location	Exp A Node	Exp A Port
One Side Stale	ib	NA	2	NA	NA	Media Unplugged	PXX/27	sw-hdr-proton01	1/3
One Side Stale	ib	NA	3	NA			PXX/27	sw-hdr-proton01	1/4

This will open a modal to upload the amBER file. The modal contains two input fields:

1. **Amber File** – Accepts a CSV file or a compressed CSV file (if the file is large).
2. **Device Name** – Required for nodes with multiple devices (e.g., hosts and NVL switches).



After loading the amBER file, the simulated endpoint row will be highlighted in **blue**.



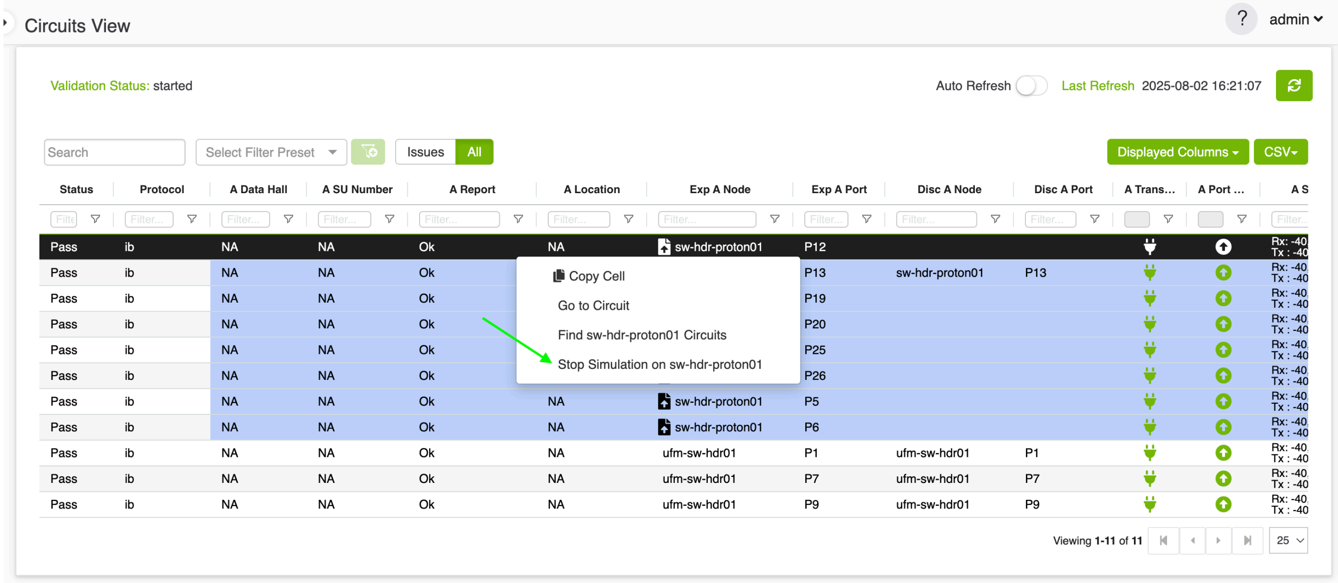
Validation Status: started

Last Refresh 2025-08-02 16:21:07

Status	Protocol	A Data Hall	A SU Number	A Report	A Location	Exp A Node	Exp A Port	Disc A Node	Disc A Port	A Trans...	A Port ...	A S
Pass	ib	NA	NA	Ok	NA	sw-hdr-proton01	P12			🔌	⬆	Rx: -40 Tx : -40
Pass	ib	NA	NA	Ok	NA	sw-hdr-proton01	P13	sw-hdr-proton01	P13	🔌	⬆	Rx: -40 Tx : -40
Pass	ib	NA	NA	Ok	NA	sw-hdr-proton01	P19			🔌	⬆	Rx: -40 Tx : -40
Pass	ib	NA	NA	Ok	NA	sw-hdr-proton01	P20			🔌	⬆	Rx: -40 Tx : -40
Pass	ib	NA	NA	Ok	NA	sw-hdr-proton01	P25			🔌	⬆	Rx: -40 Tx : -40
Pass	ib	NA	NA	Ok	NA	sw-hdr-proton01	P26			🔌	⬆	Rx: -40 Tx : -40
Pass	ib	NA	NA	Ok	NA	sw-hdr-proton01	P5			🔌	⬆	Rx: -40 Tx : -40
Pass	ib	NA	NA	Ok	NA	sw-hdr-proton01	P6			🔌	⬆	Rx: -40 Tx : -40
Pass	ib	NA	NA	Ok	NA	ufm-sw-hdr01	P1	ufm-sw-hdr01	P1	🔌	⬆	Rx: -40 Tx : -40
Pass	ib	NA	NA	Ok	NA	ufm-sw-hdr01	P7	ufm-sw-hdr01	P7	🔌	⬆	Rx: -40 Tx : -40
Pass	ib	NA	NA	Ok	NA	ufm-sw-hdr01	P9	ufm-sw-hdr01	P9	🔌	⬆	Rx: -40 Tx : -40

Viewing 1-11 of 11 25

To remove the simulation, the user can simply right-click on the simulated endpoint and select **'Stop Simulation on <endpoint name>'**. This action is also only available to non-cabler users.



Validation Status: started

Last Refresh 2025-08-02 16:21:07

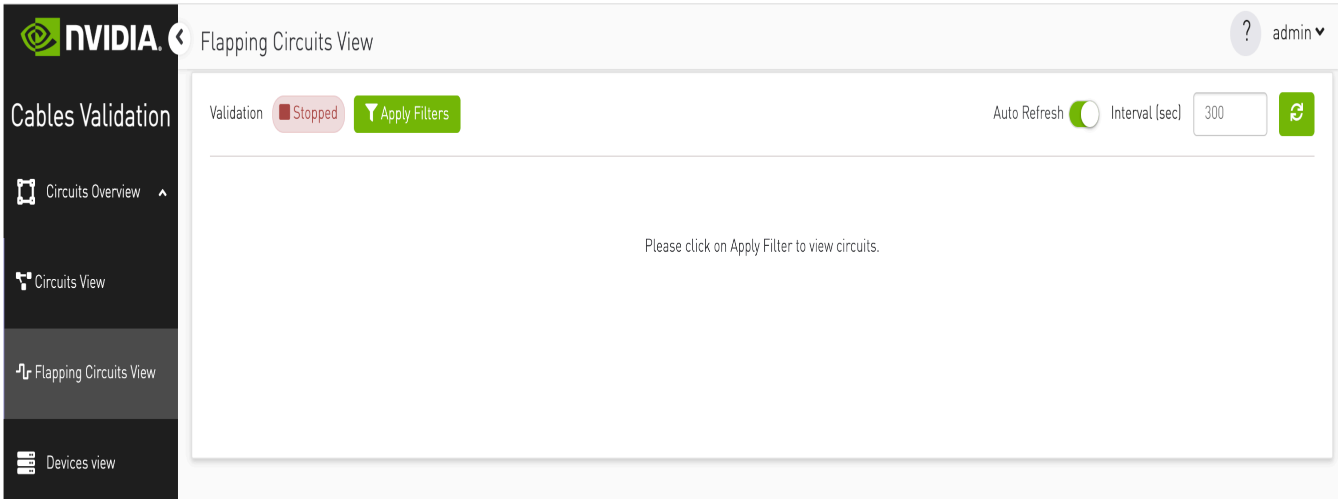
Status	Protocol	A Data Hall	A SU Number	A Report	A Location	Exp A Node	Exp A Port	Disc A Node	Disc A Port	A Trans...	A Port ...	A S
Pass	ib	NA	NA	Ok	NA	sw-hdr-proton01	P12			🔌	⬆	Rx: -40 Tx : -40
Pass	ib	NA	NA	Ok			P13	sw-hdr-proton01	P13	🔌	⬆	Rx: -40 Tx : -40
Pass	ib	NA	NA	Ok			P19			🔌	⬆	Rx: -40 Tx : -40
Pass	ib	NA	NA	Ok			P20			🔌	⬆	Rx: -40 Tx : -40
Pass	ib	NA	NA	Ok			P25			🔌	⬆	Rx: -40 Tx : -40
Pass	ib	NA	NA	Ok			P26			🔌	⬆	Rx: -40 Tx : -40
Pass	ib	NA	NA	Ok	NA	sw-hdr-proton01	P5			🔌	⬆	Rx: -40 Tx : -40
Pass	ib	NA	NA	Ok	NA	sw-hdr-proton01	P6			🔌	⬆	Rx: -40 Tx : -40
Pass	ib	NA	NA	Ok	NA	ufm-sw-hdr01	P1	ufm-sw-hdr01	P1	🔌	⬆	Rx: -40 Tx : -40
Pass	ib	NA	NA	Ok	NA	ufm-sw-hdr01	P7	ufm-sw-hdr01	P7	🔌	⬆	Rx: -40 Tx : -40
Pass	ib	NA	NA	Ok	NA	ufm-sw-hdr01	P9	ufm-sw-hdr01	P9	🔌	⬆	Rx: -40 Tx : -40

Viewing 1-11 of 11 25

Flapping Circuits View

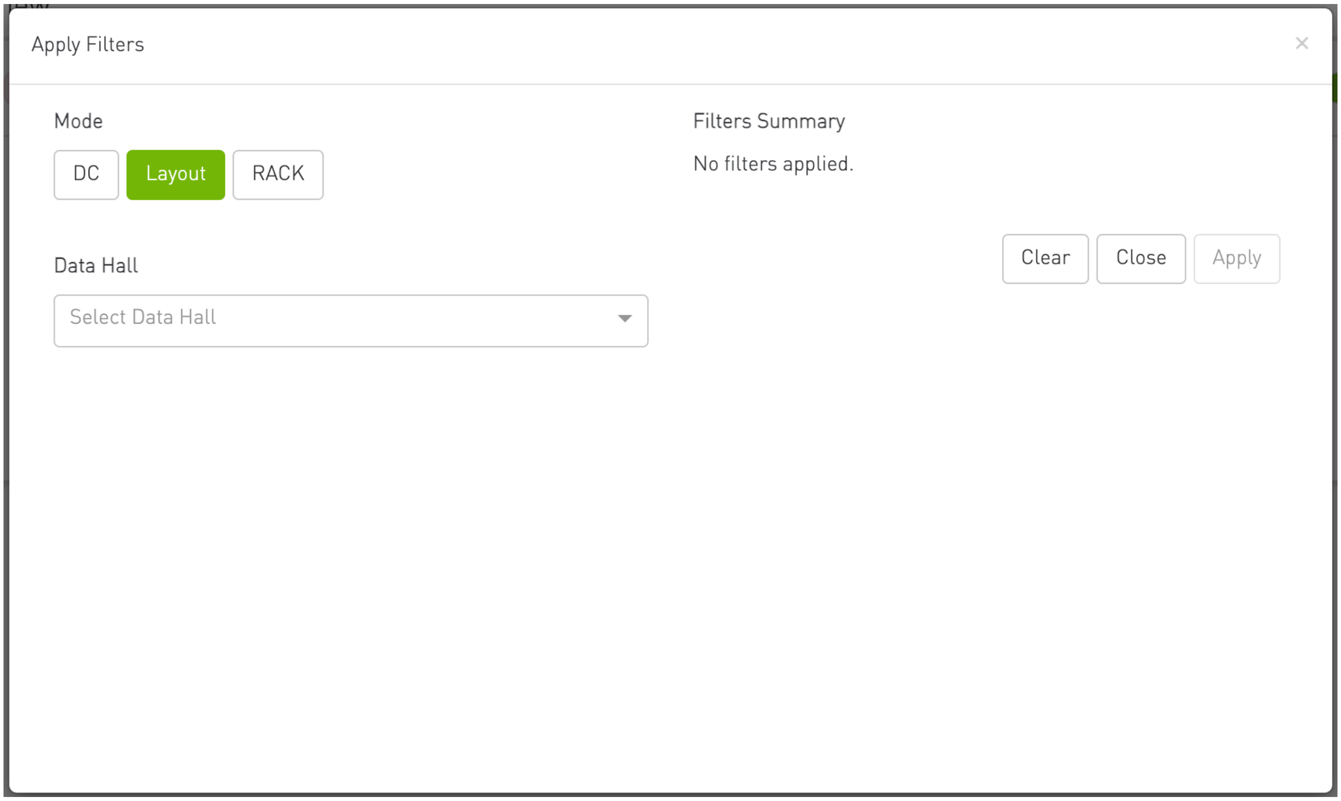
The **Flapping Circuits View** only supported for Admin users**,** to help explore the connection flapping details for A & Z endpoints.

When you open the page, you’ll see a message asking you to apply a filter to view the circuits.



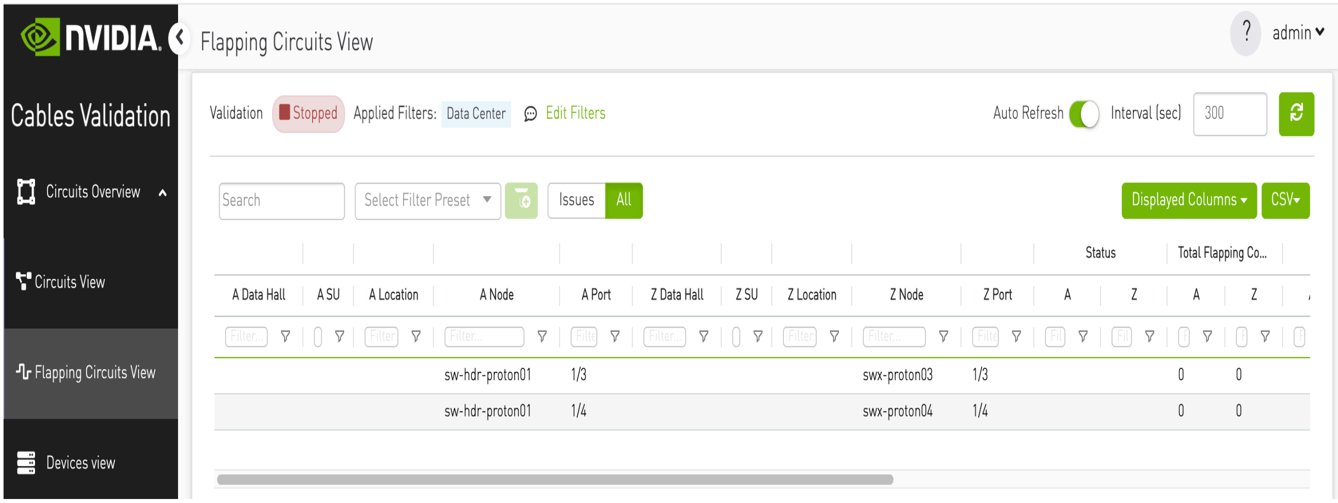
Please click on Apply Filter to view circuits.

Click **“Apply Filters”** to open the filter window, choose your preferred filter, and apply it. For more guidance, see the **Resource Filter** page.



No filters applied.

Once your filter is applied, all matching circuits will appear in a table.



Validation: Stopped

Applied Filters: Data Center

Auto Refresh

Interval (sec): 300

										Status		Total Flapping Co...	
A Data Hall	A SU	A Location	A Node	A Port	Z Data Hall	Z SU	Z Location	Z Node	Z Port	A	Z	A	Z
			sw-hdr-proton01	1/3				swx-proton03	1/3			0	0
			sw-hdr-proton01	1/4				swx-proton04	1/4			0	0

On IB and Cumulus switches, carrier transitions are monitored every 10 seconds. If the carrier transitions increments by more than 1, a link flap alarm is raised and the circuit will treated as flapping circuit.

The system provides a detailed view of current and historical flapping events for both endpoints. The historical data spans multiple time intervals, including the last 30 seconds, 1 minute, 5 minutes, 1 hour, 12 hours, and 24 hours. This information is displayed in a table.

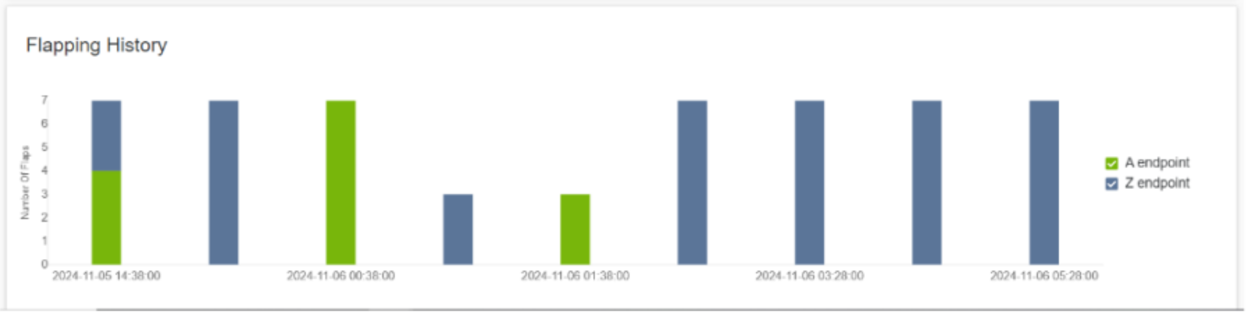
The table includes the following columns for both endpoints:

- **Data Hall and SU**
- **Location**: Specifies the endpoint's location, Rack, and Unit.
- **Node and Port Details**: information about the nodes and the associated ports.
- **Status**: flapping status and it could be one of the following values
    1. **Ok** - no flapping events since agent started
    2. **Flapping** - agent detected a flapping event **in the last 1 minute**
    3. **Flapped** - agent detected a flapping event at some point. The Flapping event counter >= 1.
- **Total Flapping Count:** This is the total number of transitions occurring **since the bringup agent** is started on the switch.
- **Flap 30 sec:** how many flaps happened in the last 30 seconds.

- **Flap 1 min:** how many flaps happened in the last 1 minute.
- **Flap 5 min:** how many flaps happened in the last 5 minutes.
- **Flap 1 hour:** how many flaps happened in the last 1 hour.
- **Flap 12 hour:** how many flaps happened in the last 12 hours.
- **Flap 24 hour:** how many flaps happened in the last 24 hours.

Flapping History

The circuit maintains a record of the number of flaps occurred, along with the corresponding timestamps, over a 24-hour period. This record is referred to as the **flapping history**. Users can view this history by **selecting any row in the flapping circuits table**.



The history is displayed as a bar chart, where **green bars represent the A endpoint** and **blue bars represent the Z endpoint**.

If flaps are detected simultaneously for both the A and Z endpoints, the Z value is stacked on top of the A value in the bar chart. For example, in the first bar of the chart, the A endpoint has a value of 4, and the Z endpoint has a value of 3, making the total height of the bar 7.

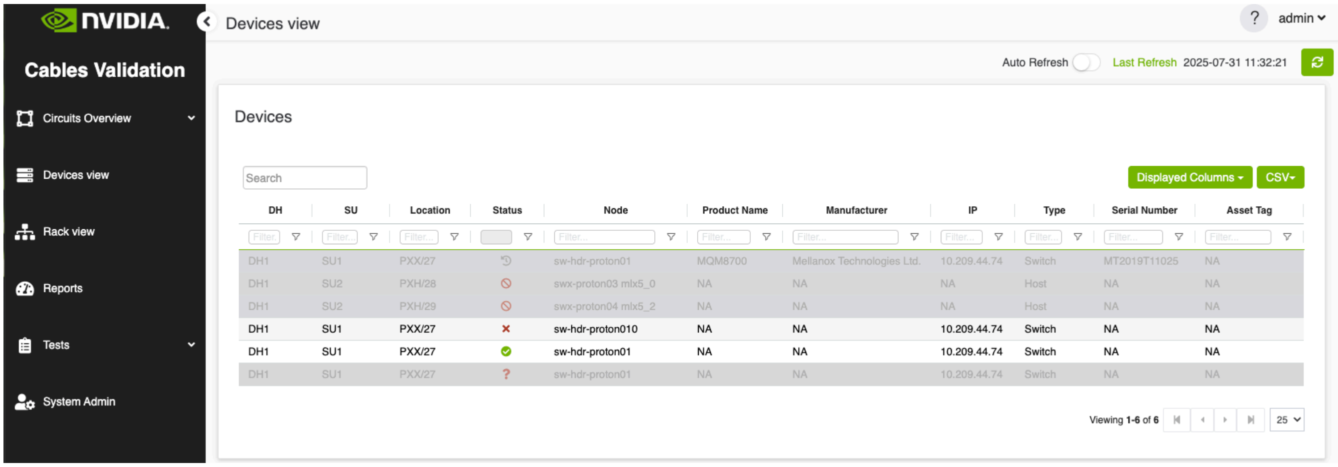
Additionally, users can toggle the visibility of individual bars by interacting with the chart legend located on the right-hand side of the chart.

Devices View

The Devices View presents network devices in a structured table format, allowing users to browse, filter, and select devices for detailed analysis.

Upon device selection, the interface provides detailed information through multiple tabs, including circuits, agent status, and device health metrics.

The system automatically refreshes device information and provides real-time status updates. Users can export device data, filter results, and navigate through different device details.



Last Refresh 2025-07-31 11:32:21

Devices

DH	SU	Location	Status	Node	Product Name	Manufacturer	IP	Type	Serial Number	Asset Tag
DH1	SU1	PXX/27	↺	sw-hdr-proton01	MQM8700	Mellanox Technologies Ltd.	10.209.44.74	Switch	MT2019T11025	NA
DH1	SU2	PXH/28	⊘	swx-proton03 mlx5_0	NA	NA	NA	Host	NA	NA
DH1	SU2	PXH/29	⊘	swx-proton04 mlx5_2	NA	NA	NA	Host	NA	NA
DH1	SU1	PXX/27	✗	sw-hdr-proton010	NA	NA	10.209.44.74	Switch	NA	NA
DH1	SU1	PXX/27	✓	sw-hdr-proton01	NA	NA	10.209.44.74	Switch	NA	NA
DH1	SU1	PXX/27	?	sw-hdr-proton01	NA	NA	10.209.44.74	Switch	NA	NA

Viewing 1-6 of 6 25

The main interface displays devices in a tabular format with the following columns:

Column	Description
DH (Data Hall)	The data hall identifier where the device is located
SU (Scalable Unit)	The scalable unit within the data hall
Location	Physical location showing rack name and rack unit in format "Rack/Unit"
Status	Agent status, for more information refer to Agent Status
Node	The device node name with visual indicators for the device status
Product Name	The product's name
Manufacturer	The manufacturer name
IP	The device IP address
Type	The device type
Serial Number	Serial Number
Asset Tag	Asset Tag

Agent Status

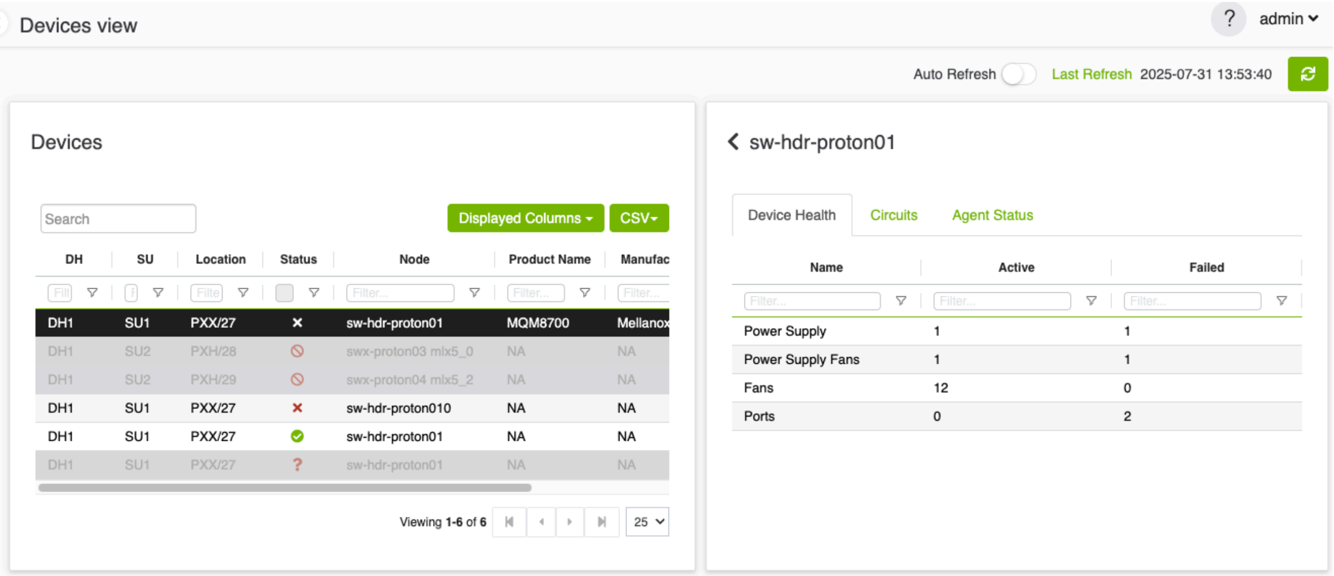
Agent status could be one of the following:

- **Unmanaged** (▯ Gray background): Agent installation is not supported on this device.
- **Agent Not Installed** (▯ Gray background): No agent is installed.
- **Stale Agent** (▯ Gray background): No update has been received from the agent in the last 15 minutes.
- **Agent Reporting Failures** (▯): The agent has reported either hardware issues (e.g., fans, power supplies) or port issues.
- **No Failures on Agent** (▯): No hardware or port issues detected by the agent.

Device Selection and Navigation

When a device is selected from the table, a details panel appears on the right side of the screen.

The selected device's information is highlighted in the device header, and the available tabs are dynamically determined based on the device's capabilities.



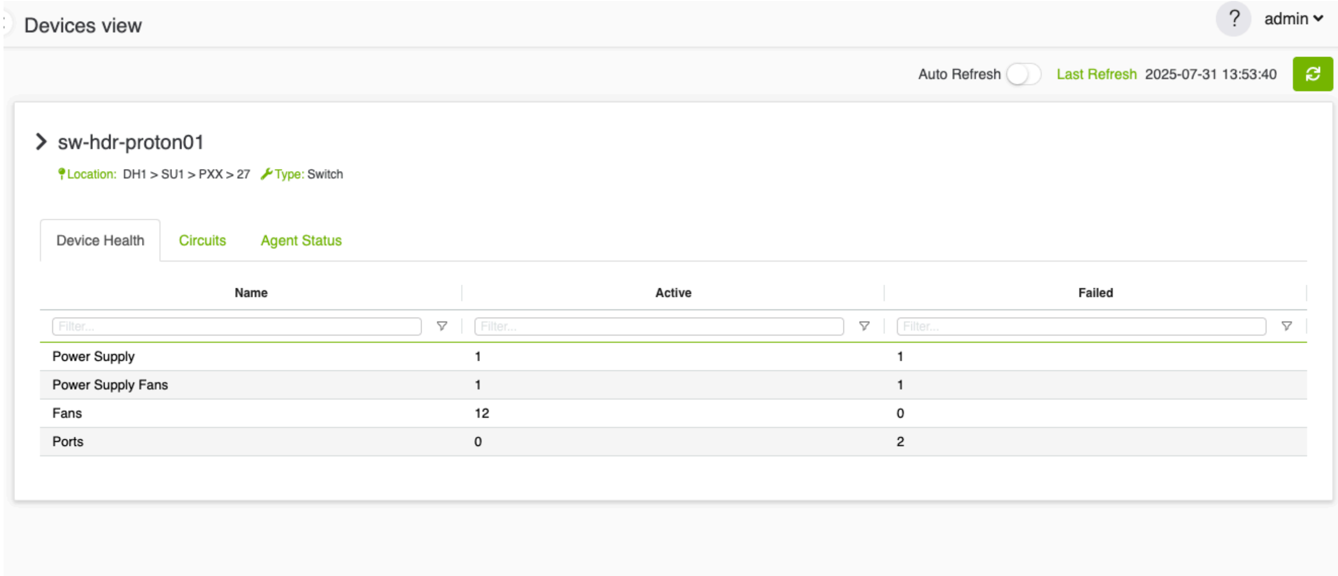
DH	SU	Location	Status	Node	Product Name	Manufac
DH1	SU1	PXX/27	✕	sw-hdr-proton01	MQM8700	Mellanox
DH1	SU2	PXH/28	⦸	swx-proton03 mlx5_0	NA	NA
DH1	SU2	PXH/29	⦸	swx-proton04 mlx5_2	NA	NA
DH1	SU1	PXX/27	✕	sw-hdr-proton010	NA	NA
DH1	SU1	PXX/27	✓	sw-hdr-proton01	NA	NA
DH1	SU1	PXX/27	?	sw-hdr-proton01	NA	NA

Name	Active	Failed
Power Supply	1	1
Power Supply Fans	1	1
Fans	12	0
Ports	0	2

The device header displays the device name with expand/collapse functionality on the left side of the name.

In expanded mode, the device details panel stretches to fit the page and displays the device's location hierarchy — Data Hall > Scalable Unit > Rack > Unit (when available).

It also includes information about the device type.



Last Refresh 2025-07-31 13:53:40

sw-hdr-proton01

Location: DH1 > SU1 > PXX > 27 Type: Switch

Name	Active	Failed
Power Supply	1	1
Power Supply Fans	1	1
Fans	12	0
Ports	0	2

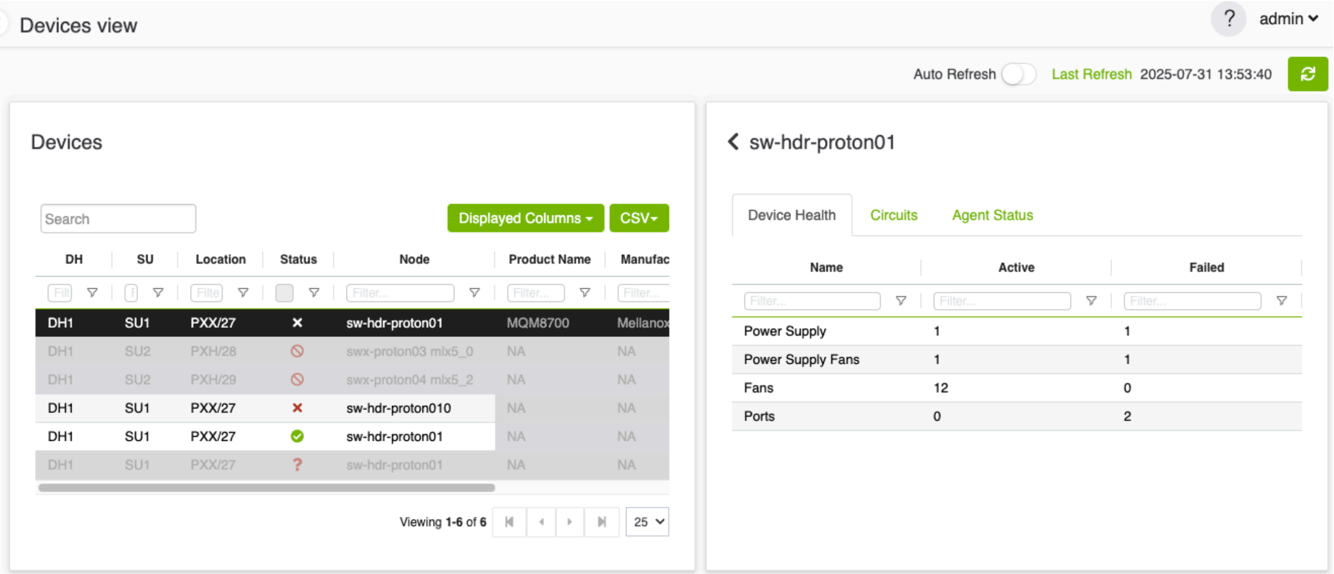
Device Tabs

Device Health Tab

The **Device Health** tab provides an overview of the selected device's health metrics in a tabular format.

Note

Note: This tab is only visible for manageable devices.

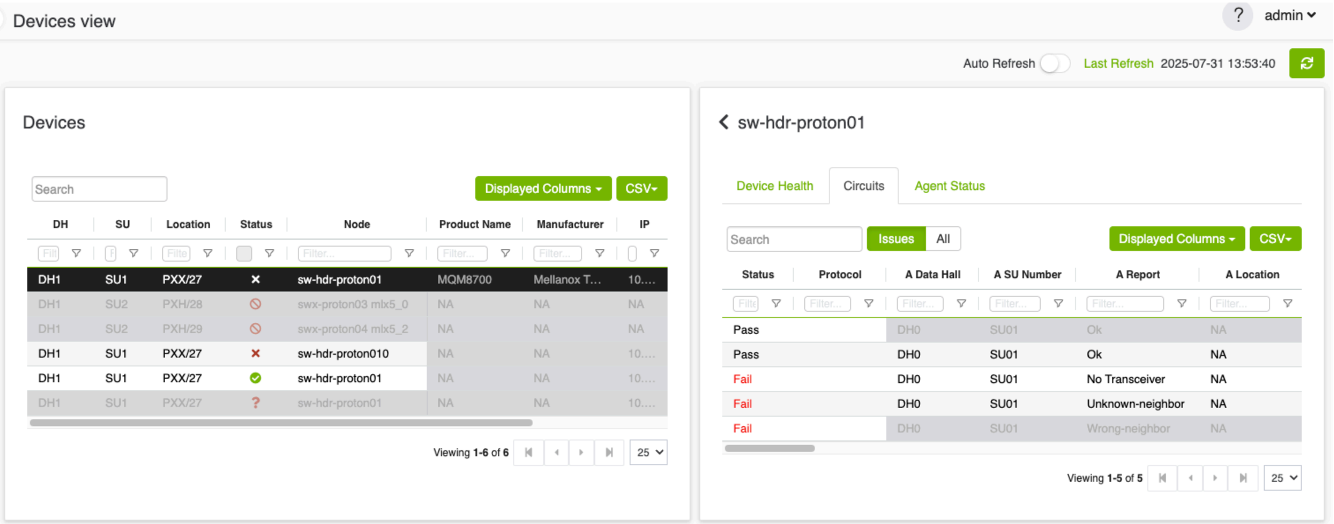


DH	SU	Location	Status	Node	Product Name	Manufac
DH1	SU1	PXX/27	✕	sw-hdr-proton01	MQM8700	Mellanox
DH1	SU2	PXH/28	⊘	swx-proton03 mlx5_0	NA	NA
DH1	SU2	PXH/29	⊘	swx-proton04 mlx5_2	NA	NA
DH1	SU1	PXX/27	✕	sw-hdr-proton010	NA	NA
DH1	SU1	PXX/27	✓	sw-hdr-proton01	NA	NA
DH1	SU1	PXX/27	?	sw-hdr-proton01	NA	NA

Name	Active	Failed
Power Supply	1	1
Power Supply Fans	1	1
Fans	12	0
Ports	0	2

Circuits Tab

The **Circuits** tab displays all circuits related to the selected device.



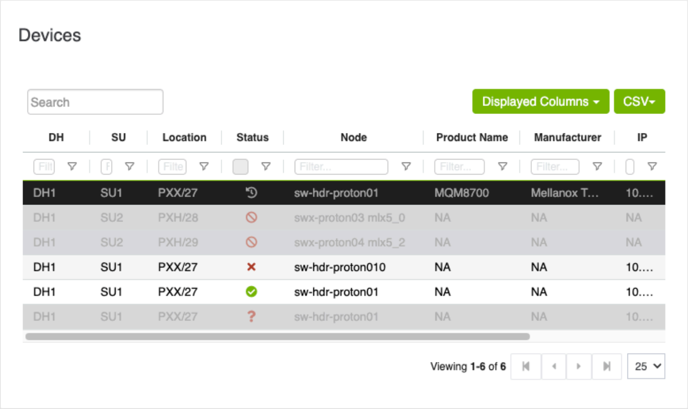
DH	SU	Location	Status	Node	Product Name	Manufacturer	IP
DH1	SU1	PXX/27	✕	sw-hdr-proton01	MQM8700	Mellanox T...	10....
DH1	SU2	PXH/28	⊘	swx-proton03 mlx5_0	NA	NA	NA
DH1	SU2	PXH/29	⊘	swx-proton04 mlx5_2	NA	NA	NA
DH1	SU1	PXX/27	✕	sw-hdr-proton010	NA	NA	10....
DH1	SU1	PXX/27	✓	sw-hdr-proton01	NA	NA	10....
DH1	SU1	PXX/27	?	sw-hdr-proton01	NA	NA	10....

Status	Protocol	A Data Hall	A SU Number	A Report	A Location
Pass		DH0	SU01	Ok	NA
Pass		DH0	SU01	Ok	NA
Fail		DH0	SU01	No Transceiver	NA
Fail		DH0	SU01	Unknown-neighbor	NA
Fail		DH0	SU01	Wrong-neighbor	NA

Agent Status Tab

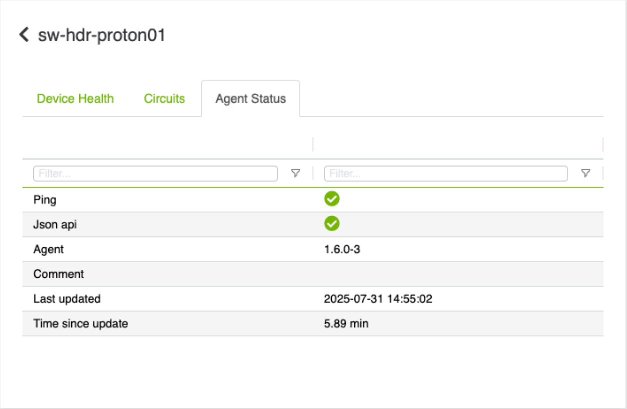
The **Agent Status** tab presents detailed information about the software agent running on managed devices. It includes the agent's connectivity status, version information, and the timestamp of the last communication.

Note: This tab is only visible for manageable devices and is accessible only to users with administrative privileges.



DH	SU	Location	Status	Node	Product Name	Manufacturer	IP
DH1	SU1	PXX/27	⟲	sw-hdr-proton01	MQM8700	Mellanox T...	10....
DH1	SU2	PXH/28	⊘	swx-proton03 mlx5_0	NA	NA	NA
DH1	SU2	PXH/29	⊘	swx-proton04 mlx5_2	NA	NA	NA
DH1	SU1	PXX/27	✗	sw-hdr-proton010	NA	NA	10....
DH1	SU1	PXX/27	✓	sw-hdr-proton01	NA	NA	10....
DH1	SU1	PXX/27	?	sw-hdr-proton01	NA	NA	10....

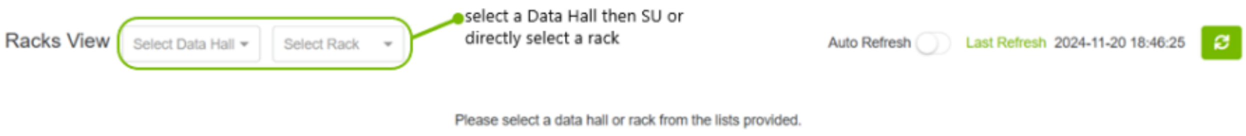
Viewing 1-6 of 6



Ping	✓
Json api	✓
Agent	1.6.0-3
Comment	
Last updated	2025-07-31 14:55:02
Time since update	5.89 min

Rack View

The Rack View provides a simulated visual representation of a selected rack, including its devices, ports, and statuses. This view is populated using data from a PTP Excel file.



Racks View
Select Data Hall
Select Rack
select a Data Hall then SU or directly select a rack
Auto Refresh
Last Refresh 2024-11-20 18:46:25

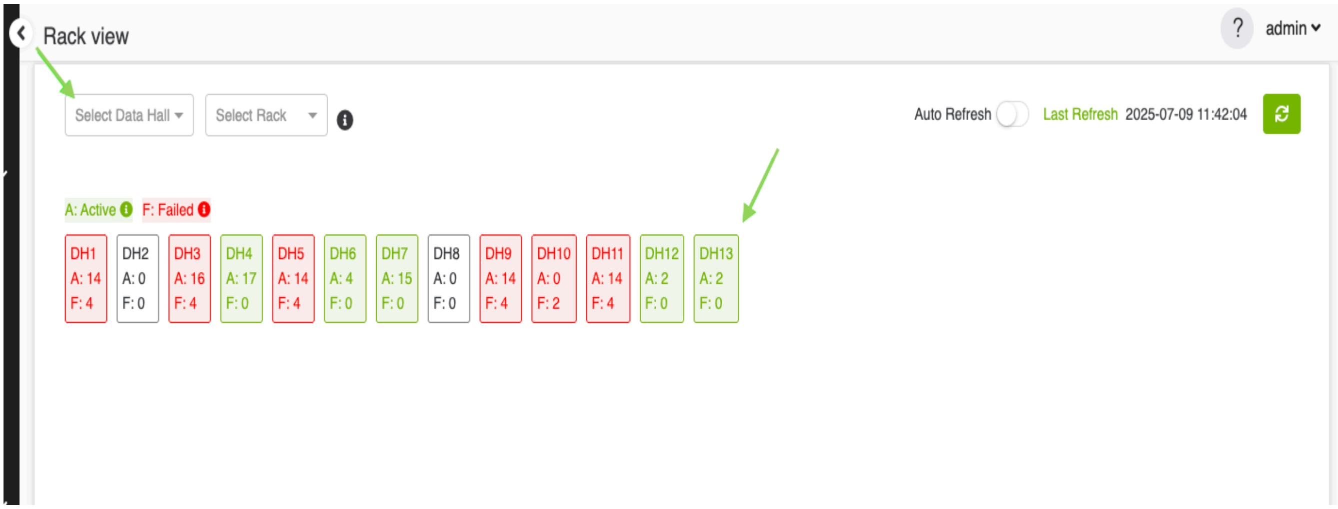
Please select a data hall or rack from the lists provided.

The page is initially blank, and user has 2 options:

Hierarchical Selection Workflow

This hierarchical selection process allows users to navigate from a Data Hall to SU and then to specific Racks. This feature is only available if the user has provided the **DC Floor Layout file** while loading the P2P file. Below are the steps the user should follow:

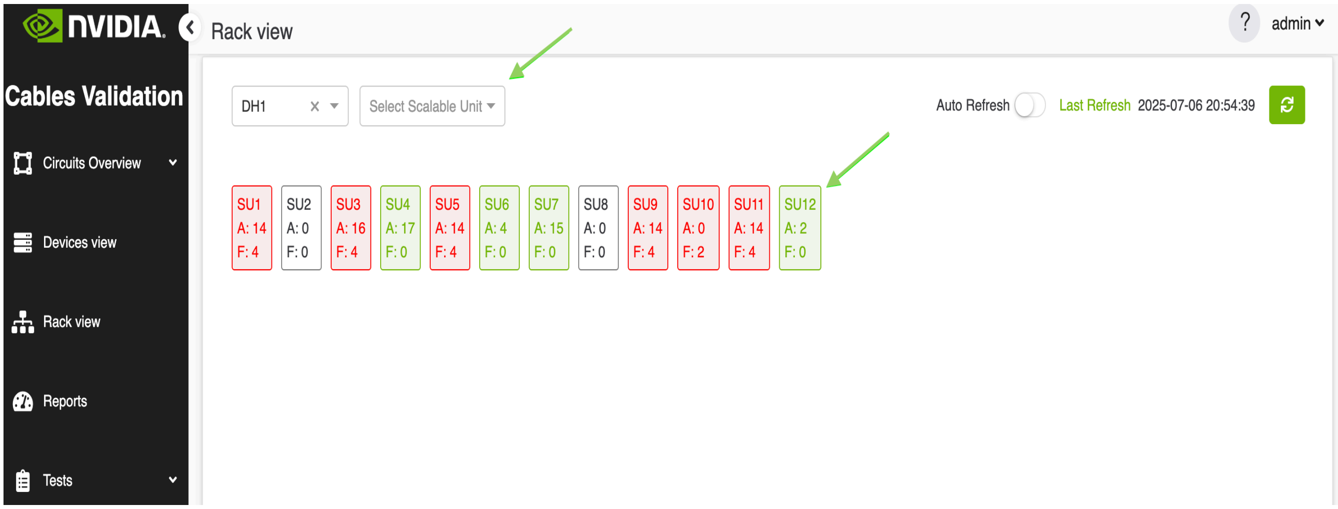
1. **Select a Data Hall:** Begin by selecting a Data Hall from the dropdown menu or by clicking on one of the DH squares displayed below, which show the number of active and failed components within each Data Hall. This selection determines the scope of available Scalable Units and Racks.



2. **Display and Select SU:** Once a Data Hall is selected, all associated **Scalable Units (SUs)** will be displayed in two forms:

-
    - **Dropdown Menu** – A list of available SUs for easy selection.
    - **Visual Squares** – Each square represents an SU and displays the number of active and failed components.

The user can either select an SU from the dropdown or click on one of the SU squares in the visual layout.

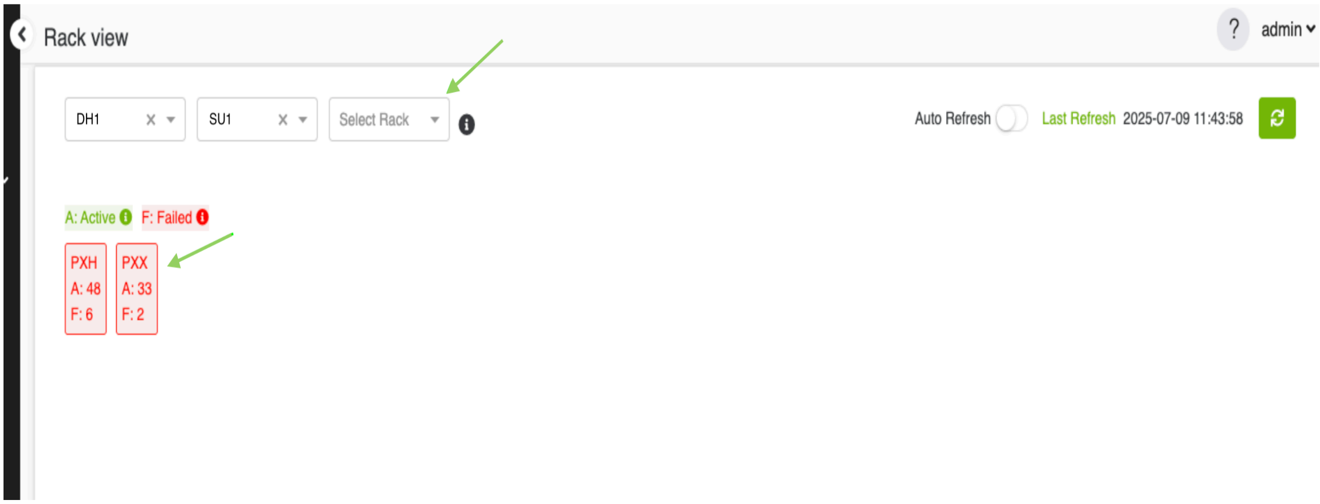


3. **Display and Select Racks:** the same as SU, after selecting an SU, all associated **Racks** within that SU will be displayed in two forms:

-

- **Dropdown Menu** – A list of all racks within the selected SU.
- **Visual Squares** – Each square represents a Rack and includes the number of active and failed components.

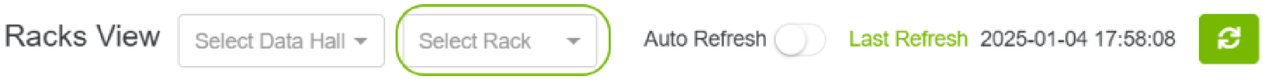
The user can either select a rack from the dropdown menu or click on a rack name to view the simulated visual representation of the selected rack.



Select a Rack Directly

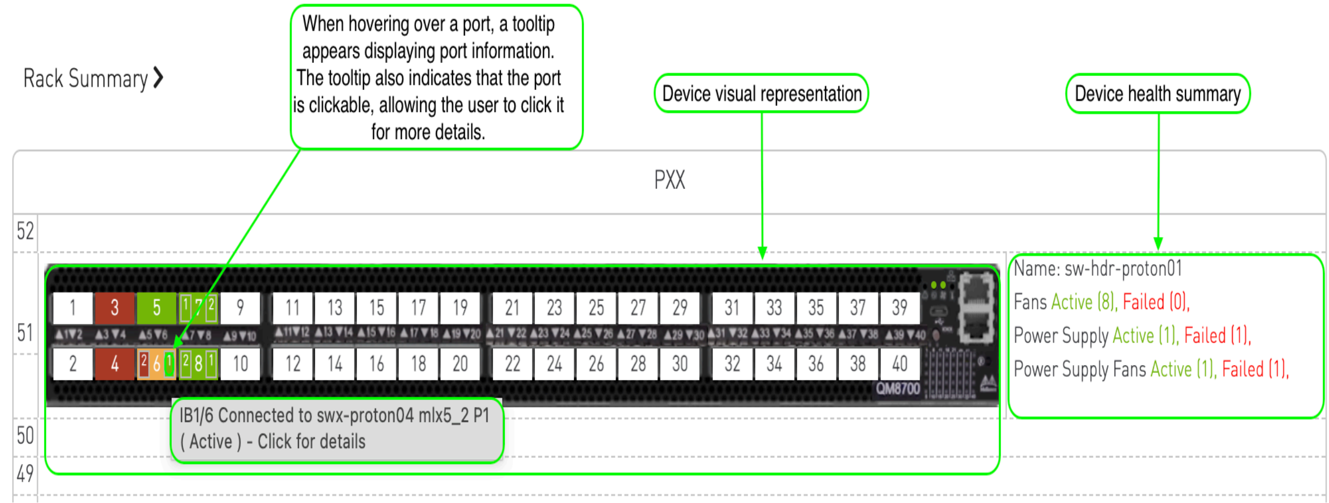
If the DC Floor layout is not loaded, user can directly select a rack from the rack menu to see the simulated visual representation for it.

The rack menu will contain all racks recognized by the CVT tool.



Rack Visual Representation

Once a rack is selected, the corresponding visualization of the rack, along with its devices and ports, will be displayed.



When hovering over a port, a tooltip appears displaying port information. The tooltip also indicates that the port is clickable, allowing the user to click it for more details.

Device visual representation

Device health summary

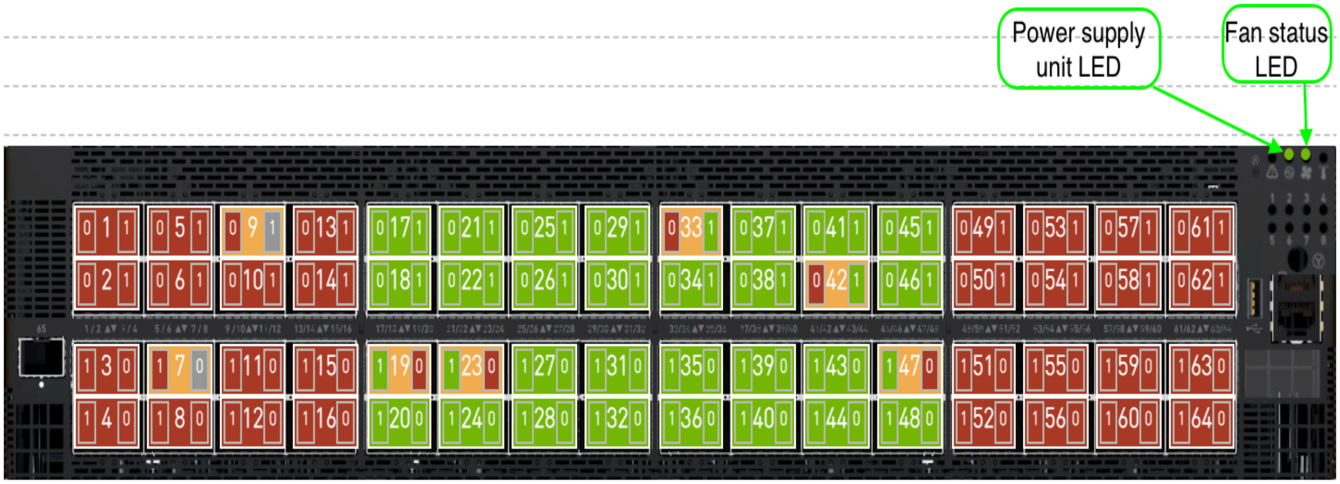
IB1/6 Connected to swx-proton04 mlx5_2 P1 (Active) - Click for details

Name: sw-hdr-proton01
Fans Active (8), Failed (0),
Power Supply Active (1), Failed (1),
Power Supply Fans Active (1), Failed (1),

The tool support visualization for the following switches and hosts:

Ethernet

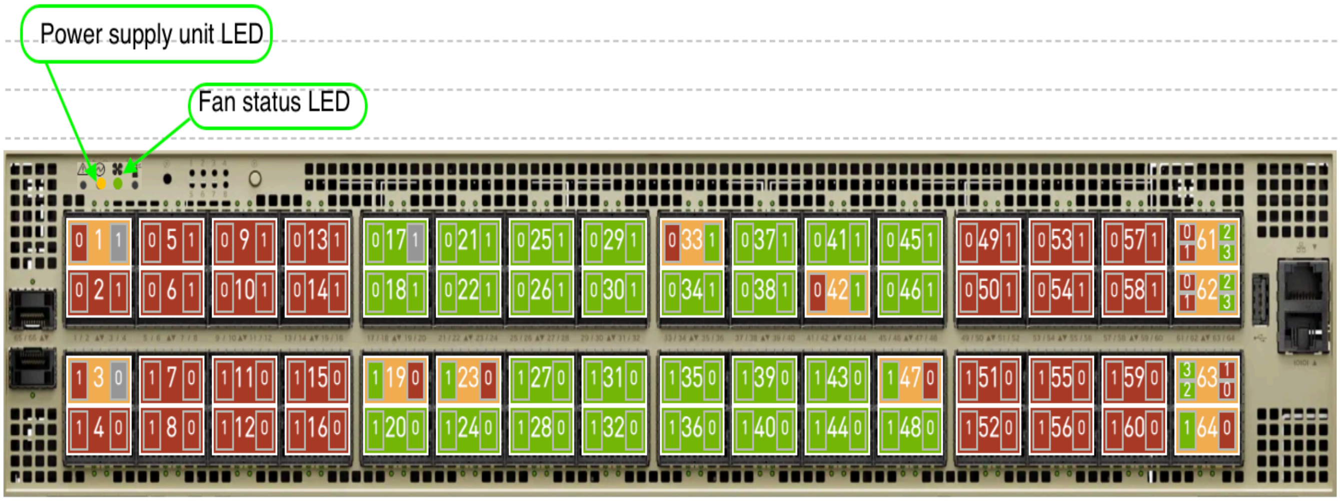
SN5600 Switch



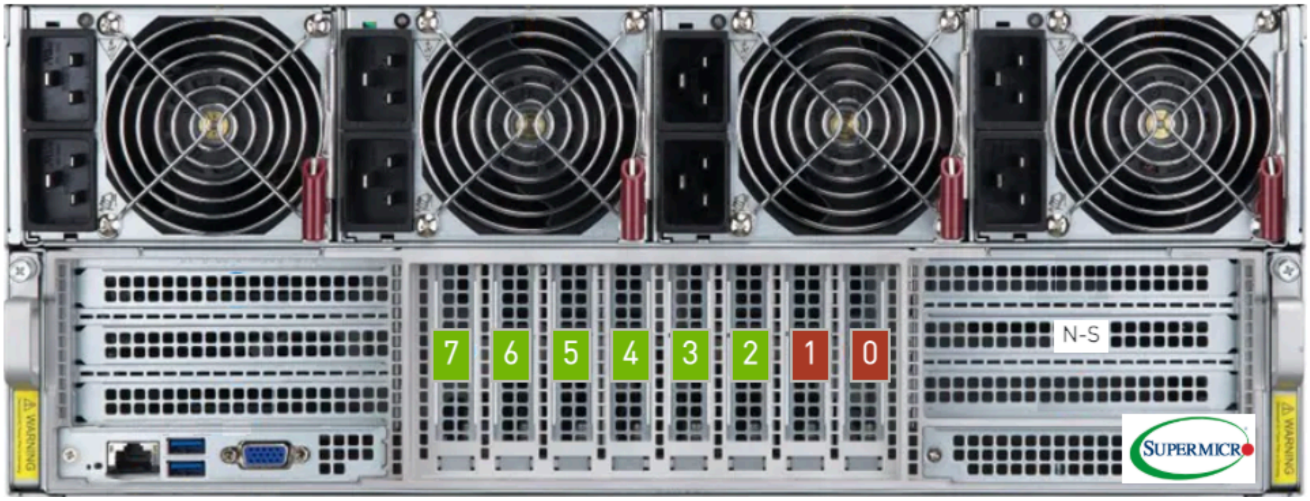
Power supply unit LED

Fan status LED

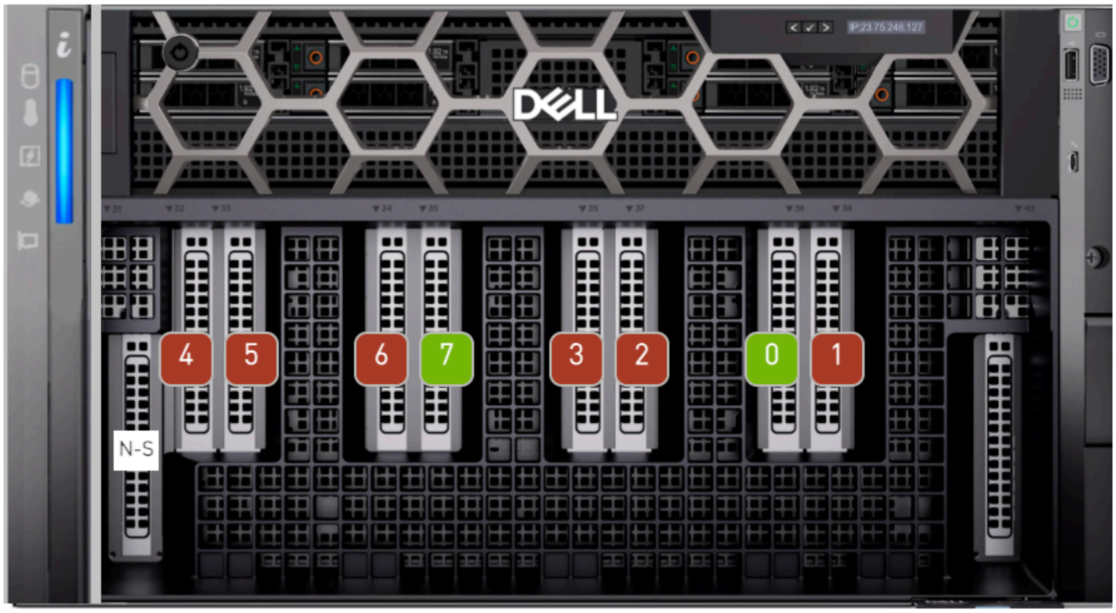
SN5600 Switch



SMC SuperServer

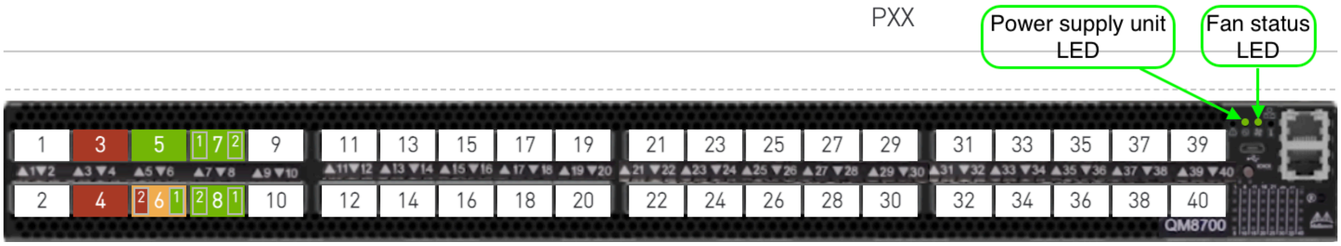


XE9680 Rack Server

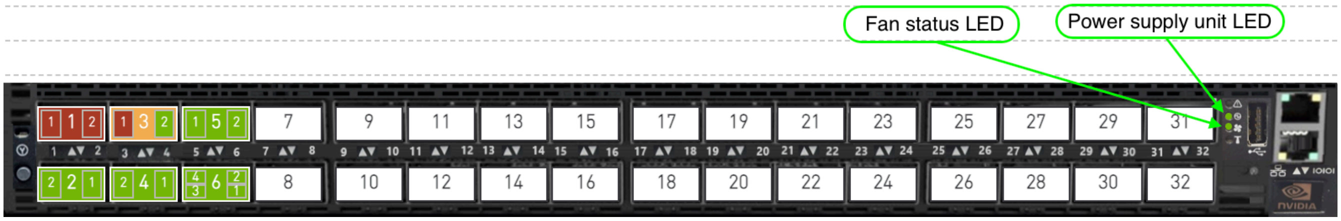


InfiniBand

MQM8700 Switch

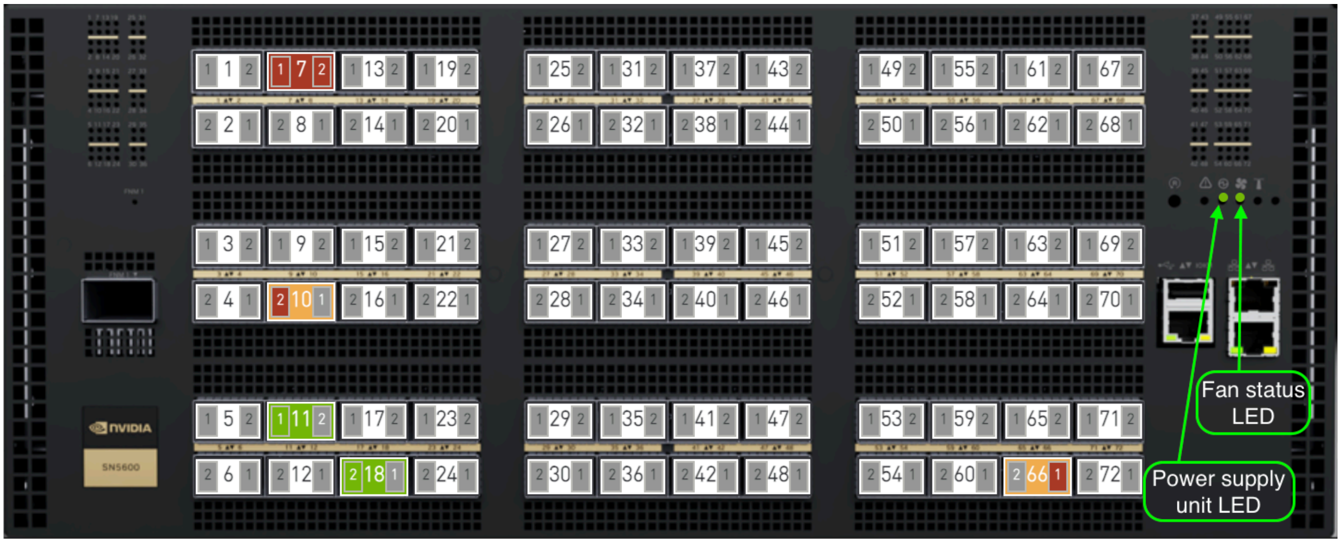


MQM9700 Switch



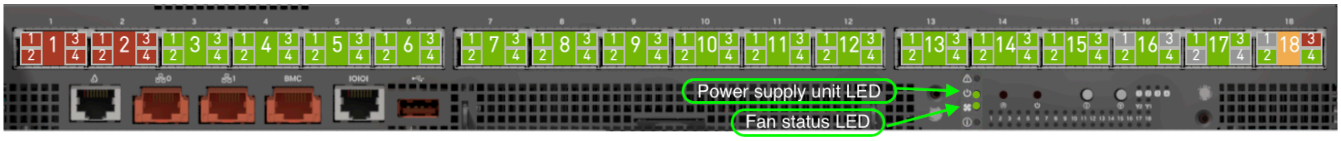
XDR

Quantum-3 XDR Switch



NVLink

Scaleout Switch



HGX Compute Tray

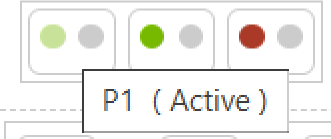
DGX Compute Tray

Dell Server 9712a

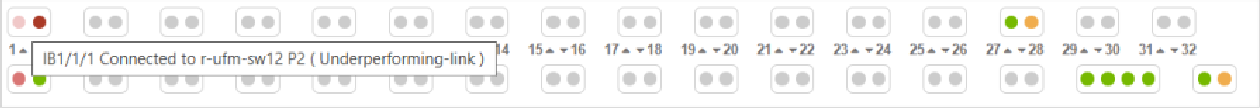
Default

The default view is used to display devices that don't have a product name.

Default Host



Default Switch



Port Status Coloring

Each port on the device is color-coded to visually represent its status, making it easier to identify and troubleshoot issues. The status of each port is indicated by the following colors:

Ports are color-coded based on their related issues:

- images/download/attachments/4465579547/worddav155b70fc8a2930e5fb137d19e version-1-modificationdate-1762936808653-api-v2.png White: Null (no connection or data).

- images/download/attachments/4465579547/worddav22d00e6b0d013f975bbda6a8c version-1-modificationdate-1762936808347-api-v2.png Green: Active and functioning properly.
- images/download/attachments/4465579547/worddav7812d2e90a8b3206b9e18a60 version-1-modificationdate-1762936808055-api-v2.png Light Red: Indicates issues such as:
    - Unknown Neighbor.
    - Underperforming Link (BER).
    - Extra Cable.
- images/download/attachments/4465579547/worddavd9979a05458cfc0a7187c765f version-1-modificationdate-1762936807696-api-v2.png Dark Red: Issues not covered by the above categories.

Split Port Status

Split ports occur when a single physical port is divided into multiple logical ports. For example, in the BlackMamba switch, a single physical port may be divided into two logical ports, named sw<num>p1 and sw<num>p2.

The color-coding system for split ports follows these guidelines:

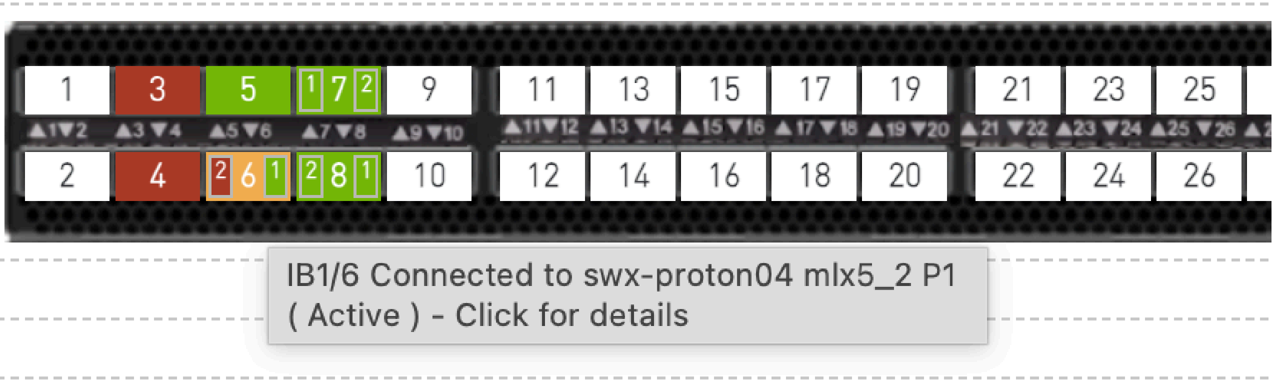
1. Uniform Status Across Logical Ports: If all logical ports of a split port share the same status, the coloring will match that of a non-split port.
2. Mixed Issues with Different Severities: If the logical ports have issues of varying severity, the background color will reflect highest severity.
3. Mixed Active and Issue States: If one or more logical ports have issues while others are active, the background color will be orange images/download/attachments/4465579547/worddav48054eef5cd40fda17ca080e6 version-1-modificationdate-1762936807422-api-v2.png , indicating a mixed state.

Port Details

When hovering over a port, a tooltip appears displaying:

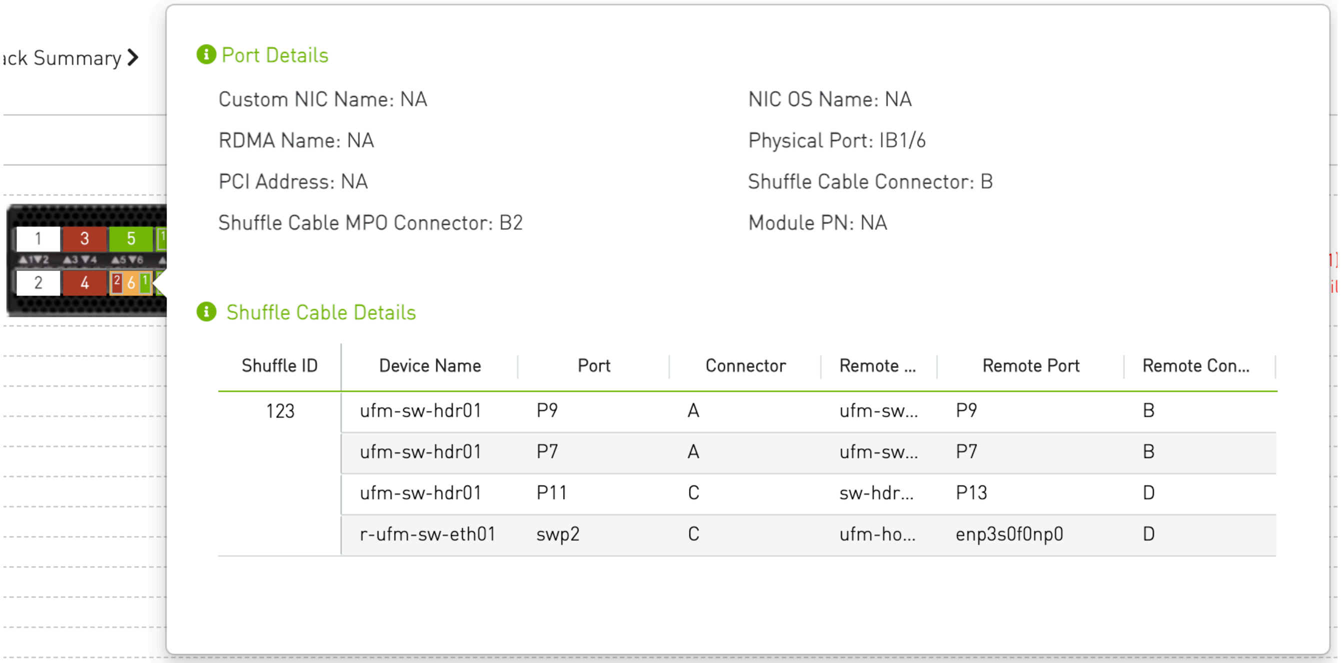
- Port-specific information

- Details about the associated issue (if any)
- An indication that the port is clickable



IB1/6 Connected to swx-proton04 mlx5_2 P1 (Active) - Click for details

Clicking on a port opens a popover that displays detailed information about the selected port, including shuffle cable details (if available).



Port Details

Custom NIC Name: NA

NIC OS Name: NA

RDMA Name: NA

Physical Port: IB1/6

PCI Address: NA

Shuffle Cable Connector: B

Shuffle Cable MPO Connector: B2

Module PN: NA

Shuffle Cable Details

Shuffle ID	Device Name	Port	Connector	Remote ...	Remote Port	Remote Con...
123	ufm-sw-hdr01	P9	A	ufm-sw...	P9	B
	ufm-sw-hdr01	P7	A	ufm-sw...	P7	B
	ufm-sw-hdr01	P11	C	sw-hdr...	P13	D
	r-ufm-sw-eth01	swp2	C	ufm-ho...	enp3s0f0np0	D

Status LEDs

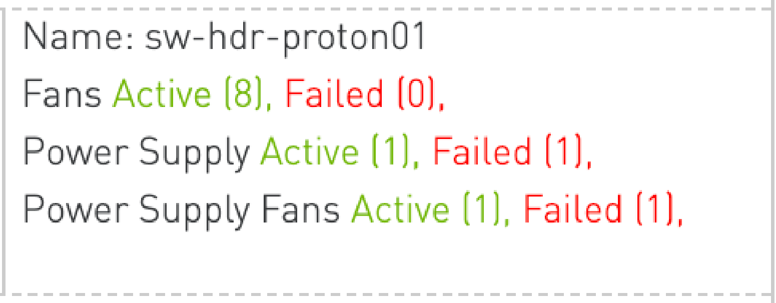
Symbol	Name	Description	Color
⏻	Power Supply Unit LED	Shows the health of the power supply units.	Amber if fail, Green if everything ok

	Fan Status LED	Shows the health of the fans	Amber if fail, Green if everything ok

Device Summary

Beside each device in the rack, a summary panel provides an overview of its status, including:

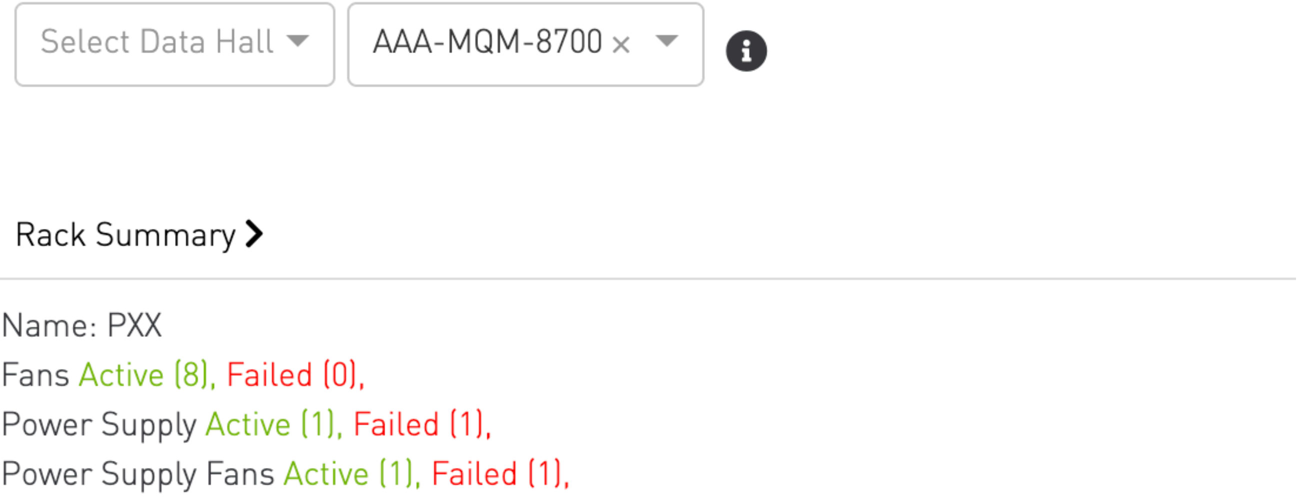
1. **Fans**: Count of active and failed fans.
2. **Ports**: Count of active and failed ports.
3. **Power Supplies**: Count of operational and failed power supply units.
4. **Power Supply Fans**: Count of operational and failed power supply fans.



Name: sw-hdr-proton01
Fans Active (8), Failed (0),
Power Supply Active (1), Failed (1),
Power Supply Fans Active (1), Failed (1),

Rack-Level Summary

At the top of the view, an aggregated summary for the entire rack is displayed.

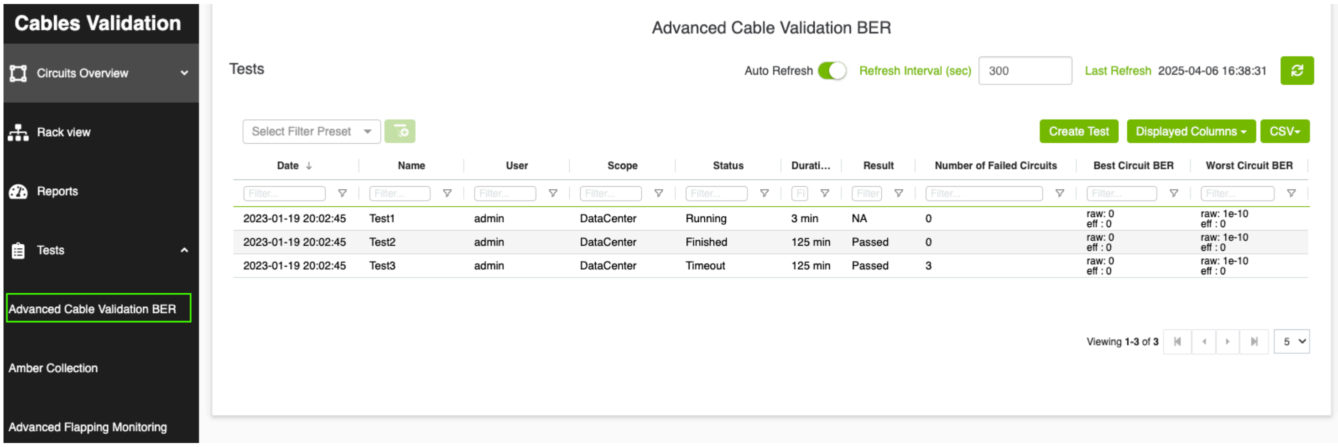


Name: PXX
Fans Active (8), Failed (0),
Power Supply Active (1), Failed (1),
Power Supply Fans Active (1), Failed (1),

Tests

This page is only supported for Admin users.

Golden BER Test



Refresh Interval (sec) 300

Last Refresh 2025-04-06 16:38:31

Date	Name	User	Scope	Status	Durati...	Result	Number of Failed Circuits	Best Circuit BER	Worst Circuit BER
2023-01-19 20:02:45	Test1	admin	DataCenter	Running	3 min	NA	0	raw: 0 eff : 0	raw: 1e-10 eff : 0
2023-01-19 20:02:45	Test2	admin	DataCenter	Finished	125 min	Passed	0	raw: 0 eff : 0	raw: 1e-10 eff : 0
2023-01-19 20:02:45	Test3	admin	DataCenter	Timeout	125 min	Passed	3	raw: 0 eff : 0	raw: 1e-10 eff : 0

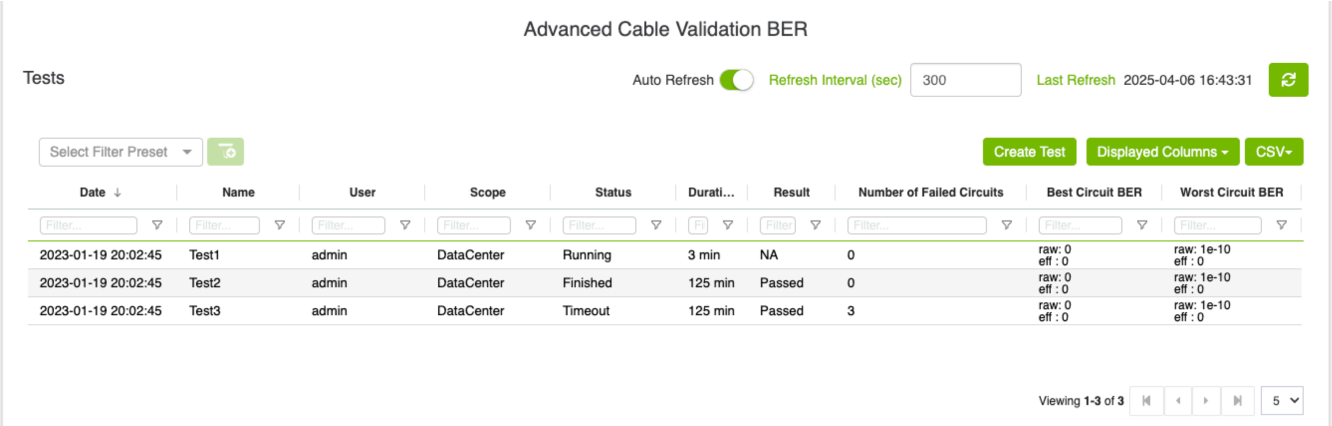
The Golden BER Test is designed to monitor Bit Error Rates (BER) and analyze the interface counters. The process begins by resetting interface counters on connected agents using the command **sudo mlxlink –pc**. The test then runs for a user-specified duration, during which the agents periodically report updated counter values every 10 minutes to a centralized collector. When the test completes, the final Amber file is sent to the collector.

In the UI, the test is presented in a tabular format with the following columns:

1. **Date**: The creation date of the test.
2. **Name**: The name assigned to the test.
3. **User**: The creator of the test.
4. **Scope**: Test scope, which usually includes **SU<n>/DH<n>, the DataCenter, or comma-separated racks or nodes.**
5. **Duration**: The total runtime of the test.
6. **Result**: Displays the test's outcome, which can be:
    1. **NA**: Initial state, test not yet completed.

    2. **Failed**: If any circuit's BER status is not "Good." BER status is determined by comparing factors such as **SerDes Technology (16nm/7nm/5nm)**, **Link Speed Active**, **Active FEC**, **Raw BER**, and **Effective BER** against specific thresholds(the same logic used in the Underperforming link). Depending on these values, the status is categorized as **Good**, **Poor**, or **Marginal**.
    3. **Passed**: All circuits have a "Good" BER status.

7. **Status**: test status, and it could be one of the following values:
    1. **Running:** test is running.
    2. **Timeout:** The test running time exceeded the test duration
    3. **Finished:** The test is successfully finished, and the collector has received the collected Amber files.
    4. **Pending:** The test is in its initial state.
    5. **Stopped:** An error occurred while the test is running.
8. **Number of Failed Circuits**: The count of circuits with a "Failed" BER status.
9. **Best Circuit BER**: The lowest value of **Raw BER** or **Effective BER** during the test.
10. **Worst Circuit BER**: The highest value of **Raw BER** or **Effective BER** during the test.



Refresh Interval (sec) 300

Last Refresh 2025-04-06 16:43:31

Date	Name	User	Scope	Status	Durati...	Result	Number of Failed Circuits	Best Circuit BER	Worst Circuit BER
2023-01-19 20:02:45	Test1	admin	DataCenter	Running	3 min	NA	0	raw: 0 eff : 0	raw: 1e-10 eff : 0
2023-01-19 20:02:45	Test2	admin	DataCenter	Finished	125 min	Passed	0	raw: 0 eff : 0	raw: 1e-10 eff : 0
2023-01-19 20:02:45	Test3	admin	DataCenter	Timeout	125 min	Passed	3	raw: 0 eff : 0	raw: 1e-10 eff : 0

Viewing **1-3** of **3** 5

Create New Test

To initiate a new test, the user clicks the Create Test button, which opens a modal dialog.

The tool can create up to five tests. If the maximum limit is reached and a new test is initiated, the system will automatically remove the oldest test to maintain a maximum of five tests in memory.

Multiple tests should be able to run simultaneously if they are on different nodes. For example, if the **Golden BER Test** is already running on **[node1, node2]**, and an **Advanced Flapping Test** is requested for **[node1, node4]**, the flapping test will fail because **node1 is already occupied by the Golden BER Test**.

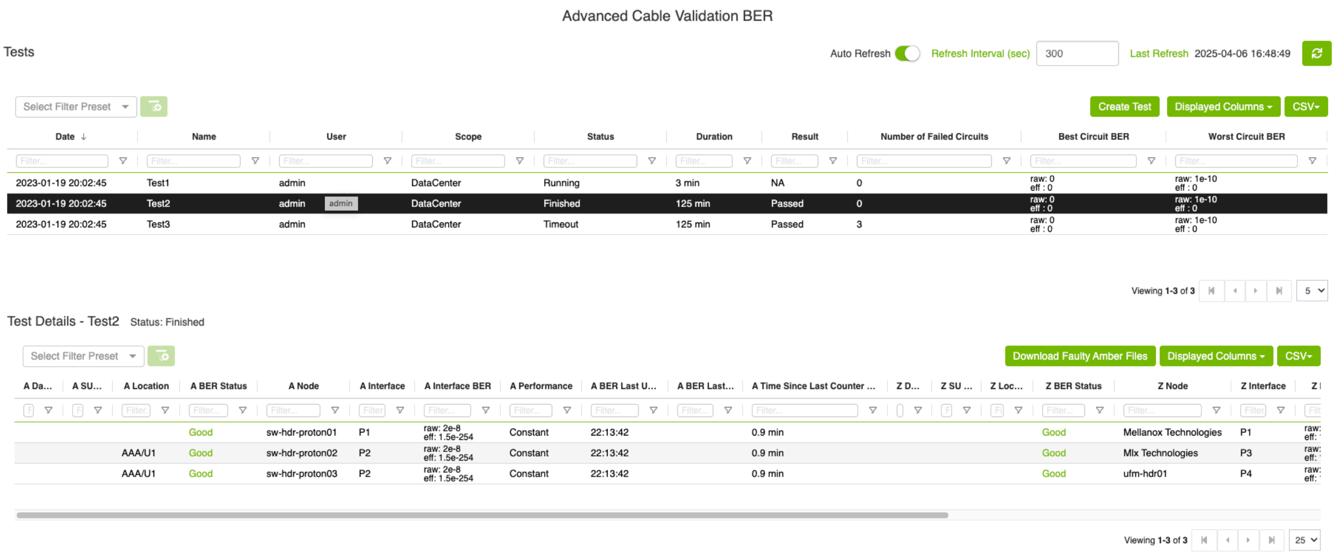
The collected amber will be saved for only 6 hours, after which it will be deleted.

Please note that the user should load the DC layout file in order to use the Scalable Unit and Data Hall options

After creation, the test is added to the table. Clicking on the test allows users to view its detailed results.

Note: If the test duration expires while the test is still incomplete due to errors or lack of responses from any agent within the test scope, a **5-minute buffer time** will be added. As a result, the total test duration will be **the original test time plus 5 minutes**.

Test Details



Advanced Cable Validation BER

Tests

Auto Refresh · Refresh Interval (sec) 300 · Last Refresh 2025-04-06 16:48:49

Date ↓	Name	User	Scope	Status	Duration	Result	Number of Failed Circuits	Best Circuit BER	Worst Circuit BER
2023-01-19 20:02:45	Test1	admin	DataCenter	Running	3 min	NA	0	raw: 0 eff : 0	raw: 1e-10 eff : 0
2023-01-19 20:02:45	Test2	admin	DataCenter	Finished	125 min	Passed	0	raw: 0 eff : 0	raw: 1e-10 eff : 0
2023-01-19 20:02:45	Test3	admin	DataCenter	Timeout	125 min	Passed	3	raw: 0 eff : 0	raw: 1e-10 eff : 0

Viewing **1-3** of **3**

Test Details - Test2 Status: Finished

A Da...	A SU...	A Location	A BER Status	A Node	A Interface	A Interface BER	A Performance	A BER Last U...	A BER Last...	A Time Since Last Counter ...	Z D...	Z SU ...	Z Loc...	Z BER Status	Z Node	Z Interface	Z
			Good	sw-hdr-proton01	P1	raw: 2e-8 eff: 1.5e-254	Constant	22:13:42		0.9 min				Good	Mellanox Technologies	P1	raw: eff:
		AAA/U1	Good	sw-hdr-proton02	P2	raw: 2e-8 eff: 1.5e-254	Constant	22:13:42		0.9 min				Good	Mlx Technologies	P3	raw: eff:
		AAA/U1	Good	sw-hdr-proton03	P2	raw: 2e-8 eff: 1.5e-254	Constant	22:13:42		0.9 min				Good	ufm-hdr01	P4	raw: eff:

Viewing **1-3** of **3**

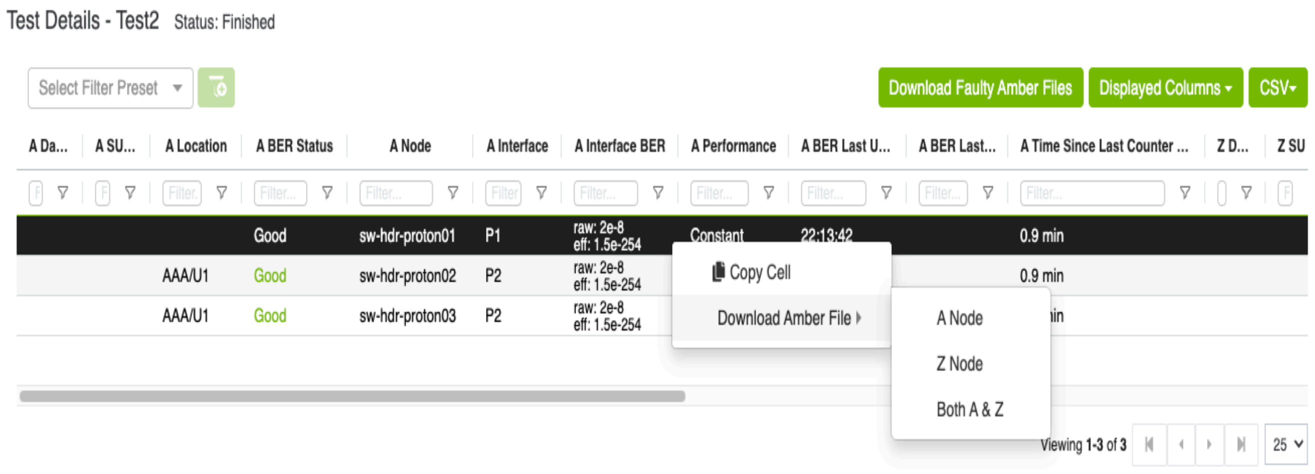
Detailed test results are displayed in a table, covering both **A** and **Z** endpoints. The columns include:

1. **Protocol**: circuit protocol (IB, Ethernet or NVLink).

2. **Data Hall / SU Number**: Data Hall and Scalable Unit
3. **Location**: Specifies the rack and unit location of the endpoint.
4. **BER Status**: Calculated based on the methodology mentioned earlier.
5. **Node**: The name of the node.
6. **Interface**: The port under analysis.
7. **BER Counters**: Displays **Raw BER** and **Effective BER** counters.
8. **Performance**: Indicates performance trends:
    - **NA**: No available **Raw BER** value or the first recorded value.
    - **Improved**: If the **Raw BER** value has decreased.
    - **Degraded**: If the **Raw BER** value has increased.
    - **Constant**: If the value remains unchanged.
9. **BER Last Update**: The timestamp of the last BER update.
10. **BER Last Update Duration**: Time since the most recent BER update.
11. **Time Since Last Counter Clear**: The elapsed time since the counters were last cleared.

Downloading Amber Files

After the test finished, user can download the Amber file for each circuit by clicking on the circuit then right-click

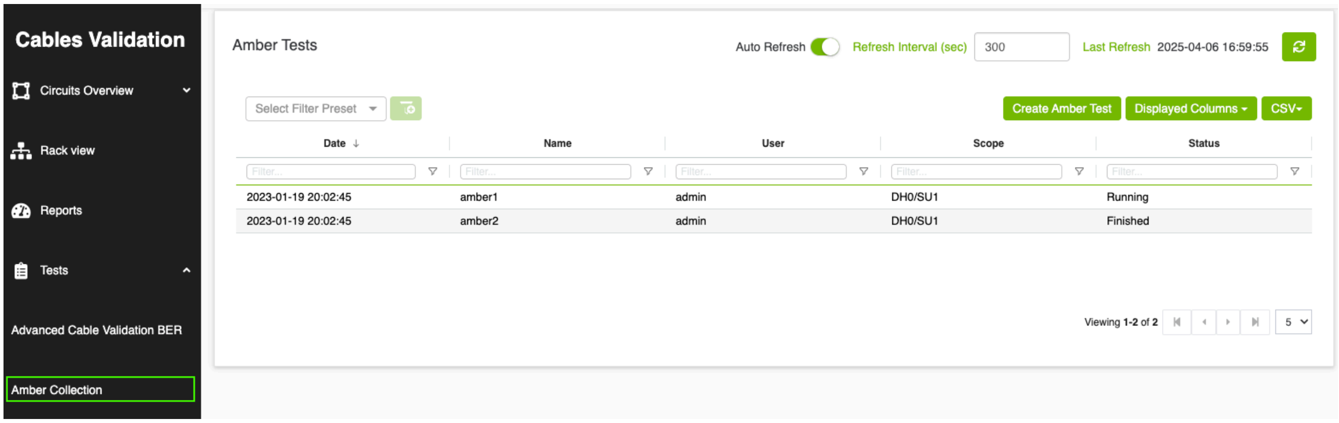

Test Details - Test2 Status: Finished

A Da...	A SU...	A Location	A BER Status	A Node	A Interface	A Interface BER	A Performance	A BER Last U...	A BER Last...	A Time Since Last Counter ...	Z D...	Z SU
			Good	sw-hdr-proton01	P1	raw: 2e-8 eff: 1.5e-254	Constant	22:13:42		0.9 min		
		AAA/U1	Good	sw-hdr-proton02	P2	raw: 2e-8 eff: 1.5e-254				0.9 min		
		AAA/U1	Good	sw-hdr-proton03	P2	raw: 2e-8 eff: 1.5e-254				in		

User can download the amber file for A, Z or both endpoints

Also, user can download faulty Amber files by click on **Download Faulty Amber Files** button

Amber Collection



Refresh Interval (sec) 300

Last Refresh 2025-04-06 16:59:55

Date	Name	User	Scope	Status
2023-01-19 20:02:45	amber1	admin	DH0/SU1	Running
2023-01-19 20:02:45	amber2	admin	DH0/SU1	Finished

Viewing 1-2 of 2

The **Amber Collection** test is designed to Collecting Amber File upon request. The process starts by utilizing the mlxlink command to gather amber files from connected agents within the specified test scope. Once all agents complete the collection and send the amber files to the collector, these files are compressed into a single tar file and stored in the directory:/cable_bringup_root/data/tests/amber/<test_name> and the file name will be <test_name>.tar.gz. Afterward, the collector marks the test as **Finished**.

In the UI, the test is presented in a tabular format with the following columns:

1. **Date**: The creation date of the test.

2. **Name**: The name assigned to the test.

3. **Scope**: Test scope, which usually includes **SU<n>/DH<n>, the DataCenter, or comma-separated racks or nodes.**

4. **Status:** The current status of the test, which can be one of the following:

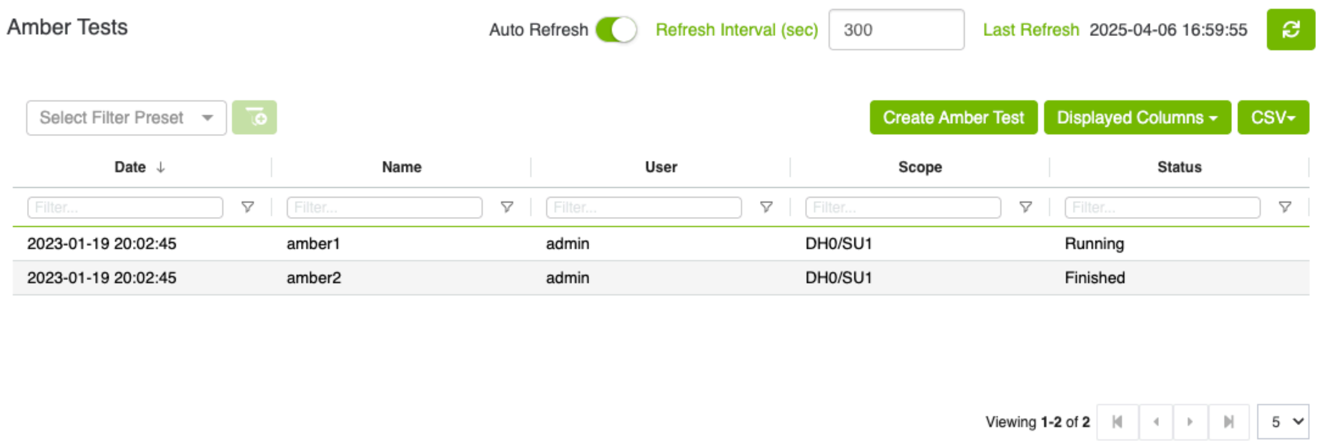
    1. **Running**: The test is ongoing, and not all agents have returned their amber files.

    2. **Timeout**: The test has exceeded the 5-minute time limit (test timeout = 5 min), and not all agents have returned their amber files.

    3. **Finished**: All agents have successfully returned their amber files.

    4. **Pending**: The test is in its initial state, awaiting execution.

    5. **Stopped:** An error occurred while the test is running.



Auto Refresh

Refresh Interval (sec) 300

Last Refresh 2025-04-06 16:59:55

Date ↓	Name	User	Scope	Status
2023-01-19 20:02:45	amber1	admin	DH0/SU1	Running
2023-01-19 20:02:45	amber2	admin	DH0/SU1	Finished

Viewing **1-2** of **2** 5

Create New Test

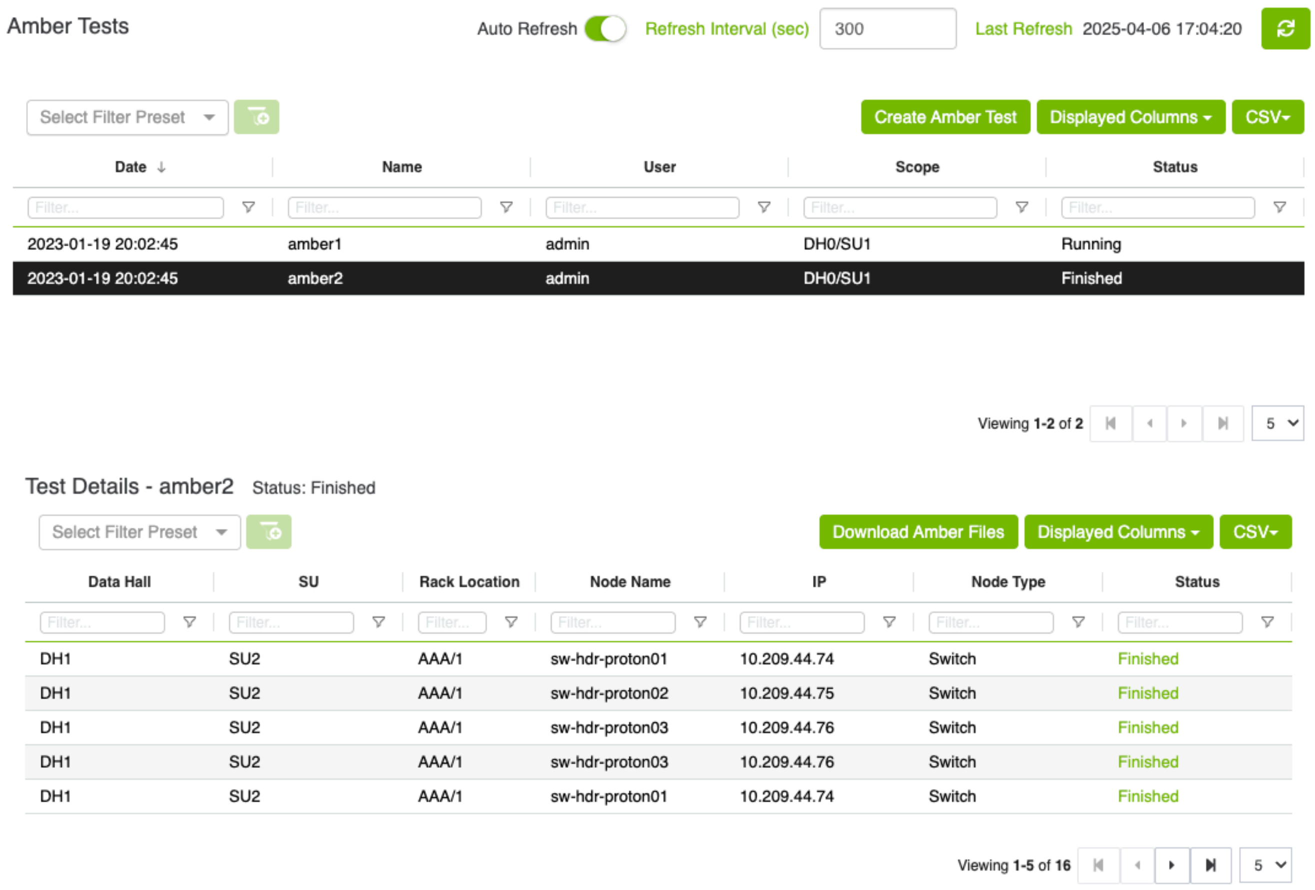
To initiate a new test, the user clicks the Create Test button, which opens a modal dialog.

The tool can create up to five tests. If the maximum limit is reached and a new test is initiated, the system will automatically remove the oldest test to maintain a maximum of five tests in memory.

Please note that the user should load the DC layout file in order to use the Scalable Unit and Data Hall options

After creation, the test is added to the table. Clicking on the test allows users to view its detailed results.

Test Details



Amber Tests

Refresh Interval (sec) 300

Last Refresh 2025-04-06 17:04:20

Date	Name	User	Scope	Status
2023-01-19 20:02:45	amber1	admin	DH0/SU1	Running
2023-01-19 20:02:45	amber2	admin	DH0/SU1	Finished

Viewing 1-2 of 2

Test Details - amber2 Status: Finished

Data Hall	SU	Rack Location	Node Name	IP	Node Type	Status
DH1	SU2	AAA/1	sw-hdr-proton01	10.209.44.74	Switch	Finished
DH1	SU2	AAA/1	sw-hdr-proton02	10.209.44.75	Switch	Finished
DH1	SU2	AAA/1	sw-hdr-proton03	10.209.44.76	Switch	Finished
DH1	SU2	AAA/1	sw-hdr-proton03	10.209.44.76	Switch	Finished
DH1	SU2	AAA/1	sw-hdr-proton01	10.209.44.74	Switch	Finished

Viewing 1-5 of 16

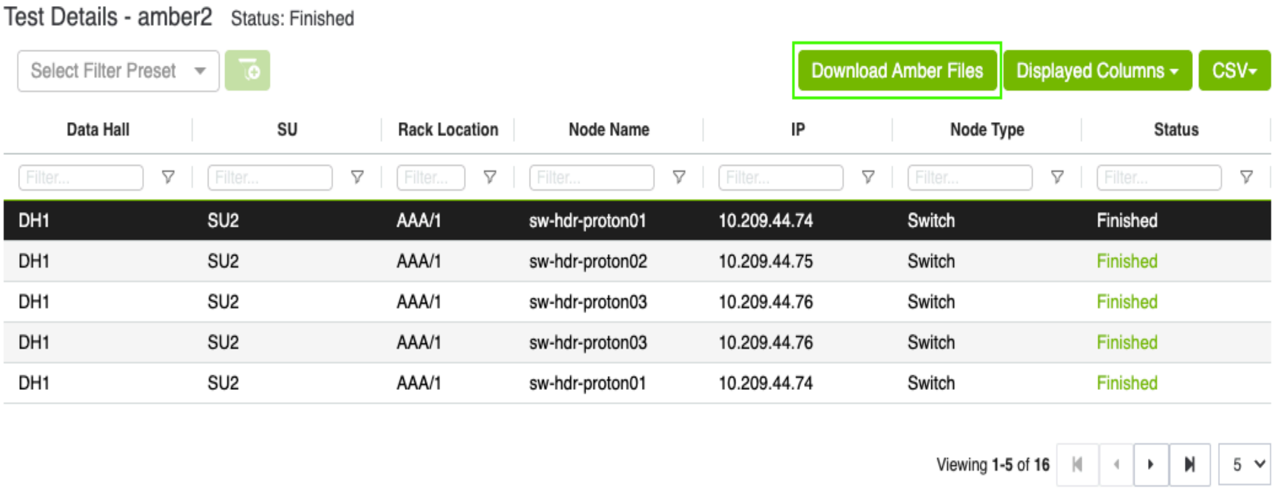
Detailed test results are displayed in a table, covering the node information where the test is running. The columns include:

1. **Data Hall / SU Number**: Data Hall and Scalable Unit
2. **Location**: Specifies the rack and unit location of the endpoint.
3. **Node Name**: The name of the node.
4. **IP**: The IP address.
5. **Node Type**: switch or host.
6. **Status**: Indicates the status of amber file collection for each agent:

-
  - **Running:** Amber file collection is in progress.
  - **Timeout:** The amber file collection has exceeded the 5-minute time limit without completion.
  - **Finished:** The amber file is successfully collected, and the collector has received it.
  - **Pending:** The test is in its initial state.
  - **Stopped:** An error occurred during the amber file collection process.

Downloading Amber Files

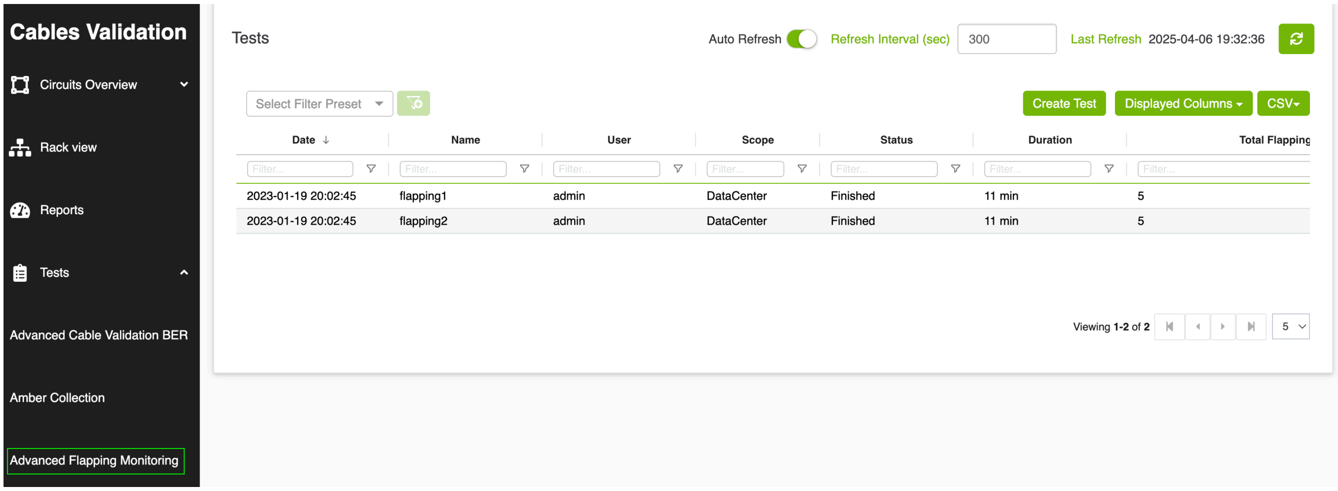
Once a test is marked as **Finished**, users can download the amber files by clicking the **Download Amber Files** button. This will download the compressed tar file containing all the collected amber files.



Test Details - amber2 Status: Finished

Data Hall	SU	Rack Location	Node Name	IP	Node Type	Status
DH1	SU2	AAA/1	sw-hdr-proton01	10.209.44.74	Switch	Finished
DH1	SU2	AAA/1	sw-hdr-proton02	10.209.44.75	Switch	Finished
DH1	SU2	AAA/1	sw-hdr-proton03	10.209.44.76	Switch	Finished
DH1	SU2	AAA/1	sw-hdr-proton03	10.209.44.76	Switch	Finished
DH1	SU2	AAA/1	sw-hdr-proton01	10.209.44.74	Switch	Finished

Advanced Flapping Monitoring



Refresh Interval (sec) 300

Last Refresh 2025-04-06 19:32:36

Date	Name	User	Scope	Status	Duration	Total Flapping
2023-01-19 20:02:45	flapping1	admin	DataCenter	Finished	11 min	5
2023-01-19 20:02:45	flapping2	admin	DataCenter	Finished	11 min	5

Viewing **1-2** of **2**

The **Advanced Flapping Monitoring** is designed to analyze the metrics that could lead to a flapping event. It tracks the circuit flapping events along with Bit Error Rates (BER), temperature, and Rx/Tx counters.

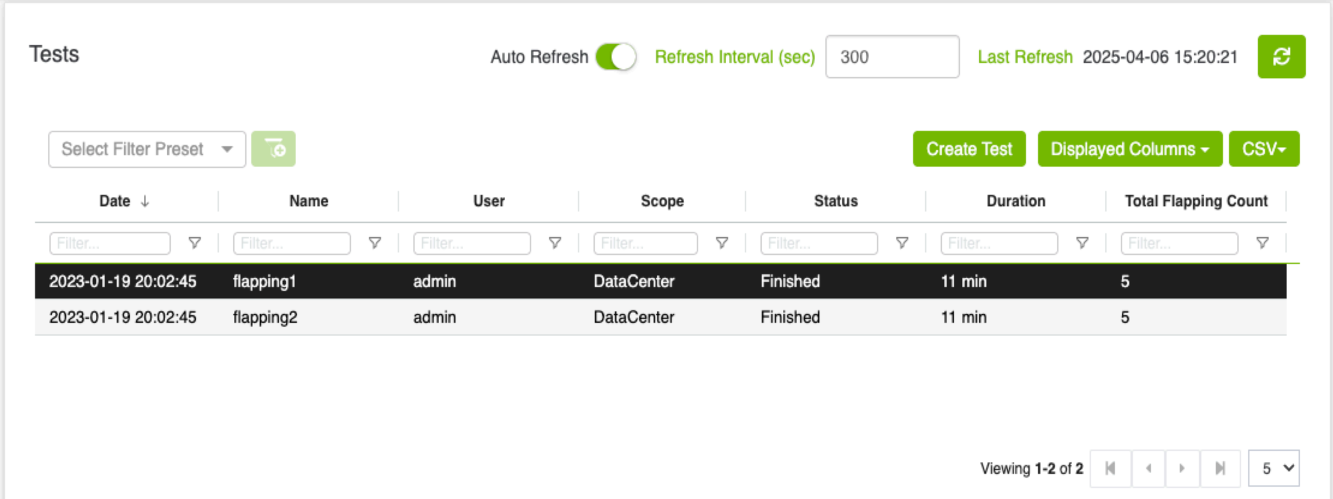
The process begins by resetting interface counters on connected agents using the command sudo mlxlink –pc, along with resetting the **flap counter** in **CVT memory**. Once initialized, the test runs for a user-defined duration, during which agents periodically report updated counter values to the collector every **10 minutes**.

In the UI, the test is presented in a tabular format with the following columns:

1. **Date**: The creation date of the test.
2. **Name**: The name assigned to the test.
3. **User**: The creator of the test.
4. **Scope**: Test scope, which usually includes **SU<n>/DH<n>, the DataCenter, or comma-separated racks or nodes.**
5. **Duration**: The total runtime of the test.
6. **Status**: test status, and it could be one of the following values:
    1. **Running:** test is running.
    2. **Timeout:** The test running time exceeded the test duration
    3. **Finished:** The test is successfully finished, and the collector has received the advanced stats.
    4. **Pending:** The test is in its initial state.

   5. **Stopped:** An error occurred while the test is running.

7. Total Flapping Count: The total flaps occurred across all switches for all ports from the time the test started.



Refresh Interval (sec) 300

Last Refresh 2025-04-06 15:20:21

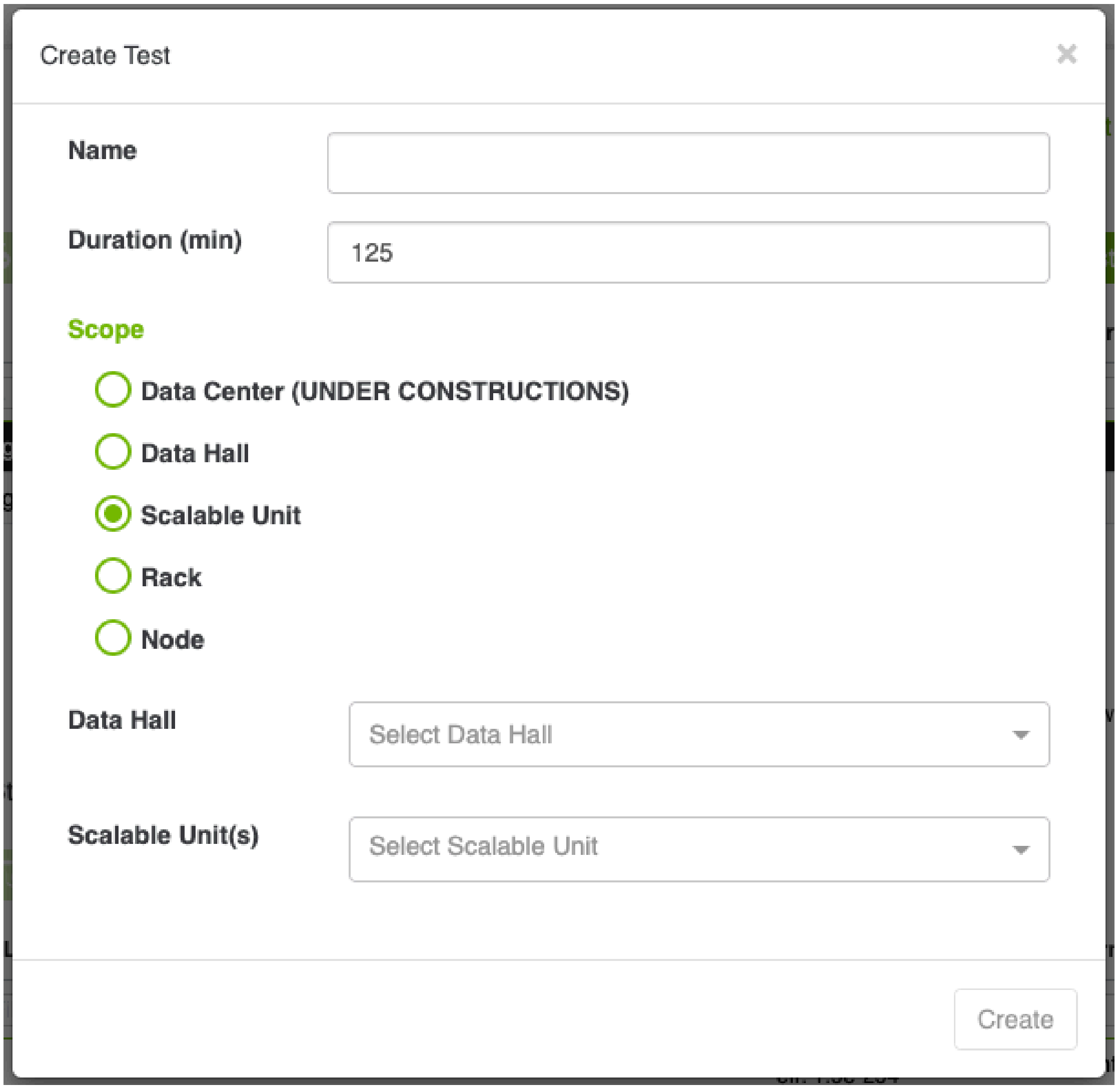
Date ↓	Name	User	Scope	Status	Duration	Total Flapping Count
2023-01-19 20:02:45	flapping1	admin	DataCenter	Finished	11 min	5
2023-01-19 20:02:45	flapping2	admin	DataCenter	Finished	11 min	5

Create New Test

To initiate a new test, the user clicks the Create Test button, which opens a modal dialog.

The tool can create up to five tests. If the maximum limit is reached and a new test is initiated, the system will automatically remove the oldest test to maintain a maximum of five tests in memory.

Multiple tests should be able to run simultaneously if they are on different nodes. For example, if the **Golden BER Test** is already running on **[node1, node2]**, and an **Advanced Flapping Test** is requested for **[node1, node4]**, the flapping test will fail because **node1 is already occupied by the Golden BER Test**.


Create Test

Name

Duration (min) 125

Scope

○ Data Center (UNDER CONSTRUCTIONS)

○ Data Hall

● Scalable Unit

○ Rack

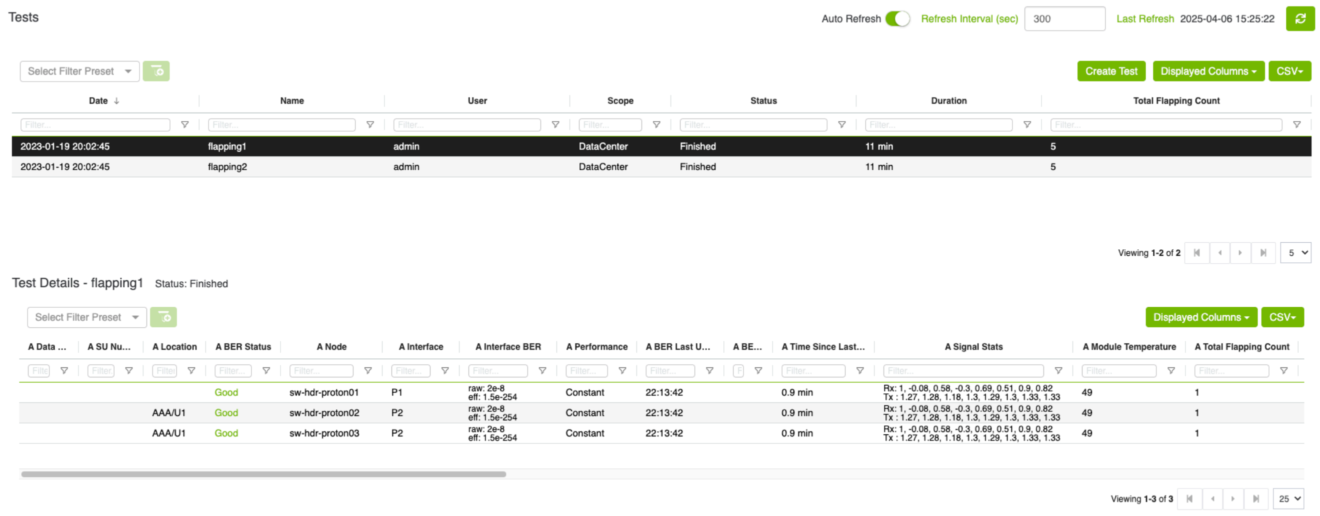
○ Node

Please note that the user should load the DC layout file in order to use the Scalable Unit and Data Hall options

After creation, the test is added to the table. Clicking on the test allows users to view its detailed results.

Note: If the test duration expires while the test is still incomplete due to errors or lack of responses from any agent within the test scope, a **5-minute buffer time** will be added. As a result, the total test duration will be **the original test time plus 5 minutes**.

Test Details



Date	Name	User	Scope	Status	Duration	Total Flapping Count
2023-01-19 20:02:45	flapping1	admin	DataCenter	Finished	11 min	5
2023-01-19 20:02:45	flapping2	admin	DataCenter	Finished	11 min	5

A Data ...	A SU Nu...	A Location	A BER Status	A Node	A Interface	A Interface BER	A Performance	A BER Last U...	A BE...	A Time Since Last...	A Signal Stats	A Module Temperature	A Total Flapping Count
			Good	sw-hdr-proton01	P1	raw: 2e-8 eff: 1.5e-254	Constant	22:13:42		0.9 min	Rx: 1, -0.08, 0.58, -0.3, 0.69, 0.51, 0.9, 0.82 Tx : 1.27, 1.28, 1.18, 1.3, 1.29, 1.3, 1.33, 1.33	49	1
		AAA/U1	Good	sw-hdr-proton02	P2	raw: 2e-8 eff: 1.5e-254	Constant	22:13:42		0.9 min	Rx: 1, -0.08, 0.58, -0.3, 0.69, 0.51, 0.9, 0.82 Tx : 1.27, 1.28, 1.18, 1.3, 1.29, 1.3, 1.33, 1.33	49	1
		AAA/U1	Good	sw-hdr-proton03	P2	raw: 2e-8 eff: 1.5e-254	Constant	22:13:42		0.9 min	Rx: 1, -0.08, 0.58, -0.3, 0.69, 0.51, 0.9, 0.82 Tx : 1.27, 1.28, 1.18, 1.3, 1.29, 1.3, 1.33, 1.33	49	1

Detailed test results are displayed in a table, covering both **A** and **Z** endpoints. The columns include:

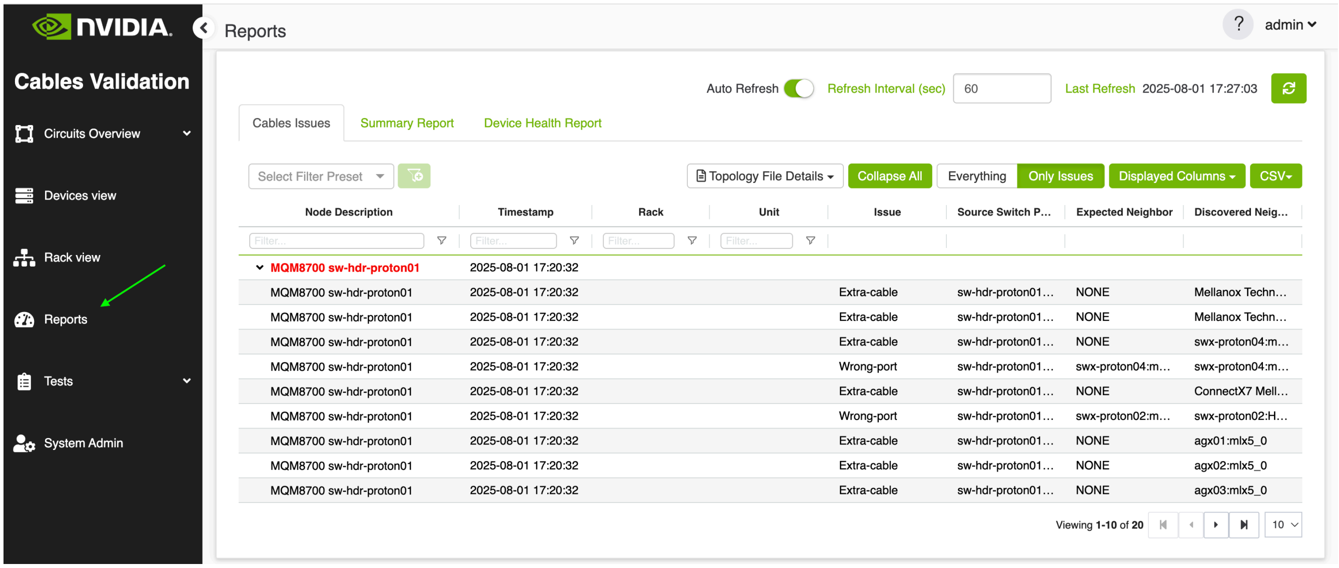
1. **Protocol**: circuit protocol (IB, Ethernet or NVLink).
2. **Data Hall / SU Number**: Data Hall and Scalable Unit
3. **Location**: Specifies the rack and unit location of the endpoint.
4. **BER Status**: BER status is determined by comparing factors such as **SerDes Technology (16nm/7nm/5nm)**, **Link Speed Active**, **Active FEC**, **Raw BER**, and **Effective BER** against specific thresholds(the same logic used in the Underperforming link). Depending on these values, the status is categorized as **Good**, **Poor**, or **Marginal**.
5. **Node**: The name of the node.
6. **Interface**: The port under analysis.
7. **BER Counters**: Displays **Raw BER** and **Effective BER** counters.
8. **Performance**: Indicates performance trends:
    1. **NA**: No available **Raw BER** value or the first recorded value.
    2. **Improved**: If the **Raw BER** value has decreased.
    3. **Degraded**: If the **Raw BER** value has increased.
    4. **Constant**: If the value remains unchanged.

9. **BER Last Update**: The timestamp of the last BER update.
10. **BER Last Update Duration**: Time since the most recent BER update.
11. **Time Since Last Counter Clear**: The elapsed time since the counters were last cleared.
12. **Signal Stats:** Shows **transmission and** reception power levels (Rx/Tx Power Lane)
13. **Module Temperature:** module temperature value.
14. **Total Flapping Count:** This is the total number of transitions occurring **during the test.**
15. **Flapping Status**: flapping status and it could be one of the following values:
    1. Ok - no flapping events during the test.
    2. Flapping - agent detected a flapping event in the last 1 minute
    3. Flapped - agent detected a flapping event at some point while the test is running (The Flapping event counter >= 1)
16. **Flap 30 sec:** how many flaps happened in the last 30 seconds (hidden by default).
17. **Flap 1 min:** how many flaps happened in the last 1 minute (hidden by default).
18. **Flap 5 min:** how many flaps happened in the last 5 minutes (hidden by default).
19. **Flap 1 hour:** how many flaps happened in the last 1 hour (hidden by default).
20. **Flap 12 hour:** how many flaps happened in the last 12 hours (hidden by default).
21. **Flap 24 hour:** how many flaps happened in the last 24 hours (hidden by default).

Reports

Cables Issues

This tab is only supported for Admin users



Auto Refresh ● Refresh Interval (sec) 60 Last Refresh 2025-08-01 17:27:03

Node Description	Timestamp	Rack	Unit	Issue	Source Switch P...	Expected Neighbor	Discovered Neig...
MQM8700 sw-hdr-proton01	2025-08-01 17:20:32						
MQM8700 sw-hdr-proton01	2025-08-01 17:20:32			Extra-cable	sw-hdr-proton01...	NONE	Mellanox Techn...
MQM8700 sw-hdr-proton01	2025-08-01 17:20:32			Extra-cable	sw-hdr-proton01...	NONE	Mellanox Techn...
MQM8700 sw-hdr-proton01	2025-08-01 17:20:32			Extra-cable	sw-hdr-proton01...	NONE	swx-proton04:m...
MQM8700 sw-hdr-proton01	2025-08-01 17:20:32			Wrong-port	sw-hdr-proton01...	swx-proton04:m...	swx-proton04:m...
MQM8700 sw-hdr-proton01	2025-08-01 17:20:32			Extra-cable	sw-hdr-proton01...	NONE	ConnectX7 Mell...
MQM8700 sw-hdr-proton01	2025-08-01 17:20:32			Wrong-port	sw-hdr-proton01...	swx-proton02:m...	swx-proton02:H...
MQM8700 sw-hdr-proton01	2025-08-01 17:20:32			Extra-cable	sw-hdr-proton01...	NONE	agx01:mlx5_0
MQM8700 sw-hdr-proton01	2025-08-01 17:20:32			Extra-cable	sw-hdr-proton01...	NONE	agx02:mlx5_0
MQM8700 sw-hdr-proton01	2025-08-01 17:20:32			Extra-cable	sw-hdr-proton01...	NONE	agx03:mlx5_0

Viewing **1-10** of **20** 10

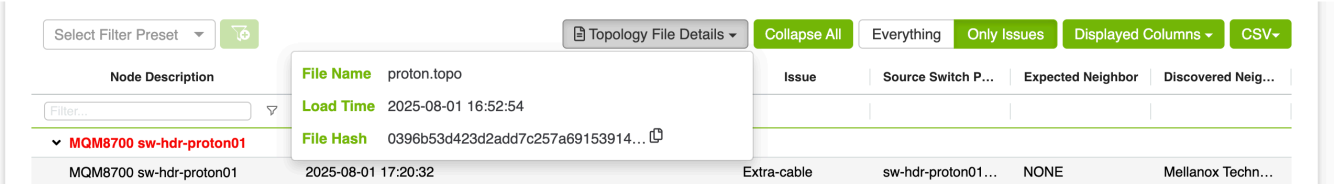
The **Cable Issues View** provides a table-based overview of node related issues, enabling users to quickly identify and resolve problems. The view organizes all issues related to a specific node under a single collapsible row.

By default, all rows are fully expanded, showing details for each issue, users can manually collapse individual nodes to hide their associated issues or click the **'Collapse All'** button to collapse all nodes at once.

Only nodes with issues are displayed by default. And the description of a node with issues is highlighted in **red** for easy identification.

To view all nodes, including those without issues, click the **'Everything'** button.

To see topology details file you can click on '**Topology File Details**' which include the File Name, Load Time and the File Hash.



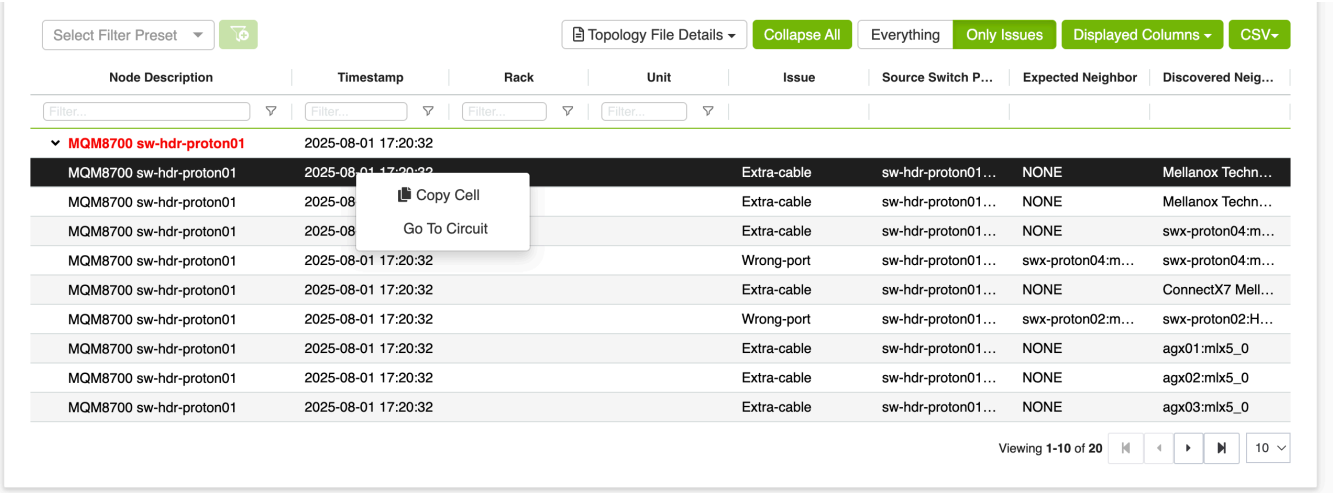
File Name proton.topo
Load Time 2025-08-01 16:52:54
File Hash 0396b53d423d2add7c257a69153914...

Node Description	Issue	Source Switch P...	Expected Neighbor	Discovered Neig...
MQM8700 sw-hdr-proton01				
MQM8700 sw-hdr-proton01 2025-08-01 17:20:32	Extra-cable	sw-hdr-proton01...	NONE	Mellanox Techn...

Navigate to Circuit

Users can directly navigate to the circuit associated with a specific issue:

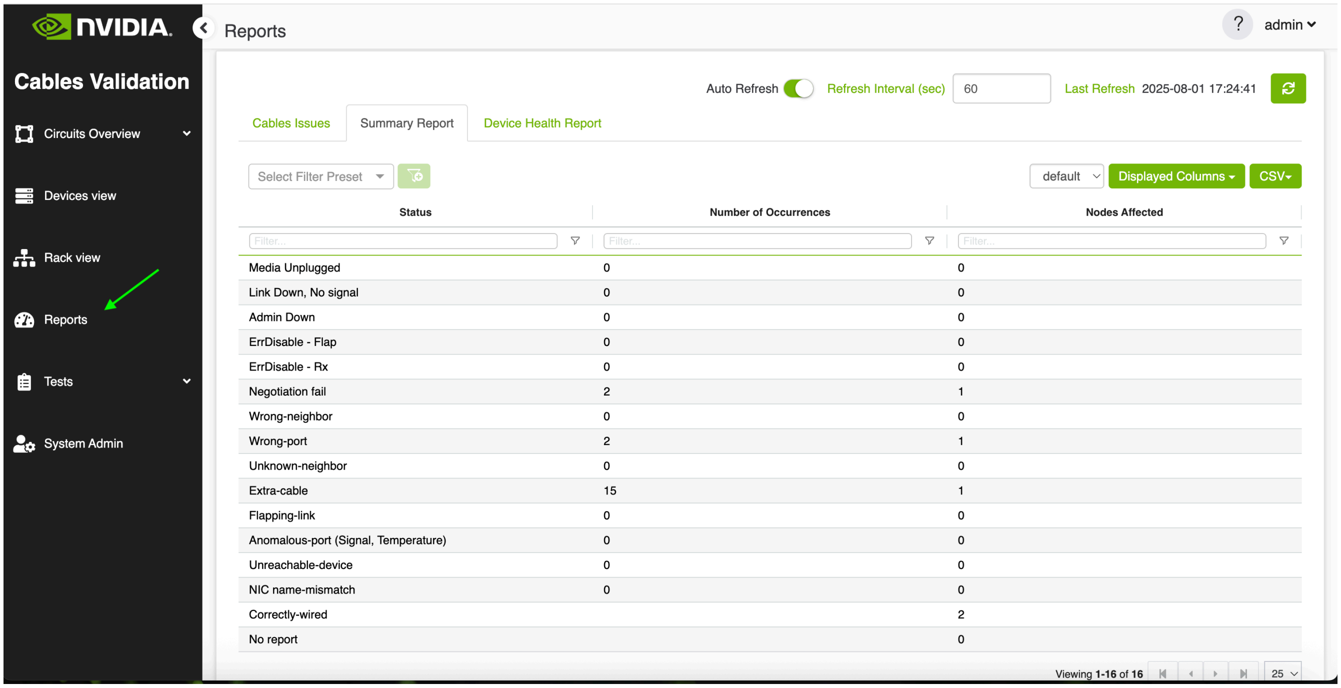
1. **Right-click** on the issue in the table.
2. Select **'Go To Circuit'** to open the circuit view for that issue.



Node Description	Timestamp	Rack	Unit	Issue	Source Switch P…	Expected Neighbor	Discovered Neig…
MQM8700 sw-hdr-proton01	2025-08-01 17:20:32						
MQM8700 sw-hdr-proton01	2025-08-01 17:20:32			Extra-cable	sw-hdr-proton01…	NONE	Mellanox Techn…
MQM8700 sw-hdr-proton01	2025-08			Extra-cable	sw-hdr-proton01…	NONE	Mellanox Techn…
MQM8700 sw-hdr-proton01	2025-08			Extra-cable	sw-hdr-proton01…	NONE	swx-proton04:m…
MQM8700 sw-hdr-proton01	2025-08-01 17:20:32			Wrong-port	sw-hdr-proton01…	swx-proton04:m…	swx-proton04:m…
MQM8700 sw-hdr-proton01	2025-08-01 17:20:32			Extra-cable	sw-hdr-proton01…	NONE	ConnectX7 Mell…
MQM8700 sw-hdr-proton01	2025-08-01 17:20:32			Wrong-port	sw-hdr-proton01…	swx-proton02:m…	swx-proton02:H…
MQM8700 sw-hdr-proton01	2025-08-01 17:20:32			Extra-cable	sw-hdr-proton01…	NONE	agx01:mlx5_0
MQM8700 sw-hdr-proton01	2025-08-01 17:20:32			Extra-cable	sw-hdr-proton01…	NONE	agx02:mlx5_0
MQM8700 sw-hdr-proton01	2025-08-01 17:20:32			Extra-cable	sw-hdr-proton01…	NONE	agx03:mlx5_0

Summary Report

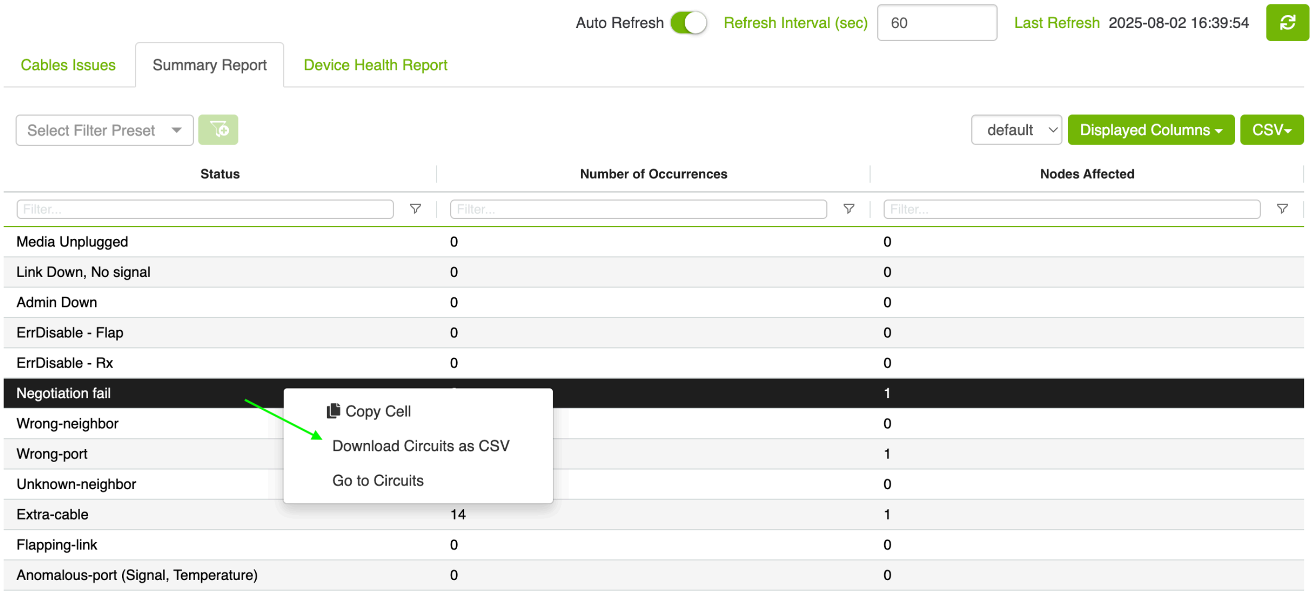
The **Summary Report View** provides a table summarizing all detected issues or syndromes across the tool. The table includes the number of occurrences and number of affected nodes for each issue.



Status	Number of Occurrences	Nodes Affected
Media Unplugged	0	0
Link Down, No signal	0	0
Admin Down	0	0
ErrDisable - Flap	0	0
ErrDisable - Rx	0	0
Negotiation fail	2	1
Wrong-neighbor	0	0
Wrong-port	2	1
Unknown-neighbor	0	0
Extra-cable	15	1
Flapping-link	0	0
Anomalous-port (Signal, Temperature)	0	0
Unreachable-device	0	0
NIC name-mismatch	0	0
Correctly-wired		2
No report		0

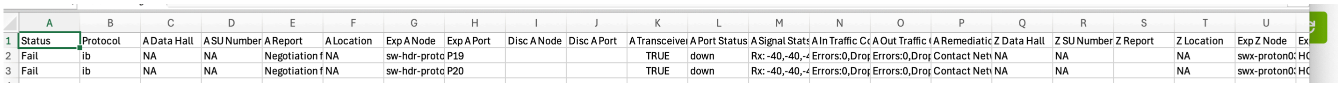
Download Circuits as CSV

Users can download all circuits associated with a specific syndrome as a CSV file by right-clicking on the syndrome and selecting **"Download as CSV."**



Status	Number of Occurrences	Nodes Affected
Media Unplugged	0	0
Link Down, No signal	0	0
Admin Down	0	0
ErrDisable - Flap	0	0
ErrDisable - Rx	0	0
Negotiation fail		1
Wrong-neighbor		0
Wrong-port		1
Unknown-neighbor		0
Extra-cable	14	1
Flapping-link	0	0
Anomalous-port (Signal, Temperature)	0	0

For example, selecting **"Download as CSV"** for the *Negotiation Fail* syndrome will generate a file that includes only circuits with that specific issue.

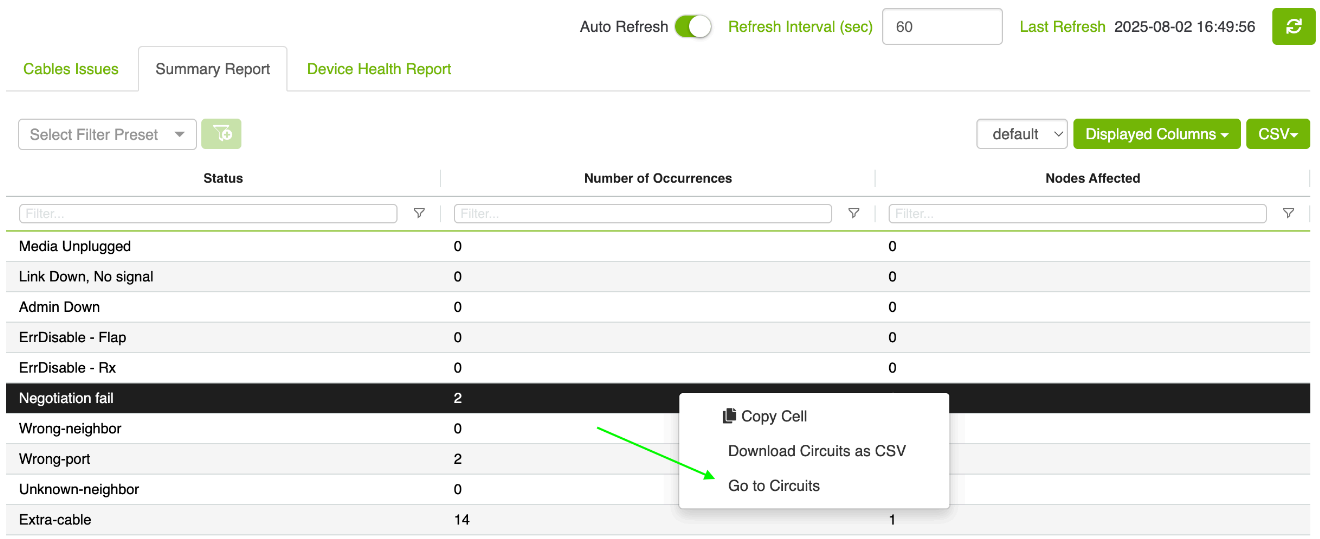


	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	
1	Status	Protocol	A Data Hall	A SU Number	A Report	A Location	Exp A Node	Exp A Port	Disc A Node	Disc A Port	A Transceiver	A Port Status	A Signal Stats	A In Traffic Co	A Out Traffic (	A Remediatic	Z Data Hall	Z SU Number	Z Report	Z Location	Exp Z Node	Ex
2	Fail	ib	NA	NA	Negotiation f	NA	sw-hdr-proto	P19			TRUE	down	Rx: -40,-40,-4	Errors:0,Drop	Errors:0,Drop	Contact Netv	NA	NA		NA	swx-proton0:	HC
3	Fail	ib	NA	NA	Negotiation f	NA	sw-hdr-proto	P20			TRUE	down	Rx: -40,-40,-4	Errors:0,Drop	Errors:0,Drop	Contact Netv	NA	NA		NA	swx-proton0:	HC

Go to Circuits

Users can also navigate to the **Circuits View** filtered by a specific syndrome by right-clicking on the syndrome and selecting **"Go to Circuits."**

Within the Circuits View, users can further download the filtered data as a CSV file if needed.



Auto Refresh

Refresh Interval (sec) 60

Last Refresh 2025-08-02 16:49:56

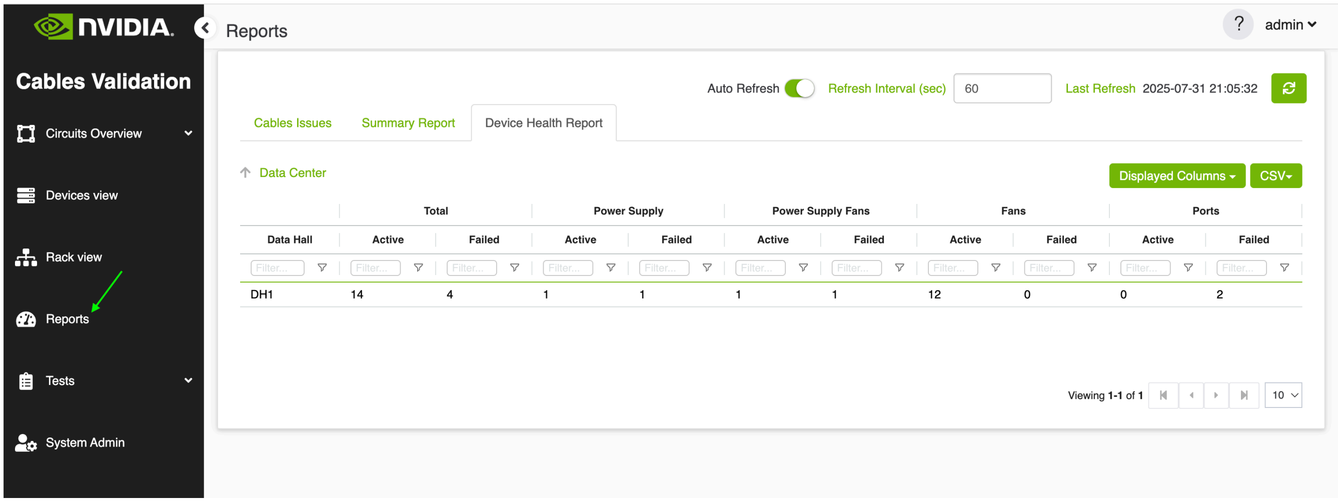
Status	Number of Occurrences	Nodes Affected
Media Unplugged	0	0
Link Down, No signal	0	0
Admin Down	0	0
ErrDisable - Flap	0	0
ErrDisable - Rx	0	0
Negotiation fail	2	
Wrong-neighbor	0	
Wrong-port	2	
Unknown-neighbor	0	
Extra-cable	14	1

Device Health Report

The **Device Health Report** provides a comprehensive, hierarchical view of device health status across the data center infrastructure. It enables users to monitor key device components—such as power supplies, fans, power supply fans, and ports—across various organizational levels, including Data Center, Data Hall, Scalable Unit, and Rack. Through a drill-down interface, users can navigate these levels to examine the health status in detail. At each level, the report displays values indicating the number of active and failed components (e.g., fans, power supplies) within the corresponding Data Hall, Scalable Unit, Rack, or individual device—depending on the selected level.

The feature displays device health information in a tabular format with the ability to:

- Navigate through hierarchical levels (DC → DH → SU → Rack)
- View detailed health status for Power Supply, Power Supply Fans, Fans, and Ports
- visual indicators for device status



Refresh Interval (sec) 60

Last Refresh 2025-07-31 21:05:32

	Total		Power Supply		Power Supply Fans		Fans		Ports	
Data Hall	Active	Failed	Active	Failed	Active	Failed	Active	Failed	Active	Failed
DH1	14	4	1	1	1	1	12	0	0	2

Topology Levels

The system provides four levels of hierarchy, listed from the largest to the smallest:

1. **Data Center (DC):** This level is always available and displays data based on the loaded topology. For example, if the user loads a DC floor layout file, the DC level will show all Data Halls (DHs) in the topology.

   This level will be described in detail later in this document.

2. **Data Hall (DH):**

   Displays all Scalable Units (SUs) within the selected Data Hall.

   This level is not always available and is only accessible when a DC floor layout file is loaded.

3. **Scalable Unit (SU):**

   Displays all Racks within the selected SU.

   Like the DH level, this level is also only available if a DC floor layout file has been loaded.

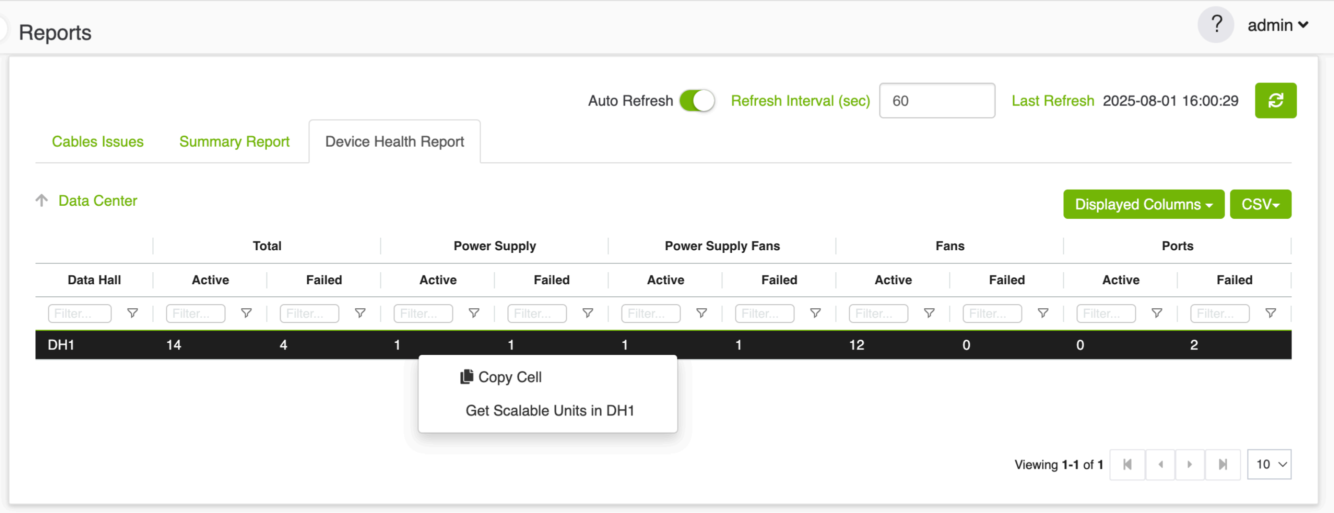
4. **Rack:**

   Displays all nodes within the selected Rack.

   This level is not always available either; it becomes accessible when the topology is loaded in P2P format.

Navigating Deeper

Users can right-click on a row to drill down into a more detailed view.

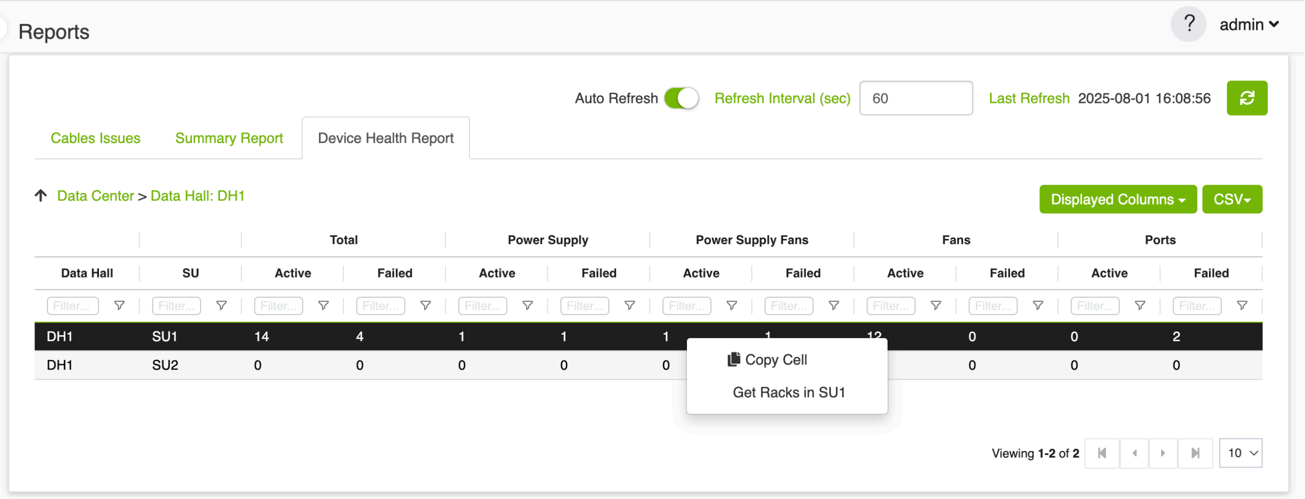


Auto Refresh
Refresh Interval (sec) 60
Last Refresh 2025-08-01 16:00:29

	Total		Power Supply		Power Supply Fans		Fans		Ports	
Data Hall	Active	Failed	Active	Failed	Active	Failed	Active	Failed	Active	Failed
DH1	14	4	1	1	1	1	12	0	0	2

Viewing 1-1 of 1 10

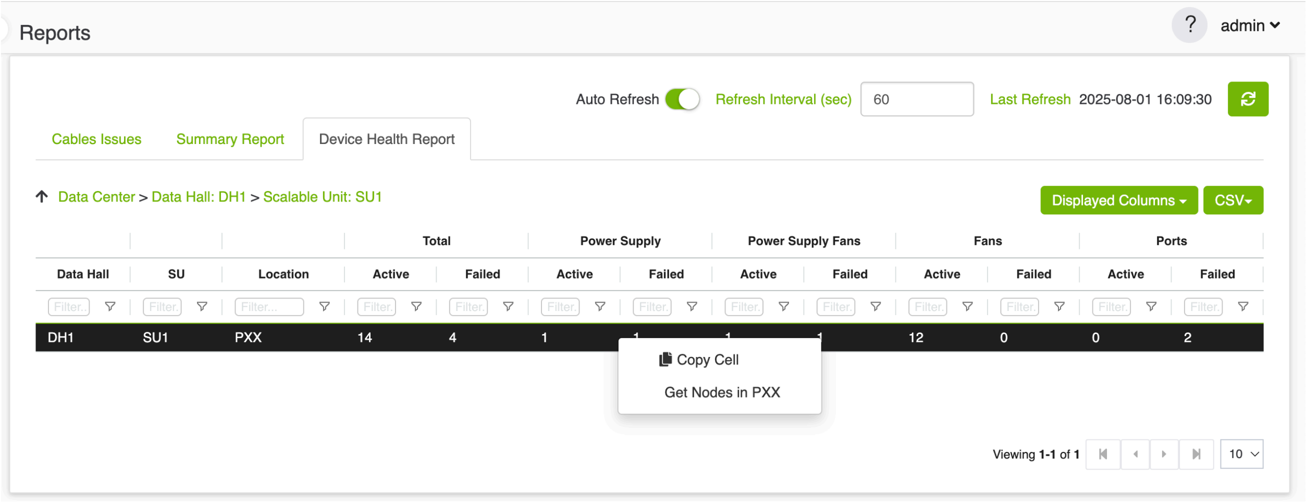
After selecting "Get sus in DH1", the view changes to show all SUs in DH1. user can keep drilling down until they reach the last available level/scope



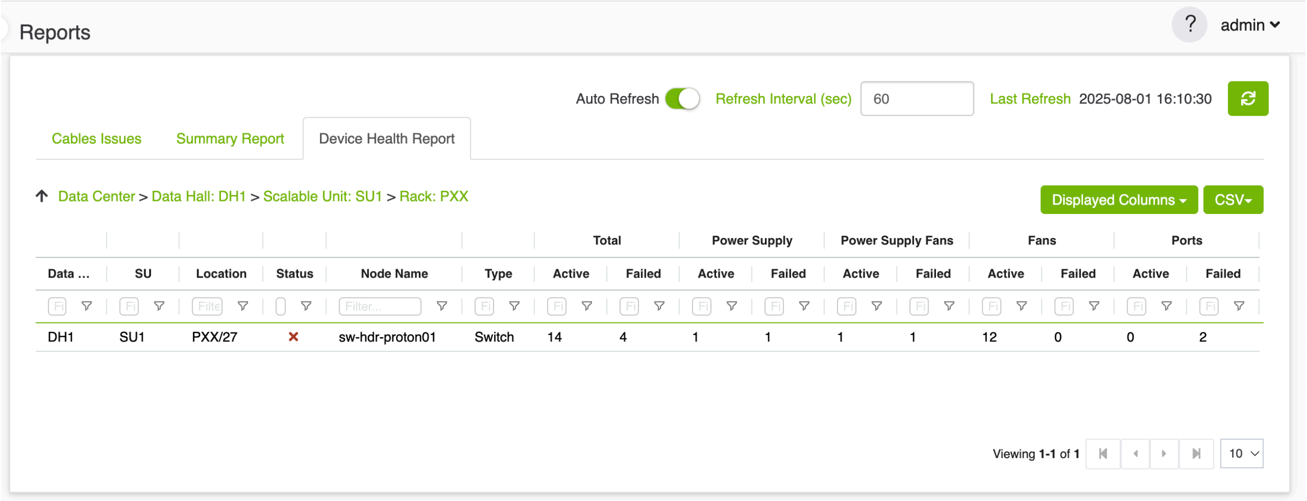
Auto Refresh
Refresh Interval (sec) 60
Last Refresh 2025-08-01 16:08:56

		Total		Power Supply		Power Supply Fans		Fans		Ports	
Data Hall	SU	Active	Failed	Active	Failed	Active	Failed	Active	Failed	Active	Failed
DH1	SU1	14	4	1	1	1	1	12	0	0	2
DH1	SU2	0	0	0	0	0			0	0	0

Viewing 1-2 of 2 10



			Total		Power Supply		Power Supply Fans		Fans		Ports	
Data Hall	SU	Location	Active	Failed	Active	Failed	Active	Failed	Active	Failed	Active	Failed
DH1	SU1	PXX	14	4	1	1	1	1	12	0	0	2



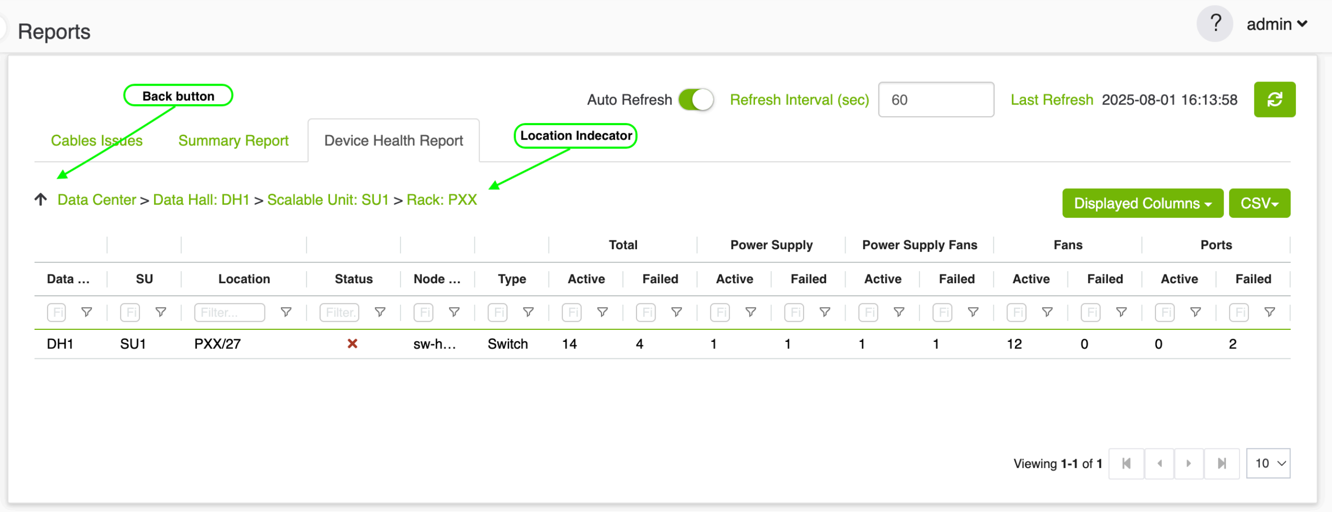
						Total		Power Supply		Power Supply Fans		Fans		Ports	
Data ...	SU	Location	Status	Node Name	Type	Active	Failed	Active	Failed	Active	Failed	Active	Failed	Active	Failed
DH1	SU1	PXX/27	✗	sw-hdr-proton01	Switch	14	4	1	1	1	1	12	0	0	2

Location Indicator

On the left side of the table, a location indicator shows the current path within the hierarchy.

Users can:

- Click on any level in the path to go back to it.
- Use the Back button to return to the previous level.



						Total		Power Supply		Power Supply Fans		Fans		Ports	
Data ...	SU	Location	Status	Node ...	Type	Active	Failed	Active	Failed	Active	Failed	Active	Failed	Active	Failed
DH1	SU1	PXX/27	✗	sw-h…	Switch	14	4	1	1	1	1	12	0	0	2

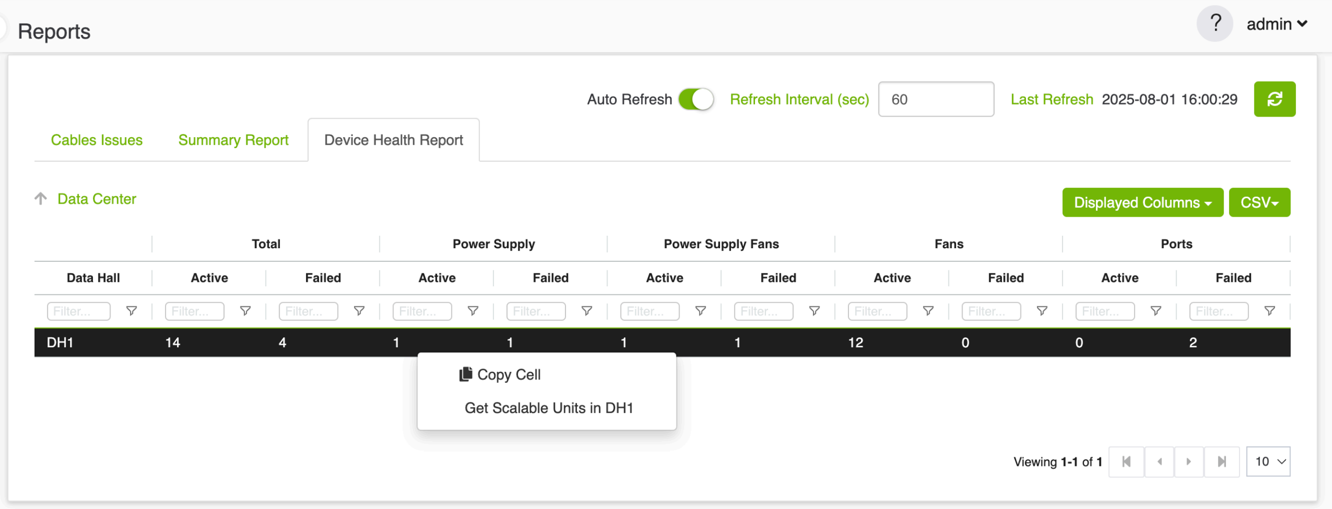
Initial Level — Data Center Level (DC)

Not all topologies in CVT include Data Halls (DHs), Scalable Units (SUs), and Racks — it depends on the type of topology the user loads.

The system initially adjusts the hierarchy based on available data.

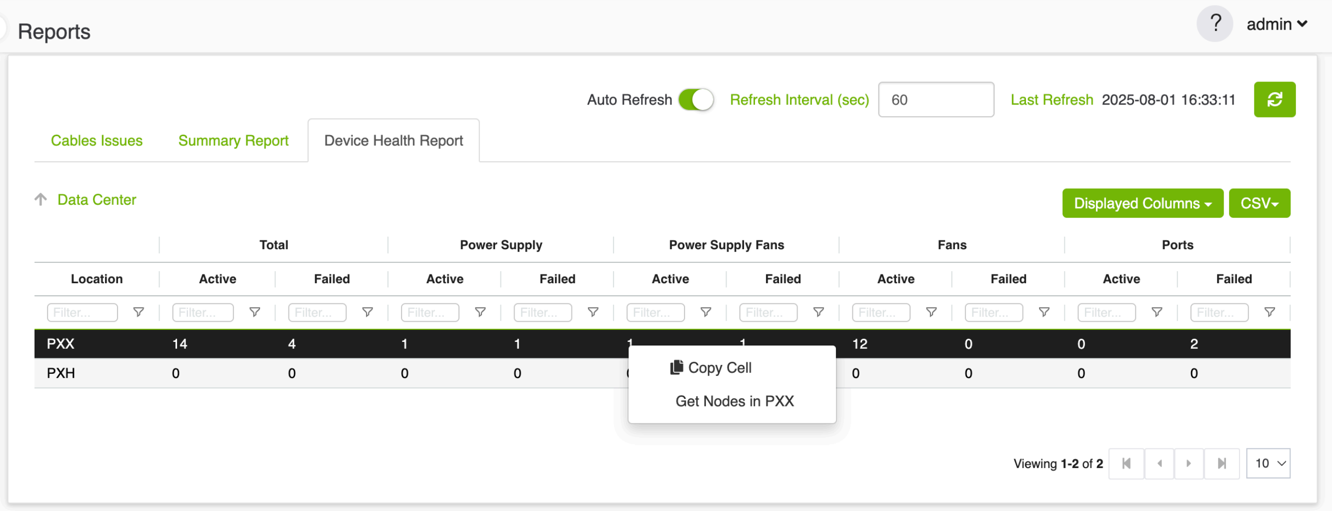
For example:

- If the user loads a **P2P file along with a DC floor layout**, the topology will include all levels: **DC → DHs → SUs → Racks**.



	Total		Power Supply		Power Supply Fans		Fans		Ports	
Data Hall	Active	Failed	Active	Failed	Active	Failed	Active	Failed	Active	Failed
DH1	14	4	1	1	1	1	12	0	0	2

- If the user loads a **P2P file without the DC floor layout**, the topology will only include **DC → Racks**.

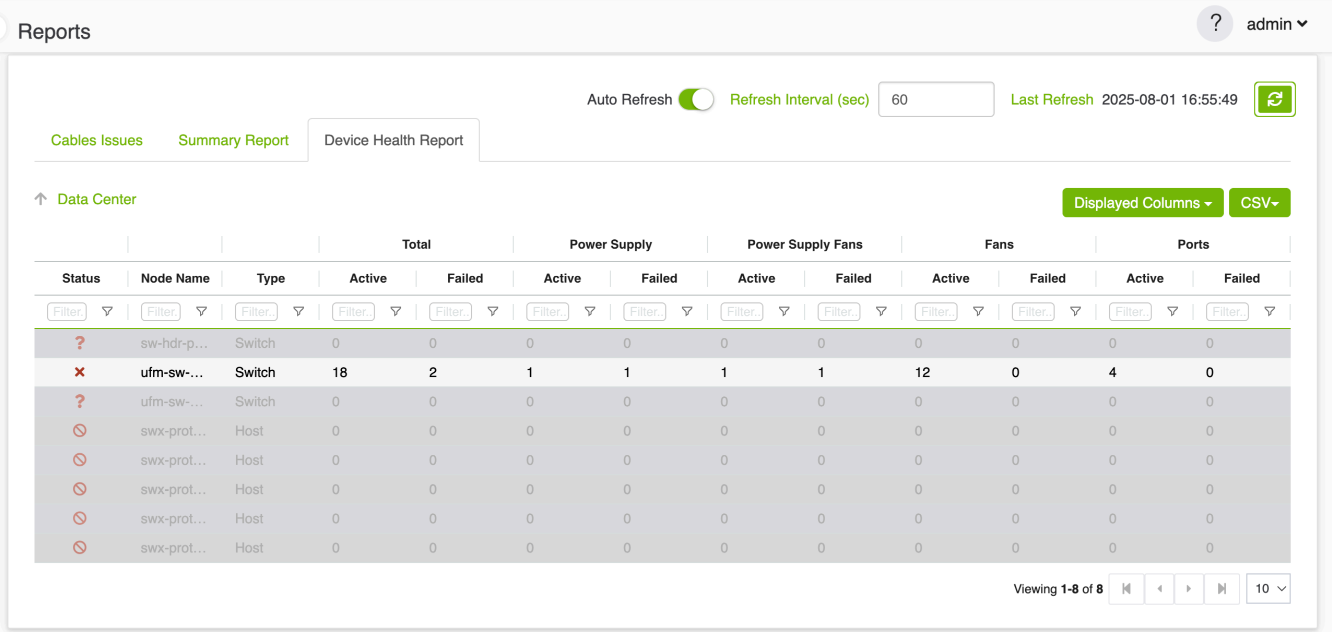


Auto Refresh: on | Refresh Interval (sec): 60 | Last Refresh: 2025-08-01 16:33:11

	Total		Power Supply		Power Supply Fans		Fans		Ports	
Location	Active	Failed	Active	Failed	Active	Failed	Active	Failed	Active	Failed
PXX	14	4	1	1	1	1	12	0	0	2
PXH	0	0	0	0			0	0	0	0

Viewing **1-2** of **2** | 10

- If the user loads a **topo file**, the topology will not support racks at all, so the available level will be only **DC**.



Auto Refresh: on | Refresh Interval (sec): 60 | Last Refresh: 2025-08-01 16:55:49

			Total		Power Supply		Power Supply Fans		Fans		Ports	
Status	Node Name	Type	Active	Failed	Active	Failed	Active	Failed	Active	Failed	Active	Failed
?	sw-hdr-p…	Switch	0	0	0	0	0	0	0	0	0	0
✕	ufm-sw-…	Switch	18	2	1	1	1	1	12	0	4	0
?	ufm-sw-…	Switch	0	0	0	0	0	0	0	0	0	0
⃠	swx-prot…	Host	0	0	0	0	0	0	0	0	0	0
⃠	swx-prot…	Host	0	0	0	0	0	0	0	0	0	0
⃠	swx-prot…	Host	0	0	0	0	0	0	0	0	0	0
⃠	swx-prot…	Host	0	0	0	0	0	0	0	0	0	0
⃠	swx-prot…	Host	0	0	0	0	0	0	0	0	0	0

Viewing **1-8** of **8** | 10

Note

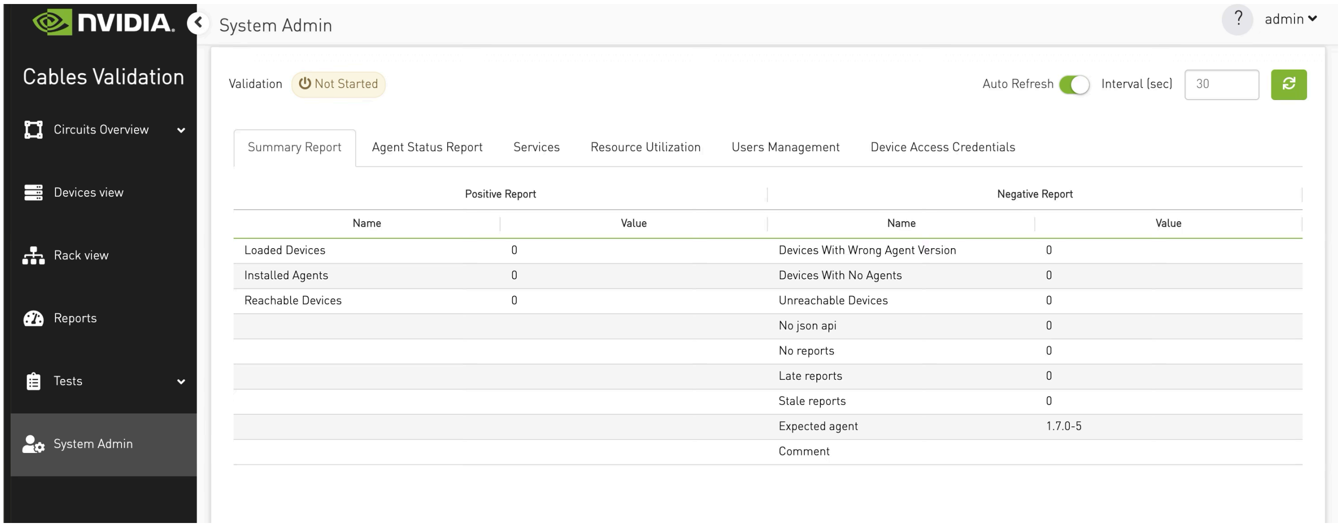
The Status column represents the agent status. For more information, please refer to Agent Status.

System Admin

This page is only supported for Admin users

Summary Report

The **Summary Report View** offers a comprehensive overview of the loaded topology, providing information about the state of devices and their associated agents.



Validation: Not Started

Interval (sec): 30

Positive Report		Negative Report	
Name	Value	Name	Value
Loaded Devices	0	Devices With Wrong Agent Version	0
Installed Agents	0	Devices With No Agents	0
Reachable Devices	0	Unreachable Devices	0
		No json api	0
		No reports	0
		Late reports	0
		Stale reports	0
		Expected agent	1.7.0-5
		Comment	

The report includes the following key metrics:

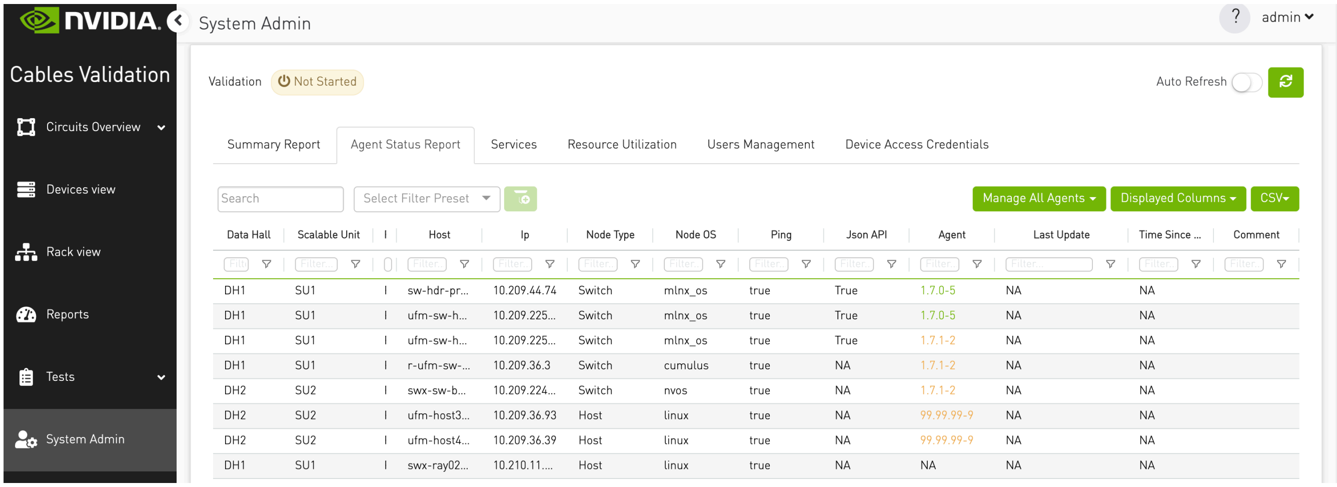
- **Loaded Devices** – The total number of devices loaded in the topology.
- **Installed Agents** – The number of devices with agents successfully installed.
- **Reachable Devices** – Devices that are accessible and responsive.
- **Unreachable Devices** – Devices that could not be reached during validation.
- **Devices with Wrong Agent Version** – Devices running an agent version different from the expected one.
- **Devices with No Agents** – Devices that do not have any agents installed.
- **No JSON API** – Devices that do not expose the required JSON API.
- **No Reports** – Devices that have not submitted any reports.
- **Late Reports** – Devices whose reports were received after the expected time.

- **Stale Reports** – Devices that have not sent updated reports within the last 15 minutes.

Agent Status Report

The **Agent Status Report** provides detailed information about the agents installed on each node within the topology.

The report includes key details such as the node's **location** (rack and unit), **type**, **name**, and **IP address**.



Data Hall	Scalable Unit	I	Host	Ip	Node Type	Node OS	Ping	Json API	Agent	Last Update	Time Since ...	Comment
DH1	SU1	I	sw-hdr-pr...	10.209.44.74	Switch	mlnx_os	true	True	1.7.0-5	NA	NA	
DH1	SU1	I	ufm-sw-h...	10.209.225...	Switch	mlnx_os	true	True	1.7.0-5	NA	NA	
DH1	SU1	I	ufm-sw-h...	10.209.225...	Switch	mlnx_os	true	True	1.7.1-2	NA	NA	
DH1	SU1	I	r-ufm-sw-...	10.209.36.3	Switch	cumulus	true	NA	1.7.1-2	NA	NA	
DH2	SU2	I	swx-sw-b...	10.209.224...	Switch	nvos	true	NA	1.7.1-2	NA	NA	
DH2	SU2	I	ufm-host3...	10.209.36.93	Host	linux	true	NA	99.99.99-9	NA	NA	
DH2	SU2	I	ufm-host4...	10.209.36.39	Host	linux	true	NA	99.99.99-9	NA	NA	
DH1	SU1	I	swx-ray02...	10.210.11....	Host	linux	true	NA	NA	NA	NA	

Note

To simplify validation, agent versions are now color-coded in the UI:

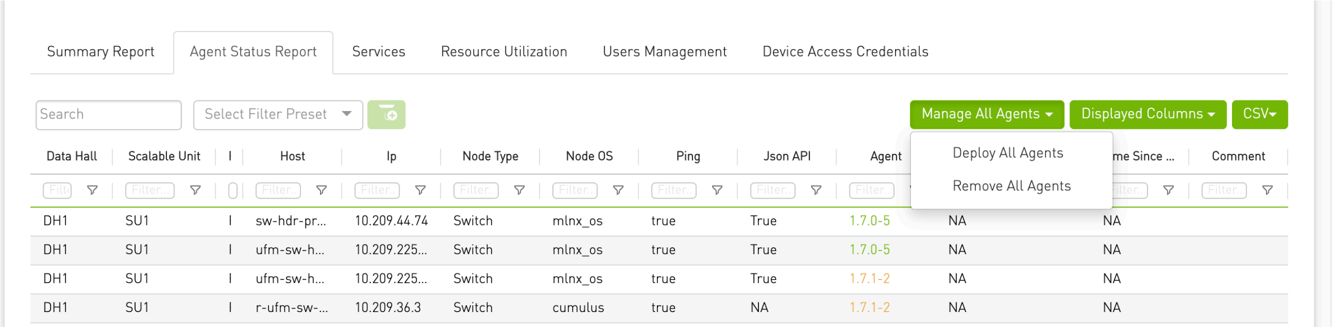
- **Green** → Version matches the bring-up version.
- **Orange** → Version does not match the bring-up version.
- **Black** → Not available (N/A).

Deployment Capabilities

The **Agent Status Report** also supports agent management operations, allowing administrators to deploy or remove agents directly from the report. Two operation modes are supported:

Bulk operations

- *Deploy All Agents*: Installs agents on all devices in the topology.
- *Remove All Agents*: Removes agents from all devices.



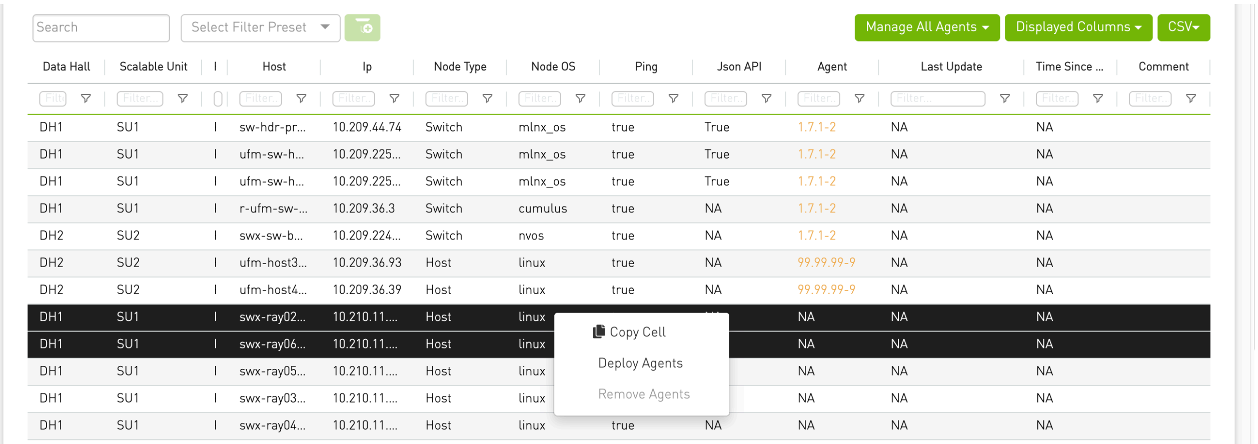
Data Hall	Scalable Unit	I	Host	Ip	Node Type	Node OS	Ping	Json API	Agent		me Since ...	Comment
DH1	SU1	I	sw-hdr-pr...	10.209.44.74	Switch	mlnx_os	true	True	1.7.0-5	NA	NA	
DH1	SU1	I	ufm-sw-h...	10.209.225...	Switch	mlnx_os	true	True	1.7.0-5	NA	NA	
DH1	SU1	I	ufm-sw-h...	10.209.225...	Switch	mlnx_os	true	True	1.7.1-2	NA	NA	
DH1	SU1	I	r-ufm-sw-...	10.209.36.3	Switch	cumulus	true	NA	1.7.1-2	NA	NA	

Individual or Multiple Selection

- Users can right-click on selected devices to access deployment or removal options.
- Multiple devices can be selected for deployment or removal in a single action.

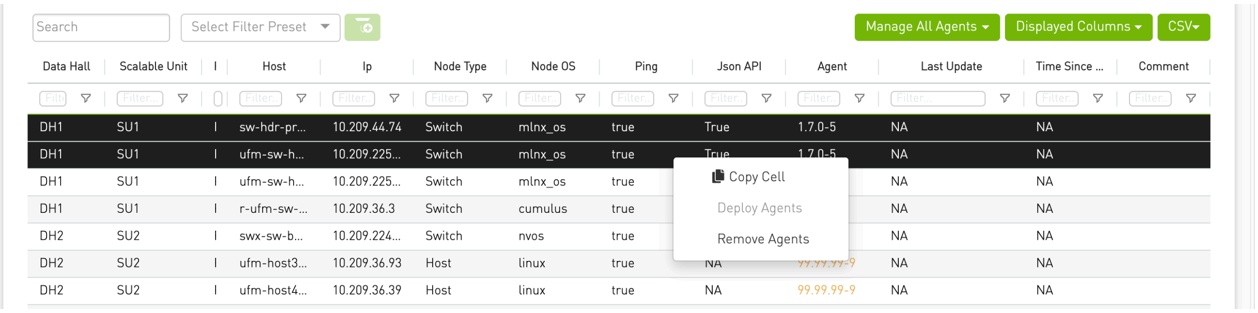
Behavior Based on Device Selection

- **Agent not installed on any selected devices** → *Remove Agents* option is disabled.



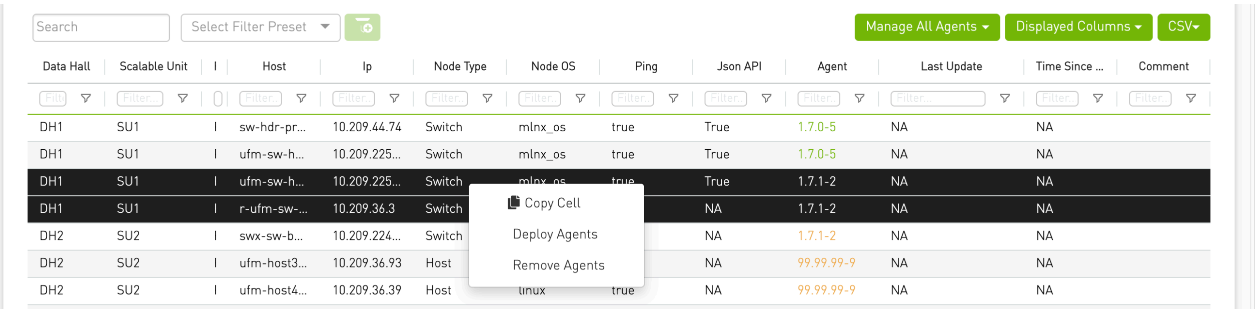
Data Hall	Scalable Unit	I	Host	Ip	Node Type	Node OS	Ping	Json API	Agent	Last Update	Time Since ...	Comment
DH1	SU1	I	sw-hdr-pr...	10.209.44.74	Switch	mlnx_os	true	True	1.7.1-2	NA	NA	
DH1	SU1	I	ufm-sw-h...	10.209.225...	Switch	mlnx_os	true	True	1.7.1-2	NA	NA	
DH1	SU1	I	ufm-sw-h...	10.209.225...	Switch	mlnx_os	true	True	1.7.1-2	NA	NA	
DH1	SU1	I	r-ufm-sw-...	10.209.36.3	Switch	cumulus	true	NA	1.7.1-2	NA	NA	
DH2	SU2	I	swx-sw-b...	10.209.224...	Switch	nvos	true	NA	1.7.1-2	NA	NA	
DH2	SU2	I	ufm-host3...	10.209.36.93	Host	linux	true	NA	99.99.99-9	NA	NA	
DH2	SU2	I	ufm-host4...	10.209.36.39	Host	linux	true	NA	99.99.99-9	NA	NA	
DH1	SU1	I	swx-ray02...	10.210.11....	Host	linux			NA	NA	NA	
DH1	SU1	I	swx-ray06...	10.210.11....	Host	linux			NA	NA	NA	
DH1	SU1	I	swx-ray05...	10.210.11....	Host	linux			NA	NA	NA	
DH1	SU1	I	swx-ray03...	10.210.11....	Host	linux			NA	NA	NA	
DH1	SU1	I	swx-ray04...	10.210.11....	Host	linux	true	NA	NA	NA	NA	

- **Correct agent (matching the bring-up version) installed on all selected devices** → *Deploy Agents* option is disabled.



Data Hall	Scalable Unit	I	Host	Ip	Node Type	Node OS	Ping	Json API	Agent	Last Update	Time Since ...	Comment
DH1	SU1	I	sw-hdr-pr...	10.209.44.74	Switch	mlnx_os	true	True	1.7.0-5	NA	NA	
DH1	SU1	I	ufm-sw-h...	10.209.225...	Switch	mlnx_os	true	True	1.7.0-5	NA	NA	
DH1	SU1	I	ufm-sw-h...	10.209.225...	Switch	mlnx_os	true			NA	NA	
DH1	SU1	I	r-ufm-sw-...	10.209.36.3	Switch	cumulus	true			NA	NA	
DH2	SU2	I	swx-sw-b...	10.209.224...	Switch	nvos	true			NA	NA	
DH2	SU2	I	ufm-host3...	10.209.36.93	Host	linux	true	NA	99.99.99-9	NA	NA	
DH2	SU2	I	ufm-host4...	10.209.36.39	Host	linux	true	NA	99.99.99-9	NA	NA	

- **Mixed selection (some devices with, some without agents)** → Both *Deploy Agents* and *Remove Agents* options are enabled, and the system will handle the case appropriately.



Data Hall	Scalable Unit	I	Host	Ip	Node Type	Node OS	Ping	Json API	Agent	Last Update	Time Since ...	Comment
DH1	SU1	I	sw-hdr-pr...	10.209.44.74	Switch	mlnx_os	true	True	1.7.0-5	NA	NA	
DH1	SU1	I	ufm-sw-h...	10.209.225...	Switch	mlnx_os	true	True	1.7.0-5	NA	NA	
DH1	SU1	I	ufm-sw-h...	10.209.225...	Switch	mlnx_os	true	True	1.7.1-2	NA	NA	
DH1	SU1	I	r-ufm-sw-...	10.209.36.3	Switch			NA	1.7.1-2	NA	NA	
DH2	SU2	I	swx-sw-b...	10.209.224...	Switch			NA	1.7.1-2	NA	NA	
DH2	SU2	I	ufm-host3...	10.209.36.93	Host			NA	99.99.99-9	NA	NA	
DH2	SU2	I	ufm-host4...	10.209.36.39	Host	linux	true	NA	99.99.99-9	NA	NA	

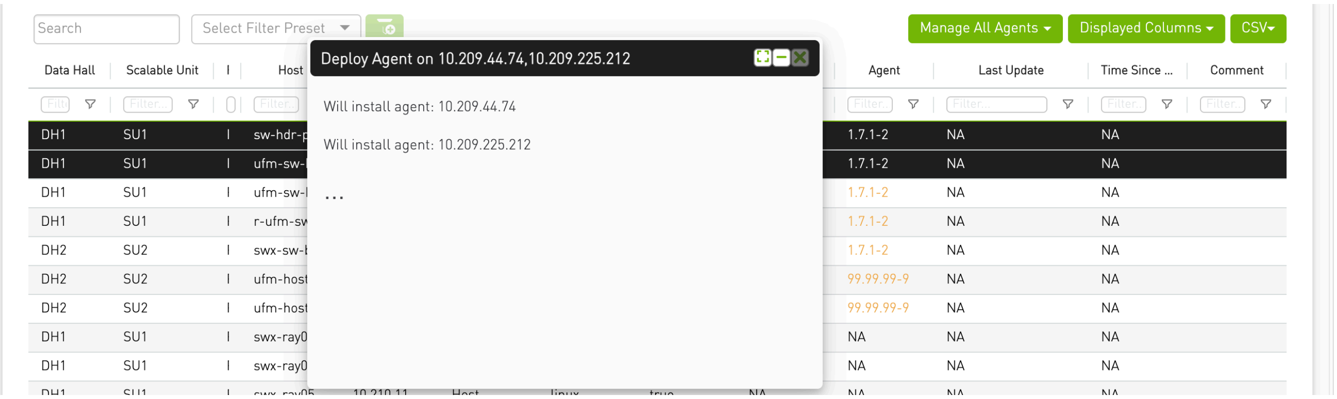
Real-time deployment monitoring

After deploying or removing agents, a **monitoring window** is displayed.

This window shows real-time output of the ongoing operation, allowing administrators to track deployment progress and verify success or failure.

The window is **resizable**, enabling users to minimize it and continue navigating through the application while the operation runs in the background.

The **Close** button remains disabled during the operation to prevent accidental closure until the process is complete.



Deploy Agent on 10.209.44.74,10.209.225.212

Will install agent: 10.209.44.74

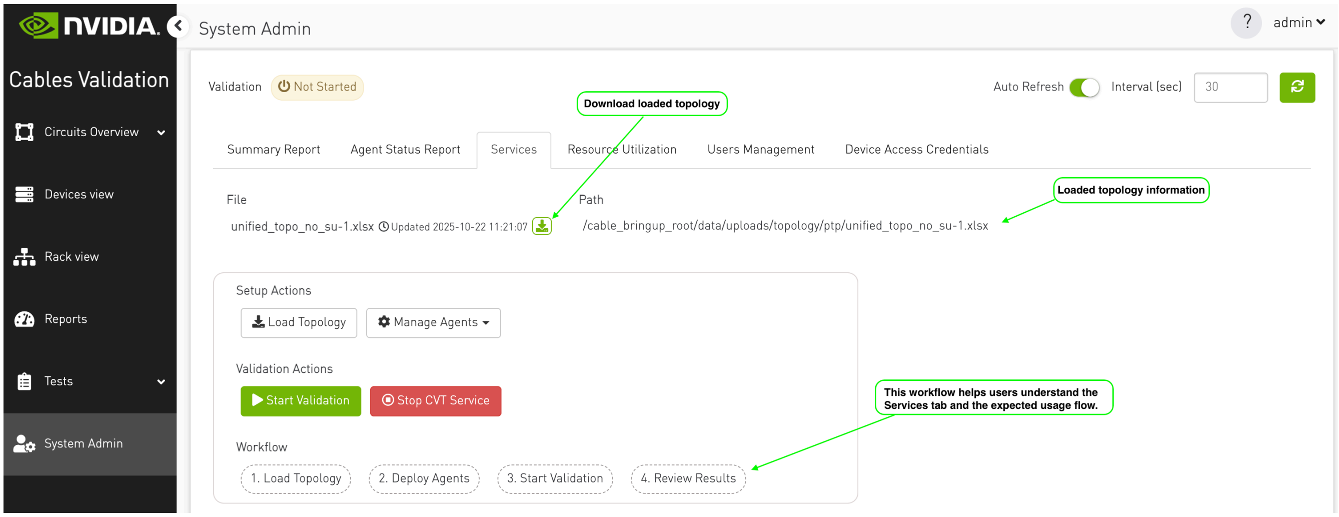
Will install agent: 10.209.225.212

...

Data Hall	Scalable Unit	I	Host	Agent	Last Update	Time Since ...	Comment
DH1	SU1	I	sw-hdr-p	1.7.1-2	NA	NA	
DH1	SU1	I	ufm-sw-	1.7.1-2	NA	NA	
DH1	SU1	I	ufm-sw-l	1.7.1-2	NA	NA	
DH1	SU1	I	r-ufm-sw	1.7.1-2	NA	NA	
DH2	SU2	I	swx-sw-l	1.7.1-2	NA	NA	
DH2	SU2	I	ufm-host	99.99.99-9	NA	NA	
DH2	SU2	I	ufm-host	99.99.99-9	NA	NA	
DH1	SU1	I	swx-ray0	NA	NA	NA	
DH1	SU1	I	swx-ray0	NA	NA	NA	

Services

The **Service View** enables users to manage the topology by allowing them to Load Topology, Manage Agents, Start/Stop Validation, and Stop CVT Service



Validation: Not Started

Auto Refresh: ●

Interval (sec): 30

File: unified_topo_no_su-1.xlsx Updated 2025-10-22 11:21:07

Path: /cable_bringup_root/data/uploads/topology/ptp/unified_topo_no_su-1.xlsx

Download loaded topology

Loaded topology information

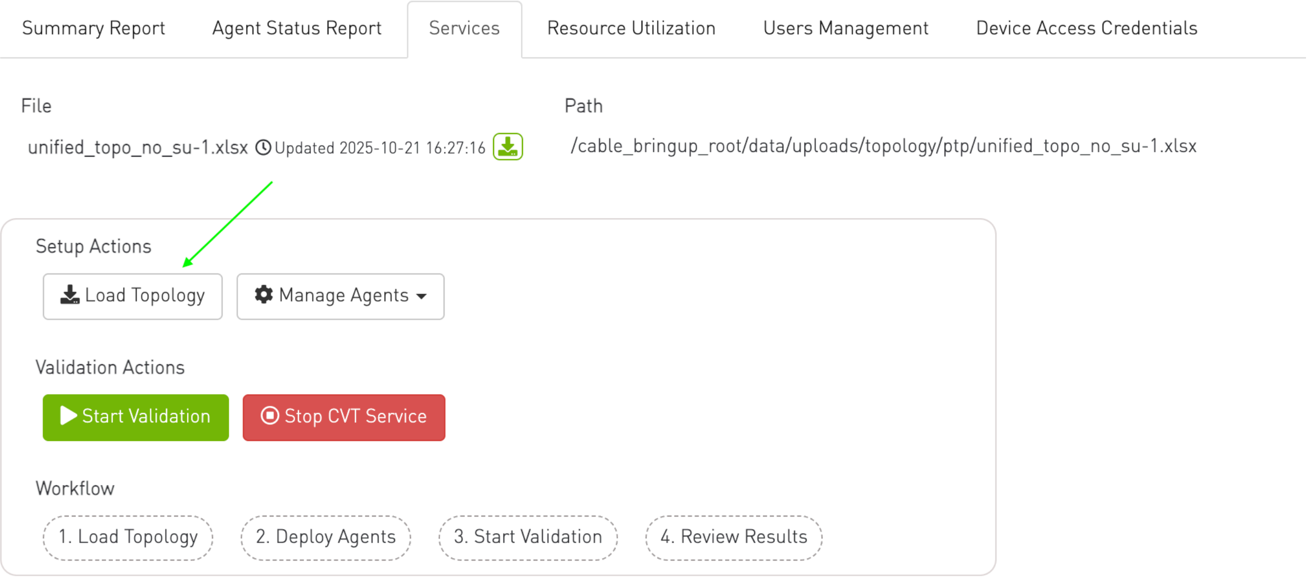
This workflow helps users understand the Services tab and the expected usage flow.

Load Topology

Based on the fabric type, the **Load Topology** allows users to upload and load a PTP, DOT, or topology file.

Users can access the **Load Topology Wizard** by clicking the **Load Topology** button.

The button is **enabled only when the validation is stopped**.



File: unified_topo_no_su-1.xlsx Updated 2025-10-21 16:27:16

Path: /cable_bringup_root/data/uploads/topology/ptp/unified_topo_no_su-1.xlsx

Wizard Steps

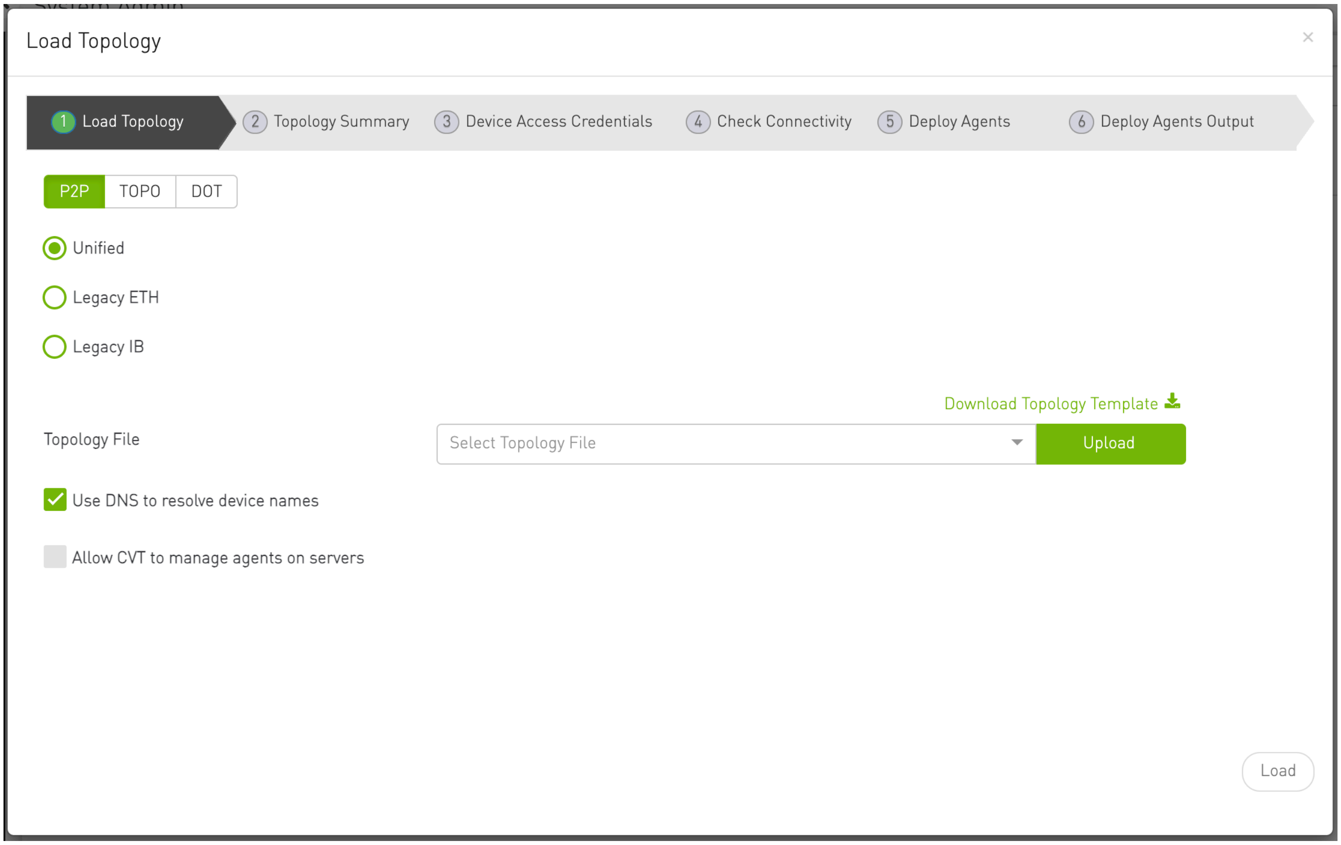
The wizard contains 6 steps

Step 1 - Load Topology

In this step, users can select one of the following three options to load the desired topology file:

P2P File

This option uses the load_ptp command to load a P2P file.



● Unified

○ Legacy ETH

○ Legacy IB

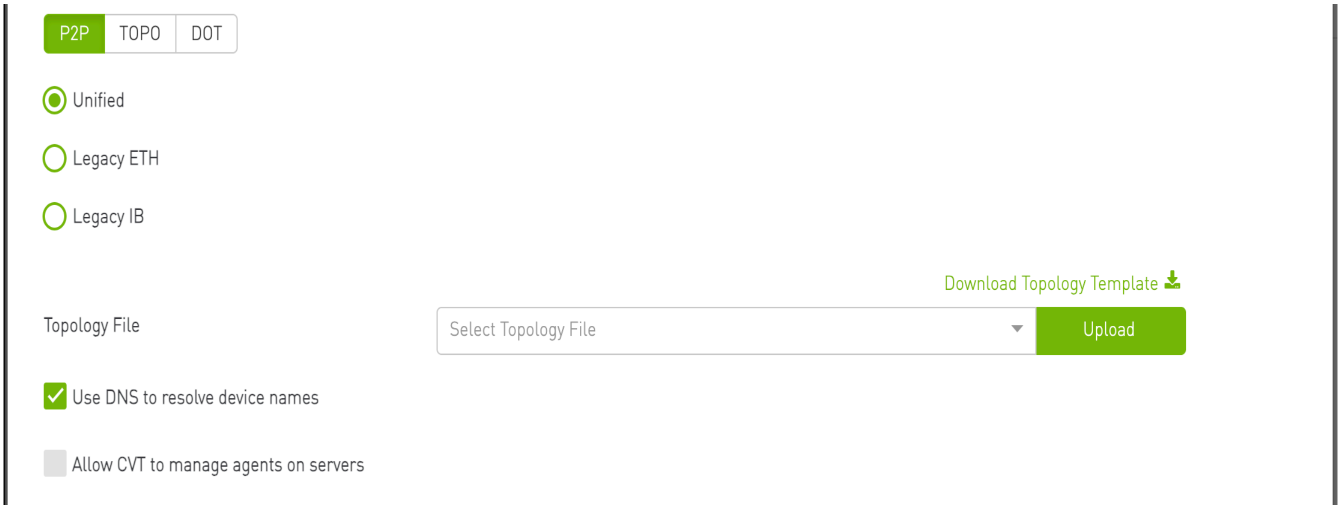
Topology File

☑ Use DNS to resolve device names

☐ Allow CVT to manage agents on servers

Users must provide the following inputs:

- **Format**: PTP file format, which can be one of the following:
  - Unified: I t accepts various cluster types, including InfiniBand , Ethernet, and NVLink. For more information, please visit Unified Topology Format.



● Unified

○ Legacy ETH

○ Legacy IB

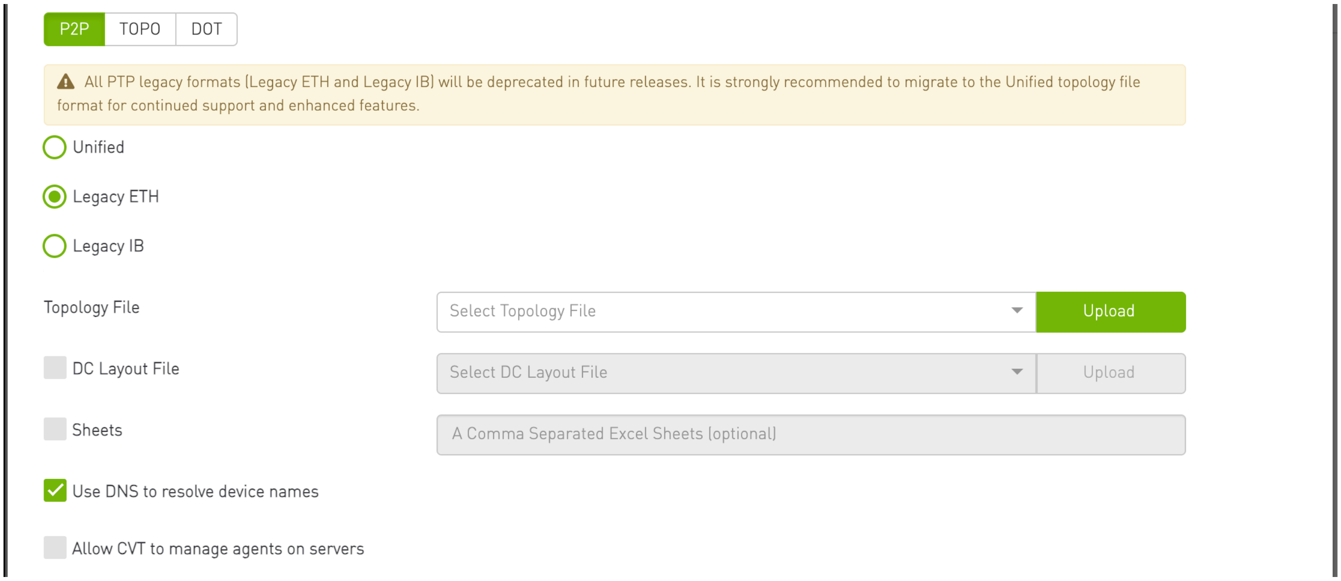
Topology File

☑ Use DNS to resolve device names

☐ Allow CVT to manage agents on servers

-

    - Legace ETH: It accepts only Ethernet fabric. For more information, please visit PTP ETH.



⚠ All PTP legacy formats (Legacy ETH and Legacy IB) will be deprecated in future releases. It is strongly recommended to migrate to the Unified topology file format for continued support and enhanced features.

○ Unified

● Legacy ETH

○ Legacy IB

Topology File — Select Topology File — Upload

☐ DC Layout File — Select DC Layout File — Upload

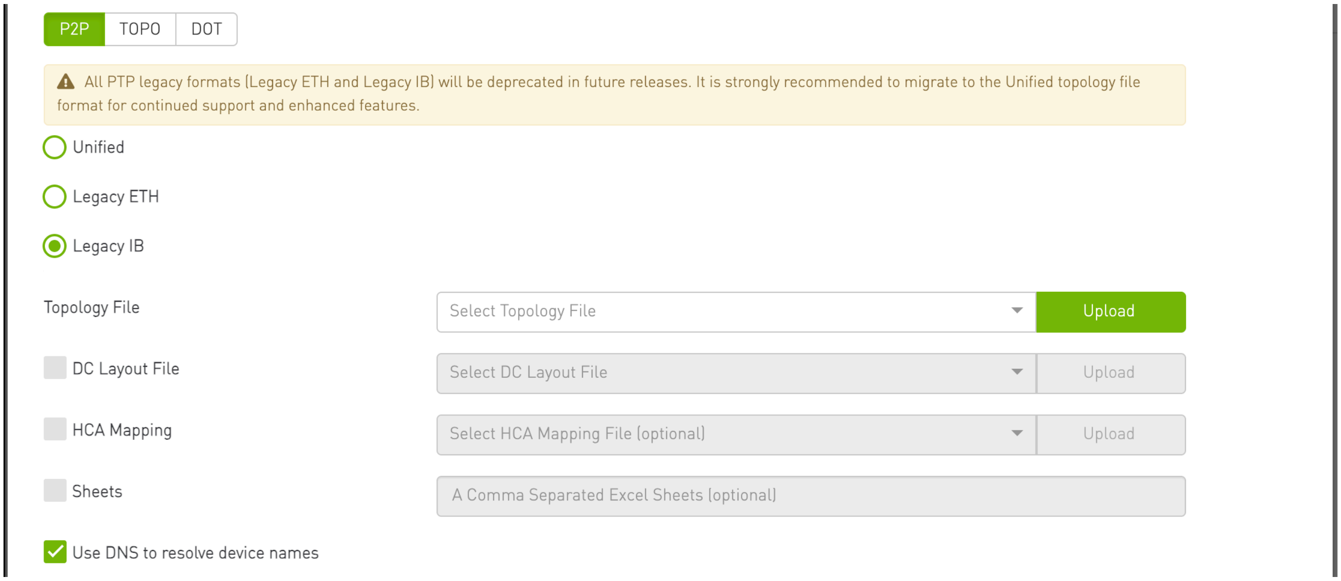
☐ Sheets — A Comma Separated Excel Sheets (optional)

☑ Use DNS to resolve device names

☐ Allow CVT to manage agents on servers

-

    - Legacy IB: It accepts only InfiniBand fabric. For more information, please visit PTP IB.



⚠ All PTP legacy formats (Legacy ETH and Legacy IB) will be deprecated in future releases. It is strongly recommended to migrate to the Unified topology file format for continued support and enhanced features.

○ Unified

○ Legacy ETH

● Legacy IB

Topology File — Select Topology File — Upload

☐ DC Layout File — Select DC Layout File — Upload

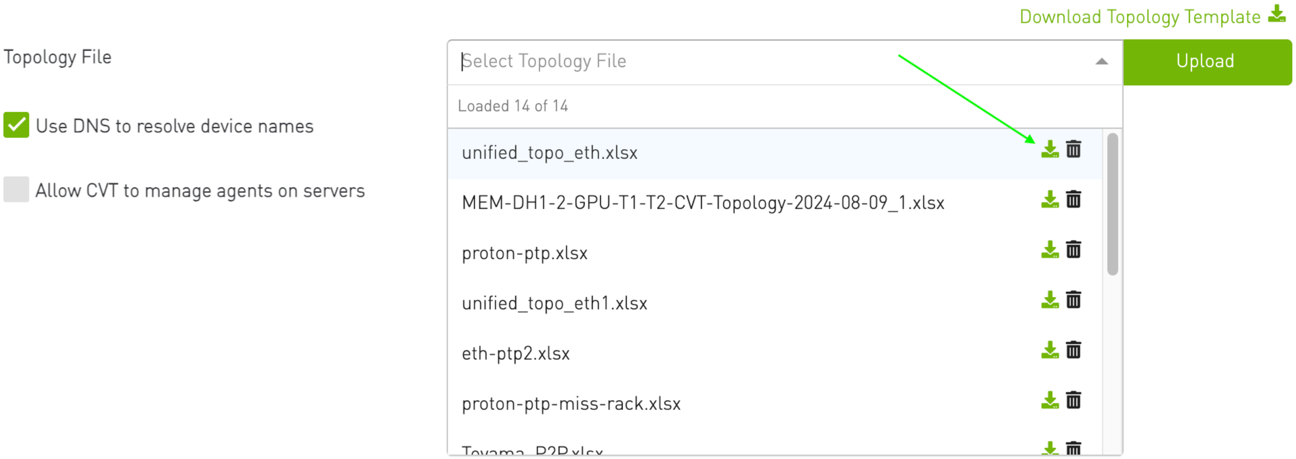
☐ HCA Mapping — Select HCA Mapping File (optional) — Upload

☐ Sheets — A Comma Separated Excel Sheets (optional)

☑ Use DNS to resolve device names

- **Topology File**:
    - Allows users to select or upload a PTP file.
    - Files can be chosen from a dropdown menu listing all .xlsx files in the directory: /opt/bringup_data/data/uploads/topology/ptp/

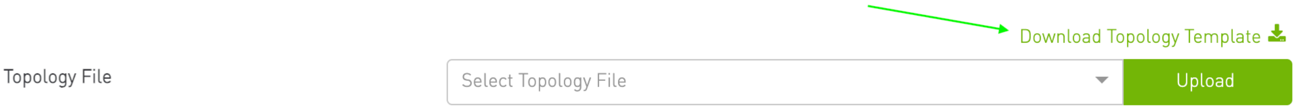
  - This directory is a shared volume linked to:/cable_bringup_root/data/uploads/topology/ptp/
  - Users can also upload a file directly from their computer using the **Upload** button located next to the dropdown menu.
  - Additionally, users can **download** any file from the dropdown to their local computer.



☑ Use DNS to resolve device names

☐ Allow CVT to manage agents on servers

-
  - For the **Unified** type, users can download a **template file** that serves as a guide for creating their own unified P2P file.



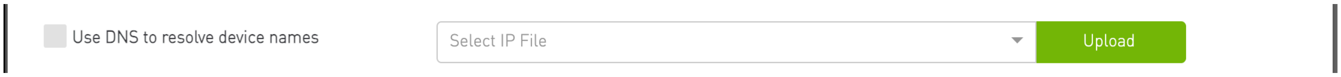
- **DC Layout File (Optional, Appears Only with Legacy formats)**:
  - Used for selecting or uploading a data center floor layout file.
  - Files can be chosen from a dropdown menu listing all .csv files in the directory: /opt/bringup_data/data/uploads/topology/dc_layout/
  - This directory is a shared volume linked to: /cable_bringup_root/data/uploads/topology/dc_layout/
  - Users can upload a file directly from their computer using the Upload button.
  - Users can enable the DC Layout field by selecting the HCA Mapping checkbox.

- **HCA Mapping (Optional, Appears Only with Legacy formats):**
    - Allows users to select or upload an HCA mapping file.
    - Files can be chosen from a dropdown menu listing all .csv files in the directory:/opt/bringup_data/data/uploads/topology/hca_mapping/
    - This directory is a shared volume linked to:

      ```
      /cable_bringup_root/data/uploads/topology/hca_mapping/
      ```

    - Users can enable the HCA Mapping field by selecting the HCA Mapping checkbox.
- **Sheets (Optional, Appears Only with Legacy formats):** The user can specify sheets to be included in the file by enabling the **Sheets** input field (via checkbox) and entering the sheet names as comma-separated values.
- **Use DNS to resolve device names**: This option is selected by default. When deselected, an additional input field appears where users can upload an IP file.

  In this mode, the **Load Topology** process runs two commands sequentially — **Load Topology** and **Load IP** — to load both topology and IP data correctly.

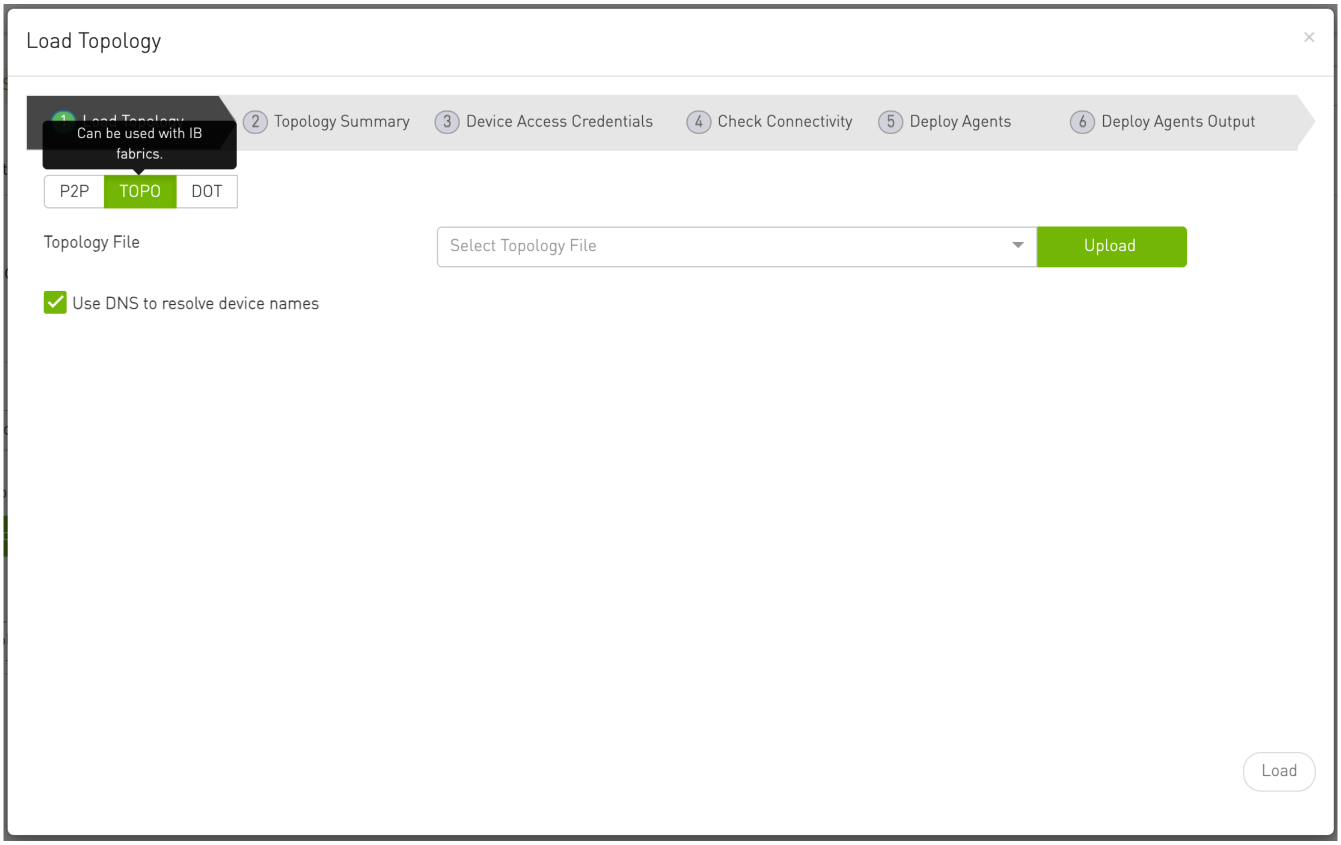


☐ Use DNS to resolve device names

- **Allow CVT to manage agents on servers (Appears Only with Unified and Legacy ETH formats):** This option is deselected by default. If selected, CVT manages the servers included in the topology file; otherwise, it manages only the switches and ignores the servers.

Topo File

This option uses the `load_topo` command to load a topology file.



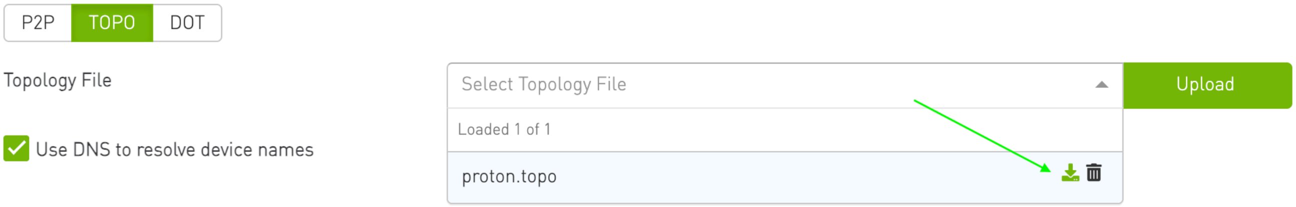
Can be used with IB fabrics.

Topology File: Select Topology File

☑ Use DNS to resolve device names

User must provide the following input:

- **Topology File:**
  - Allows users to select or upload a .topo file.
  - Files can be chosen from a dropdown menu listing all .topo files in the directory:/opt/bringup_data/data/uploads/topology/topo/
  - This directory is a shared volume linked to:/cable_bringup_root/data/uploads/topology/topo/
  - Users can upload a file directly from their computer using the **Upload** button.
  - Additionally, users can **download** any file from the dropdown to their local computer.



Topology File: Select Topology File

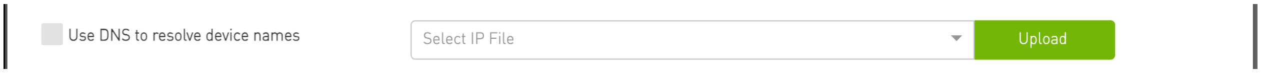
☑ Use DNS to resolve device names

Loaded 1 of 1

proton.topo

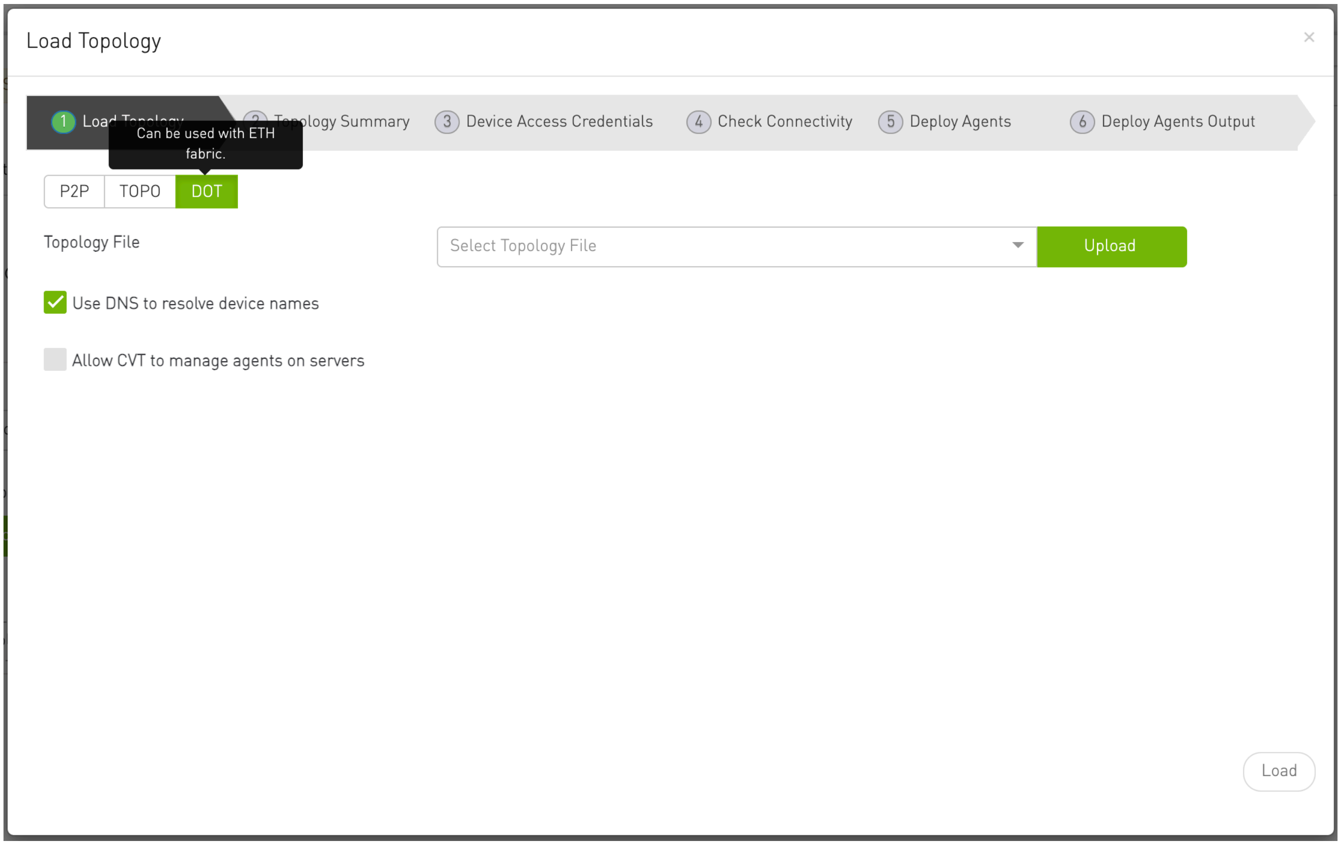
- **Use DNS to resolve device names**: This option is selected by default. When deselected, an additional input field appears where users can upload an IP file.

  In this mode, the **Load Topology** process runs two commands sequentially — **Load Topology** and **Load IP** — to load both topology and IP data correctly.



Dot File

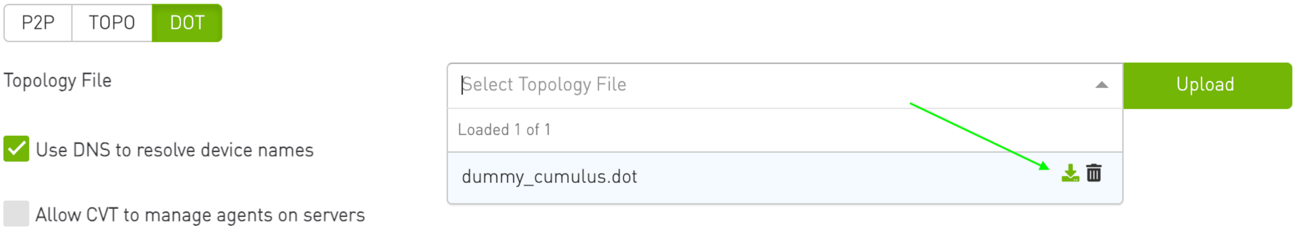
This option uses the `load_topo` command to load a DOT file.



This option uses the `load_topo` command to load a DOT file. User must provide the following input

- **Topology File:**
  - Allows users to select or upload a .dot file.

  - Files can be chosen from a dropdown menu listing all .topo files in the directory: /opt/bringup_data/data/uploads/topology/dot/
  - This directory is a shared volume linked to :/cable_bringup_root/data/uploads/topology/dot/
  - Users can upload a file directly from their computer using the **Upload** button.
  - Additionally, users can **download** any file from the dropdown to their local computer.

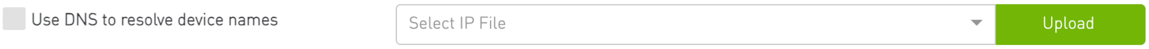


☑ Use DNS to resolve device names

☐ Allow CVT to manage agents on servers

- **Use DNS to resolve device names**: This option is selected by default. When deselected, an additional input field appears where users can upload an IP file.

  In this mode, the **Load Topology** process runs two commands sequentially — **Load Topology** and **Load IP** — to load both topology and IP data correctly.

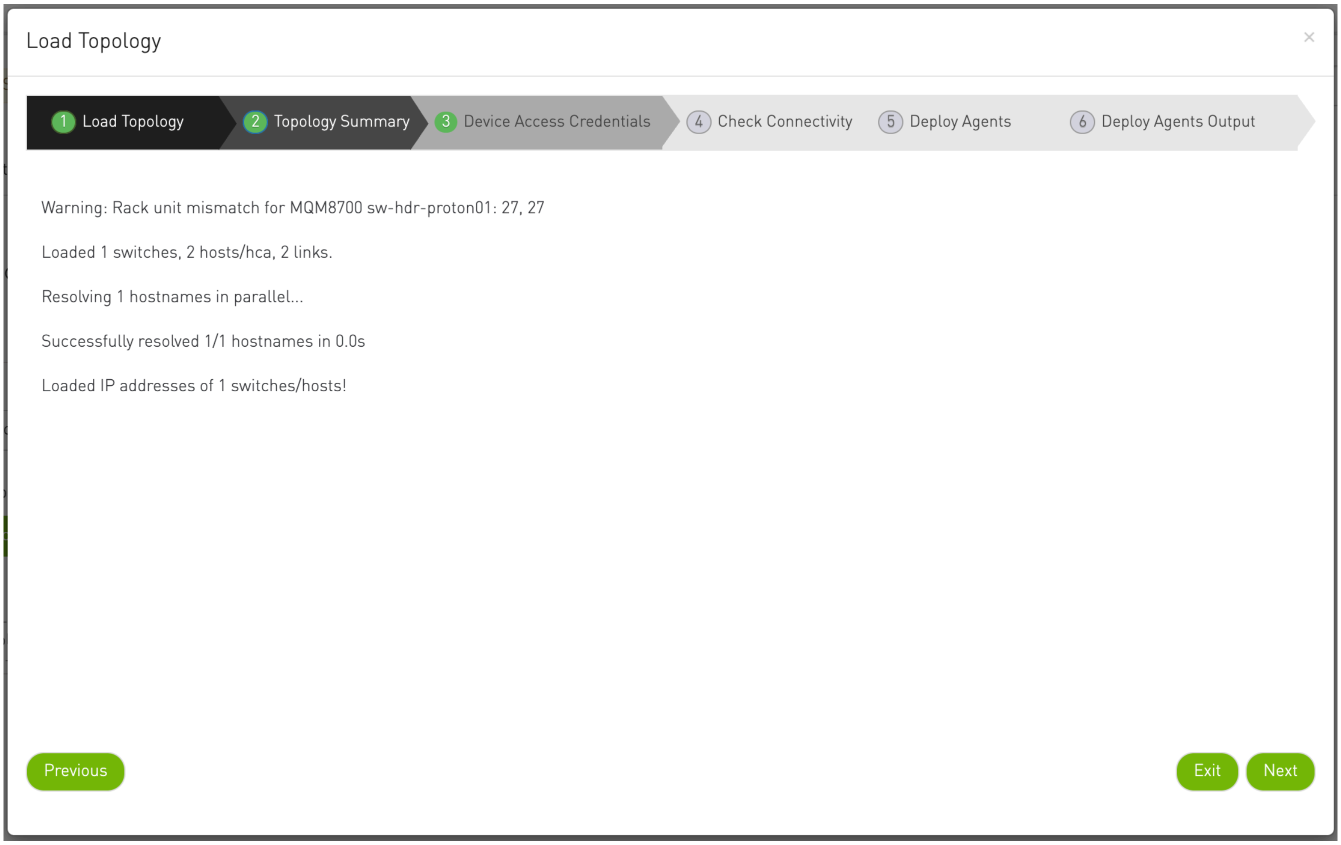


☐ Use DNS to resolve device names

- **Allow CVT to manage agents on servers:** This option is deselected by default. If selected, CVT manages the servers included in the topology file; otherwise, it manages only the switches and ignores the servers.

Step 2 - Topology Summary:

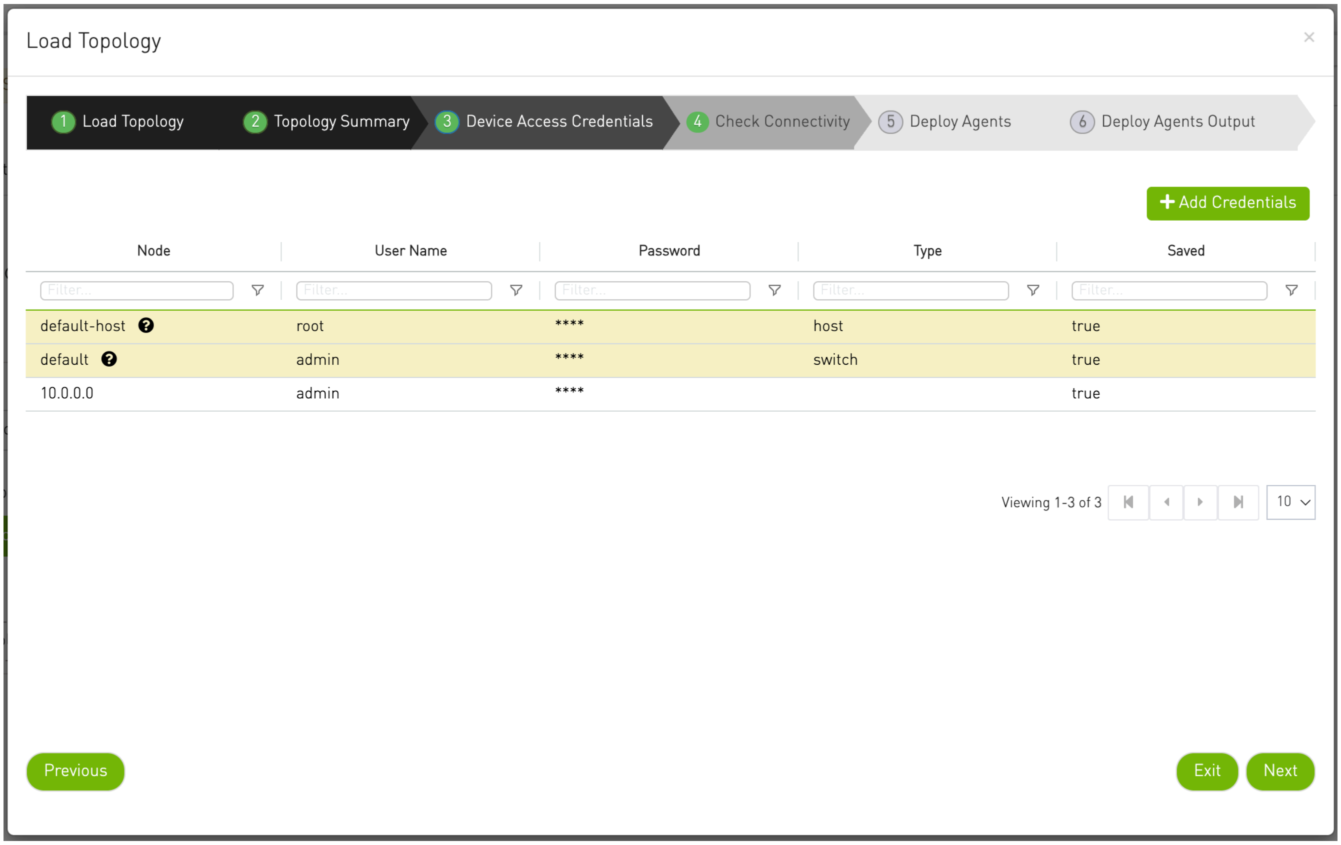
After loading the file, this step provides a summary of the loaded topology. Users can review the details to confirm that the correct file and settings have been applied.



```
Warning: Rack unit mismatch for MQM8700 sw-hdr-proton01: 27, 27

Loaded 1 switches, 2 hosts/hca, 2 links.

Resolving 1 hostnames in parallel...

Successfully resolved 1/1 hostnames in 0.0s

Loaded IP addresses of 1 switches/hosts!
```

Step 3 - Device Access Credentials:

For more information, please visit Device Access Credentials

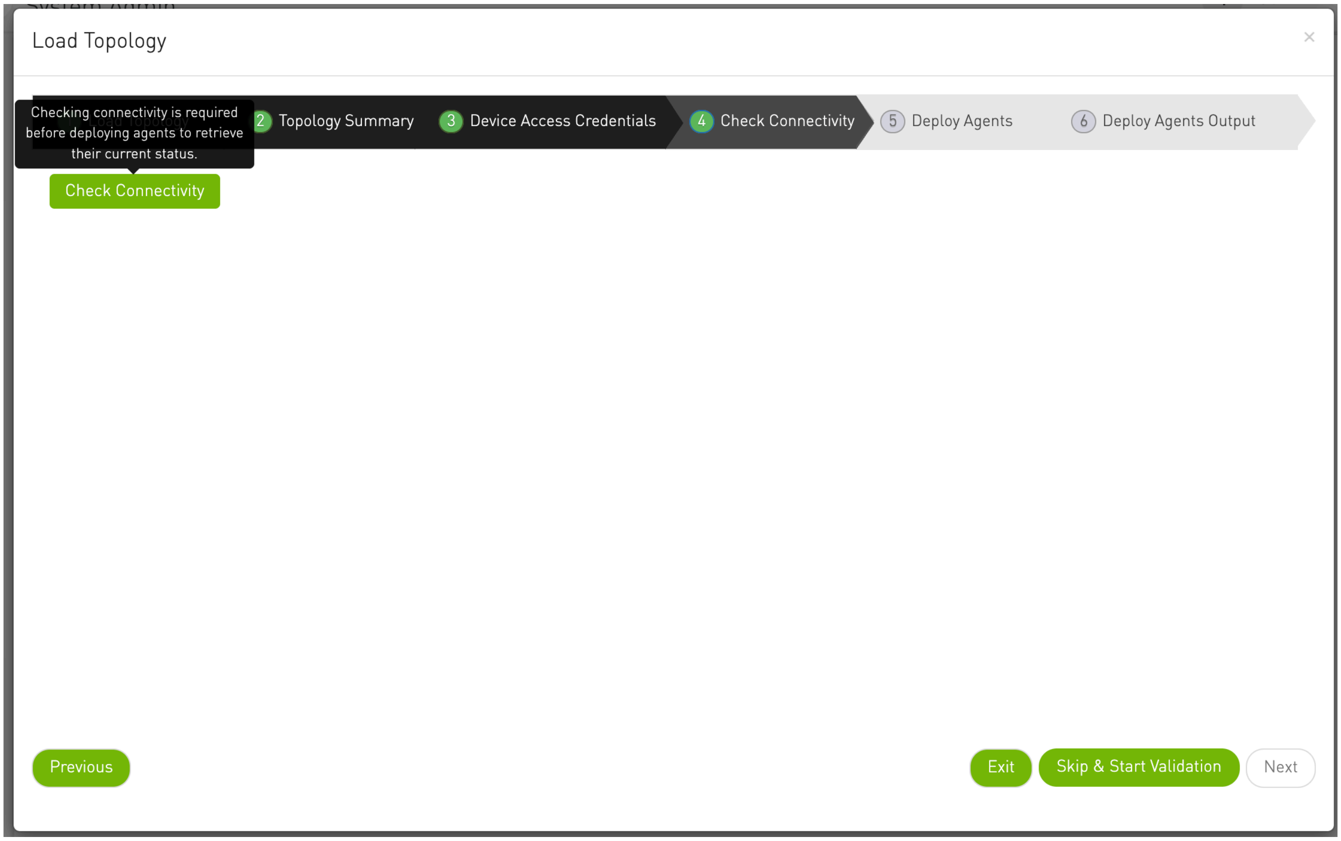


Node	User Name	Password	Type	Saved
default-host	root	****	host	true
default	admin	****	switch	true
10.0.0.0	admin	****		true

Step 4 - Check Connectivity:

After setting the credentials, the **Check Connectivity** tab verifies the connectivity of the currently loaded devices. This helps users confirm that the provided credentials are correct and view the status of the switches — for example, whether agents are present.

The tab displays a **Check Connectivity** button with a tooltip explaining its purpose. The **Next** button remains disabled until the connectivity check has been completed.

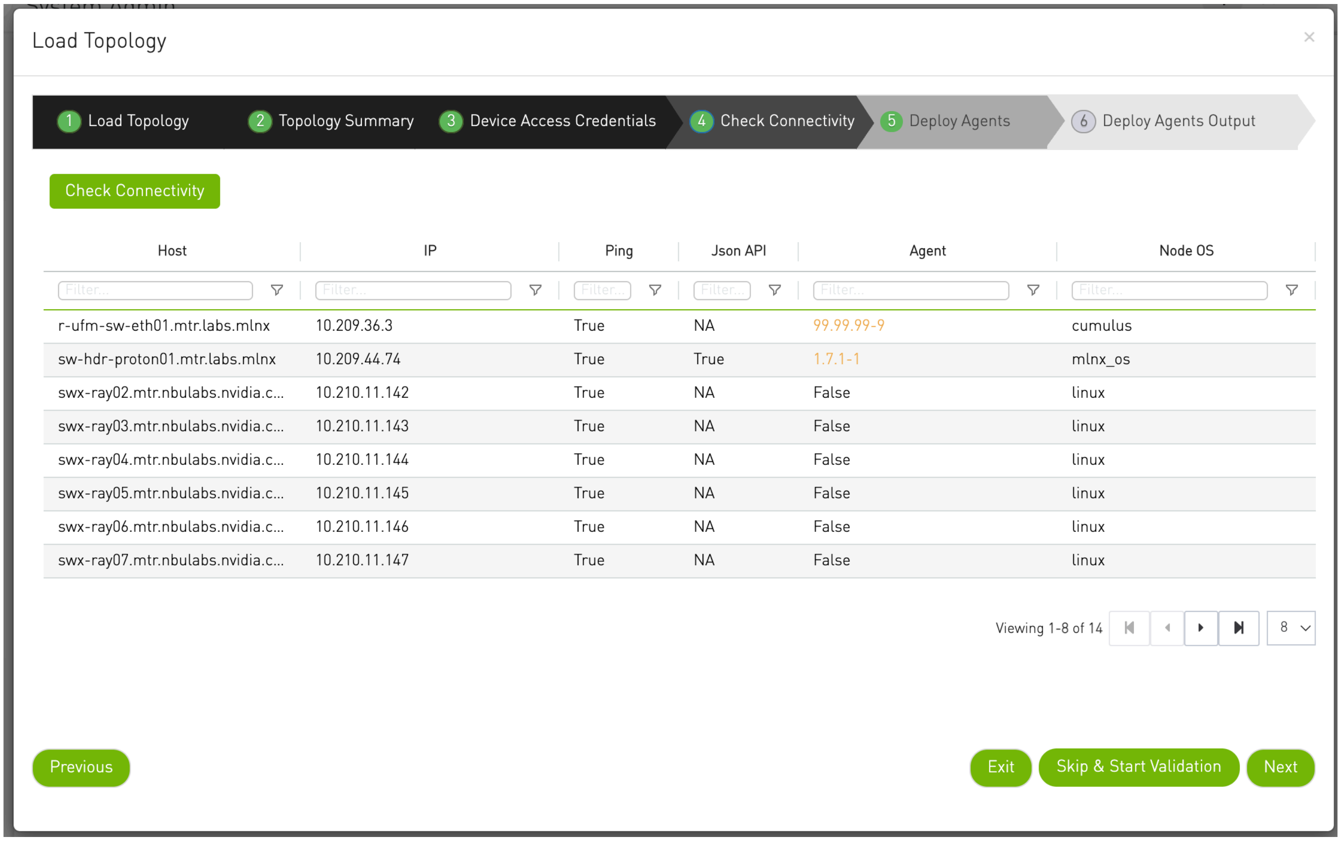


Users can either:

- Click **Check Connectivity** to run the check.
- Click **Skip & Start Validation** to skip the check and proceed to validation.

While the connectivity check is running, both **Next** and **Skip & Start Validation** are disabled.

Once complete, the results appear in a **table.**



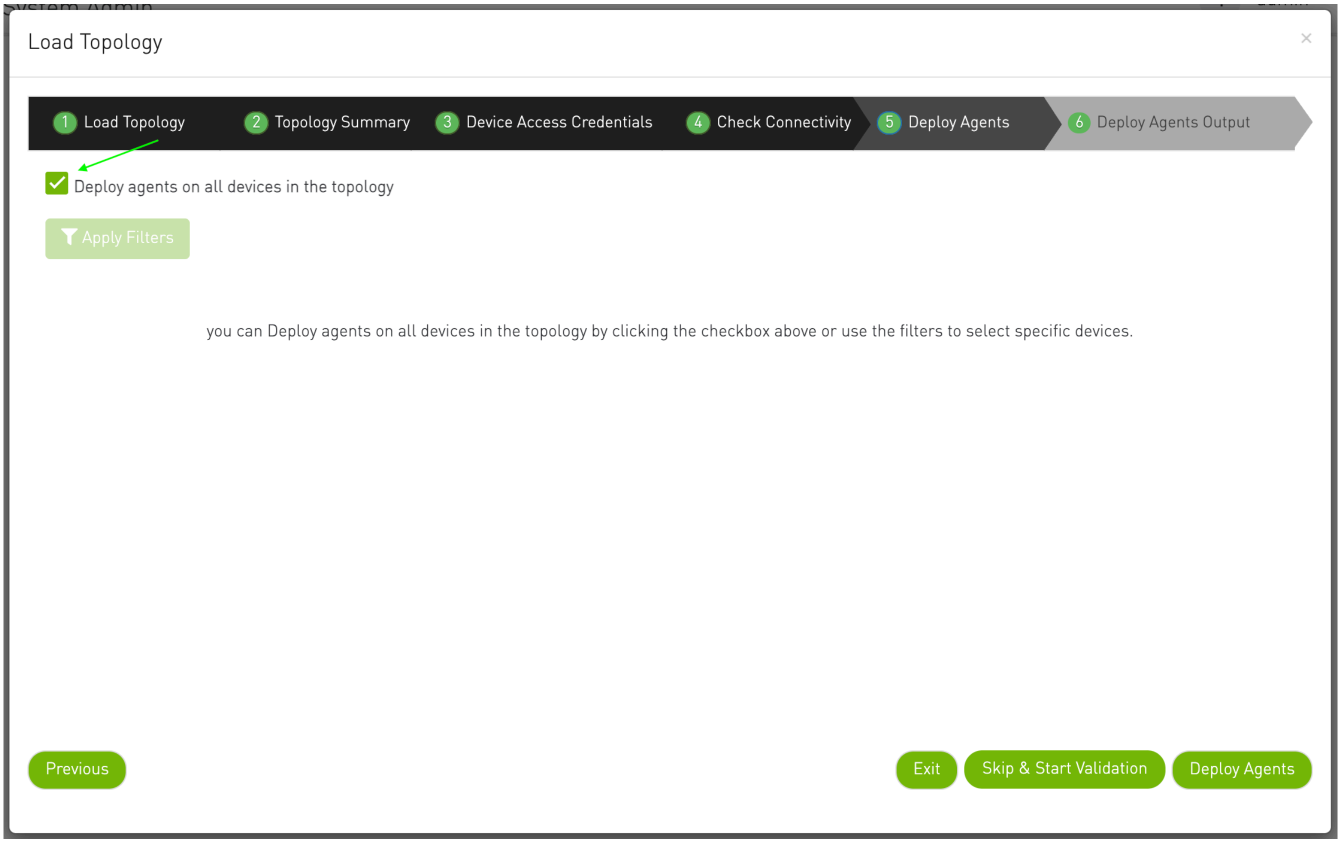
Host	IP	Ping	Json API	Agent	Node OS
r-ufm-sw-eth01.mtr.labs.mlnx	10.209.36.3	True	NA	99.99.99-9	cumulus
sw-hdr-proton01.mtr.labs.mlnx	10.209.44.74	True	True	1.7.1-1	mlnx_os
swx-ray02.mtr.nbulabs.nvidia.c...	10.210.11.142	True	NA	False	linux
swx-ray03.mtr.nbulabs.nvidia.c...	10.210.11.143	True	NA	False	linux
swx-ray04.mtr.nbulabs.nvidia.c...	10.210.11.144	True	NA	False	linux
swx-ray05.mtr.nbulabs.nvidia.c...	10.210.11.145	True	NA	False	linux
swx-ray06.mtr.nbulabs.nvidia.c...	10.210.11.146	True	NA	False	linux
swx-ray07.mtr.nbulabs.nvidia.c...	10.210.11.147	True	NA	False	linux

Step 5 - Deploy Agents:

After verifying connectivity, the **Deploy Agents** tab allows users to deploy agents to devices within the topology. This step ensures that all required agents are installed and ready before running validation.

The tab provides users with two deployment options:

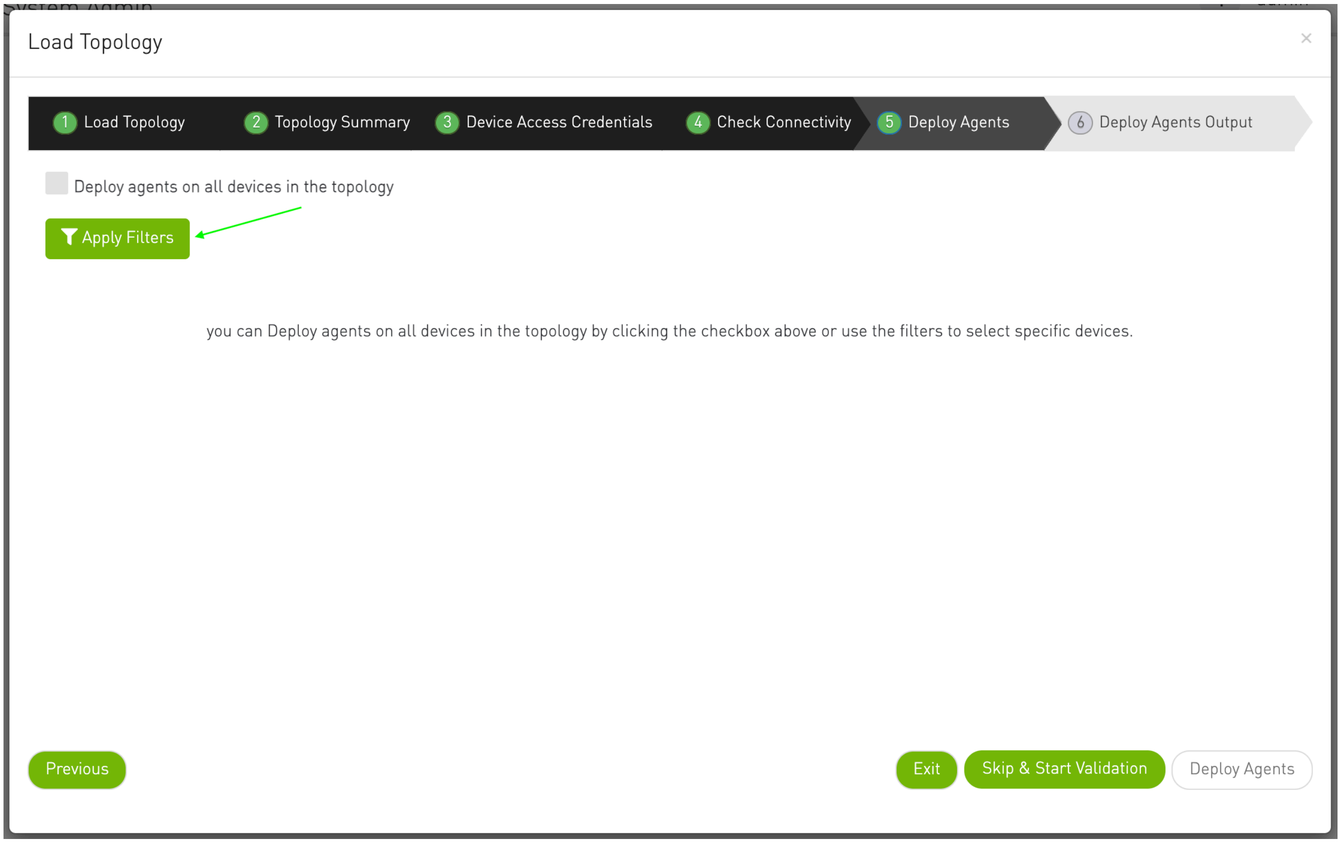
- **Deploy to All Devices** – Deploy agents to all devices in the loaded topology.



☑ Deploy agents on all devices in the topology

you can Deploy agents on all devices in the topology by clicking the checkbox above or use the filters to select specific devices.

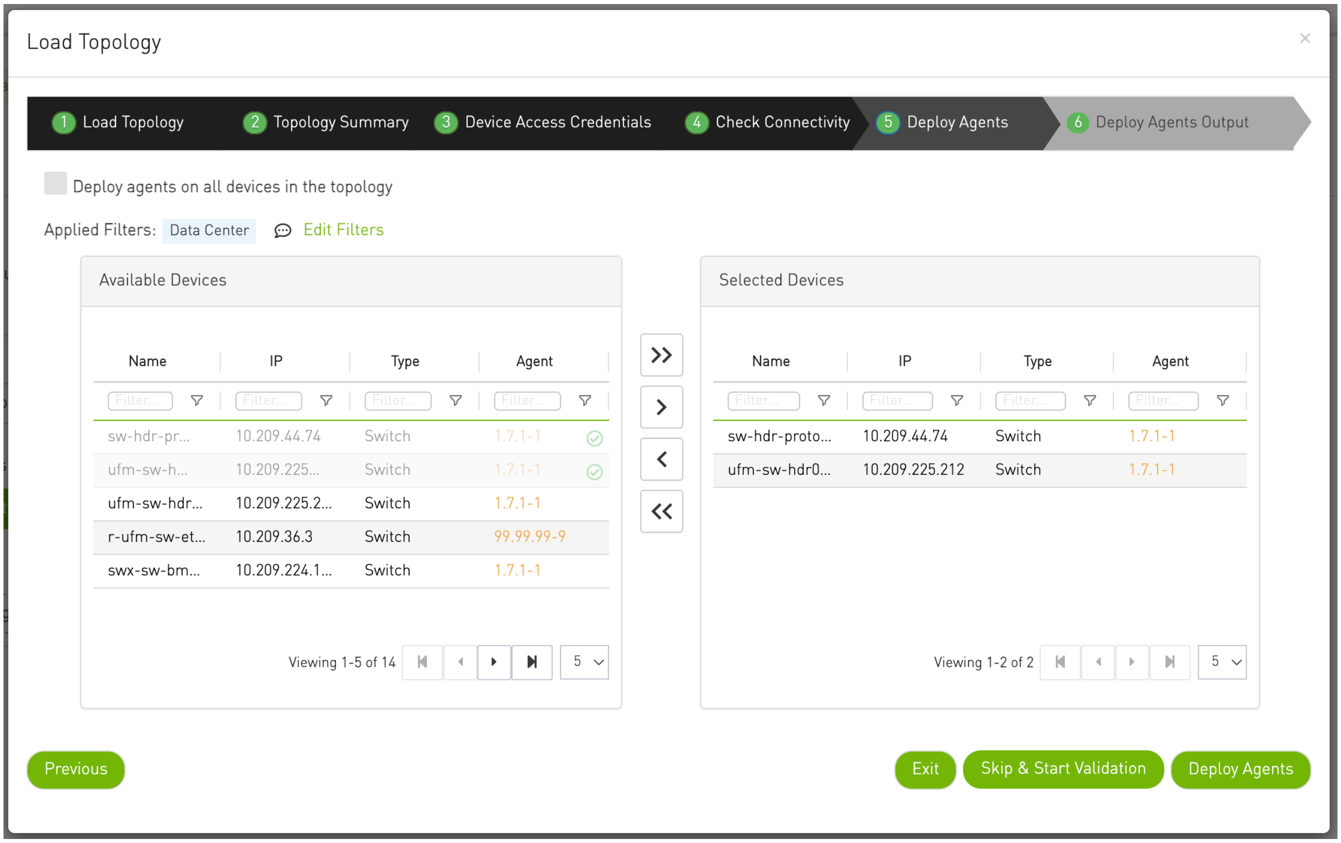
- **Deploy to Selected Devices** – Deploy agents only to specific devices using the filtering functionality.



☐ Deploy agents on all devices in the topology

you can Deploy agents on all devices in the topology by clicking the checkbox above or use the filters to select specific devices.

When using filters, all matching devices appear in a **dual-list table**, where users can move devices from the left list to the right to select them for deployment. Once the desired devices are selected, users can start the process by clicking **Deploy Agents**.



☐ Deploy agents on all devices in the topology

Applied Filters: Data Center

Available Devices

Name	IP	Type	Agent	
sw-hdr-pr...	10.209.44.74	Switch	1.7.1-1	✓
ufm-sw-h...	10.209.225...	Switch	1.7.1-1	✓
ufm-sw-hdr...	10.209.225.2...	Switch	1.7.1-1	
r-ufm-sw-et...	10.209.36.3	Switch	99.99.99-9	
swx-sw-bm...	10.209.224.1...	Switch	1.7.1-1	

Viewing 1-5 of 14 | 5

Selected Devices

Name	IP	Type	Agent
sw-hdr-proto...	10.209.44.74	Switch	1.7.1-1
ufm-sw-hdr0...	10.209.225.212	Switch	1.7.1-1

Viewing 1-2 of 2 | 5

Users may also choose to skip the deployment step and proceed directly to validation by clicking **Skip & Start Validation**.

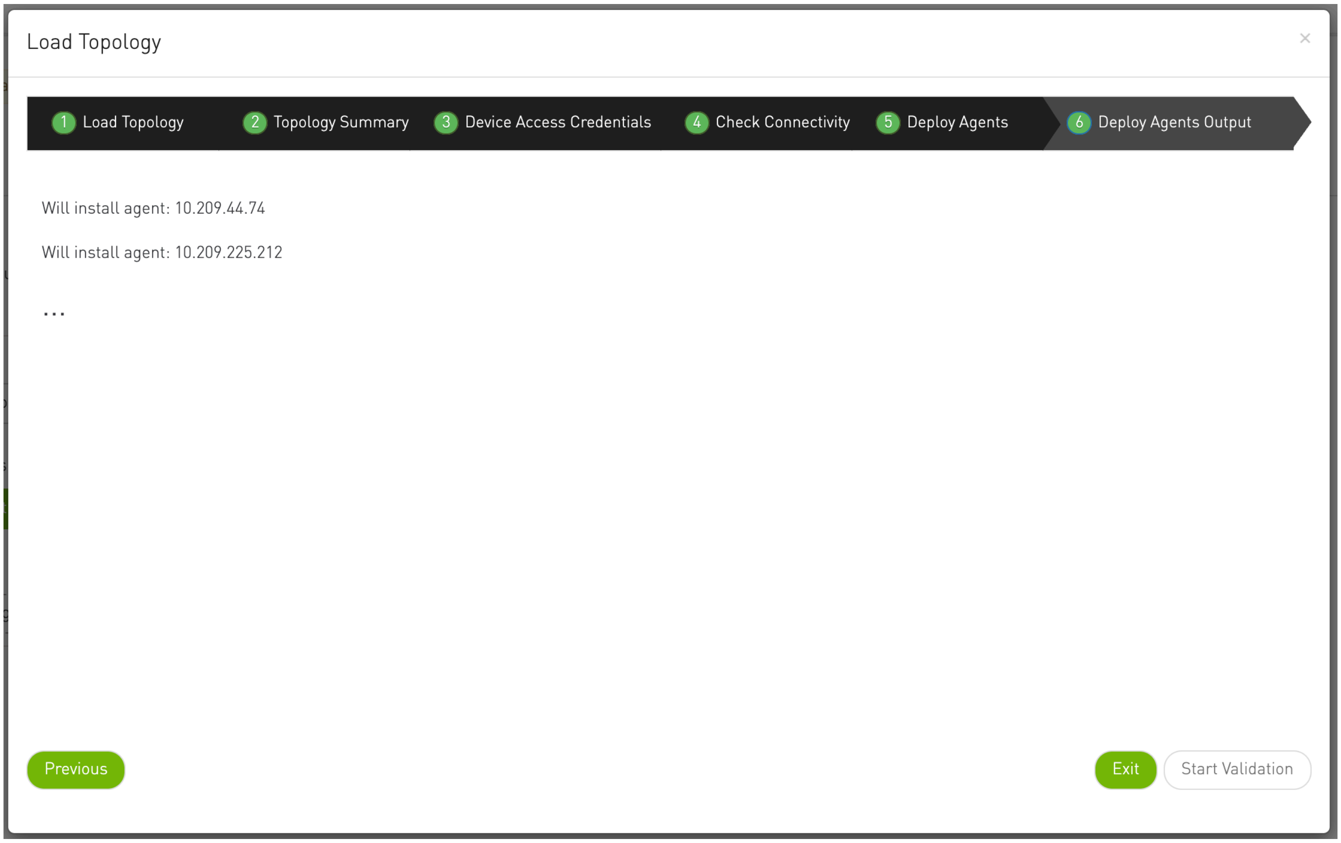
Note

To simplify validation, agent versions are now color-coded in the UI:

- **Green** → Version matches the bring-up version.
- **Orange** → Version does not match the bring-up version.
- **Black** → Not available (N/A)

Step 6 - Deploy Agents Output:

The output of the deployment process is displayed in the **Deploy Agents Output** tab, providing real-time visibility into deployment progress and results.



```
Will install agent: 10.209.44.74

Will install agent: 10.209.225.212

...
```

Manage Agents

The **Manage Agents** allows users to **deploy or remove agents** directly from the Services tab, without reloading the topology.

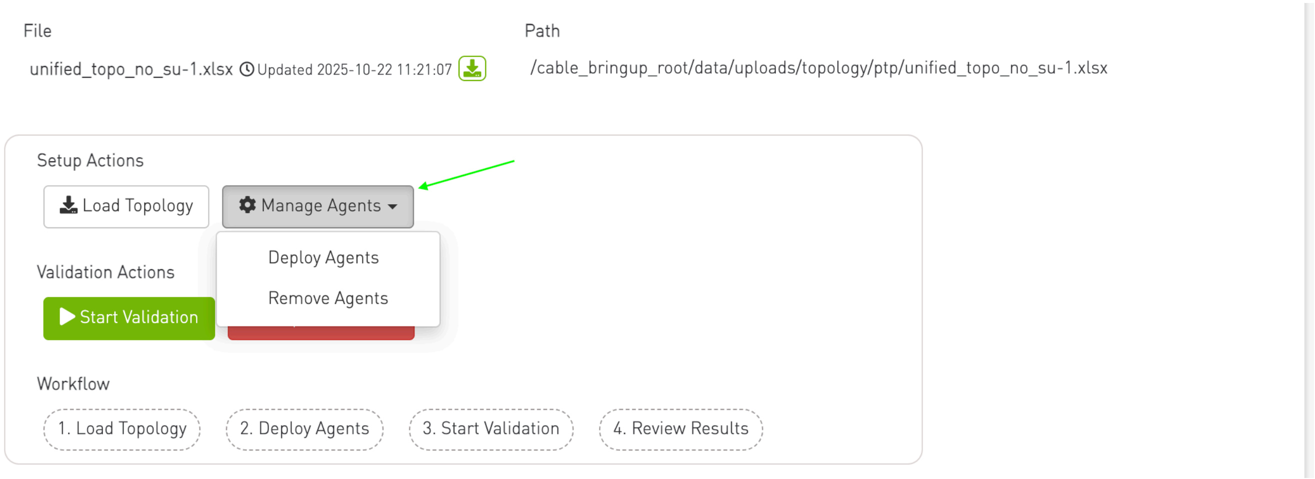
Deploying agents with the Load Topology wizard requires reloading the topology, and removing agents was not supported. Managing Agents simplifies both actions.

Clicking the **Manage Agents** button displays the following dropdown options:

- **Deploy Agents** – Deploy new agents.
- **Remove Agents** – Remove existing agents.

The button is **enabled only when the topology is loaded** and **validation is stopped**.

The wizard **skips the first two topology-related steps** and opens directly at the **Device Credentials** tab.

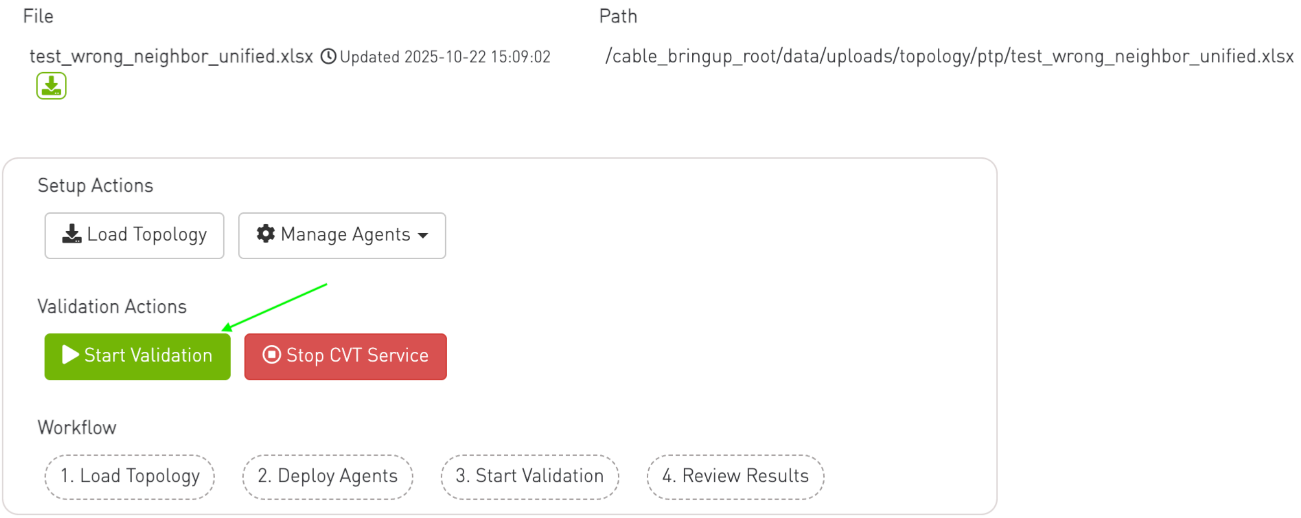

File: unified_topo_no_su-1.xlsx Updated 2025-10-22 11:21:07

Path: /cable_bringup_root/data/uploads/topology/ptp/unified_topo_no_su-1.xlsx

Start Validation

The **Start Validation** allows the user to initiate the validation process for the currently loaded topology.

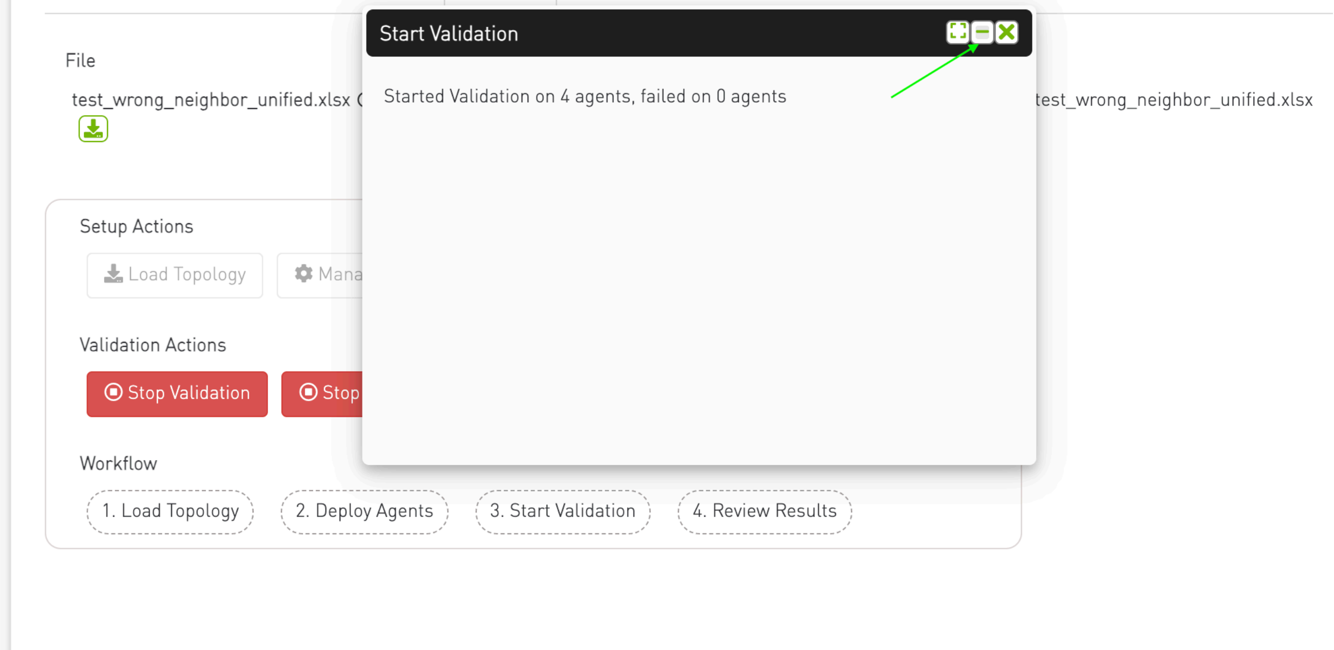
If no topology has been loaded, the **Start Validation** button remains disabled.


File: test_wrong_neighbor_unified.xlsx Updated 2025-10-22 15:09:02

Path: /cable_bringup_root/data/uploads/topology/ptp/test_wrong_neighbor_unified.xlsx

When the user clicks **Start Validation**, a pop-up window appears, providing real-time visibility into the validation progress and results. Users can minimize this window and continue navigating within the application while the validation runs in the background.

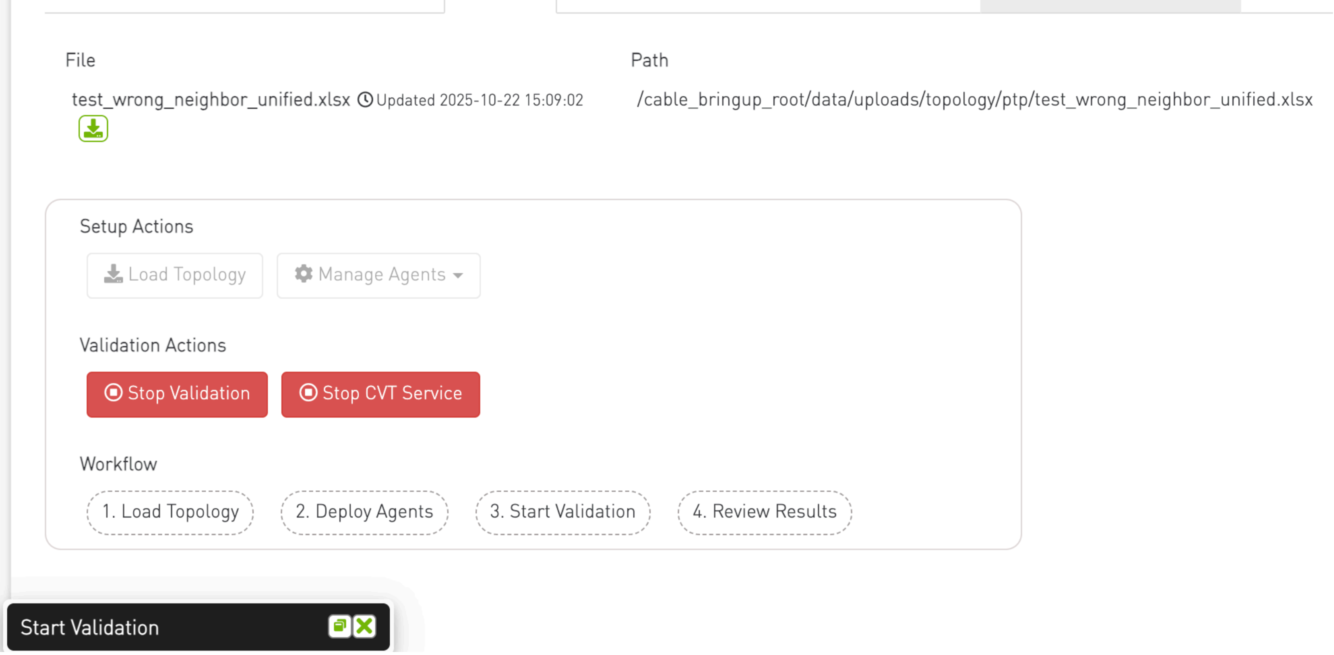
During the validation process, the **Close** button is disabled to prevent accidental closure until the validation results are ready for review.



Started Validation on 4 agents, failed on 0 agents

File: test_wrong_neighbor_unified.xlsx

test_wrong_neighbor_unified.xlsx



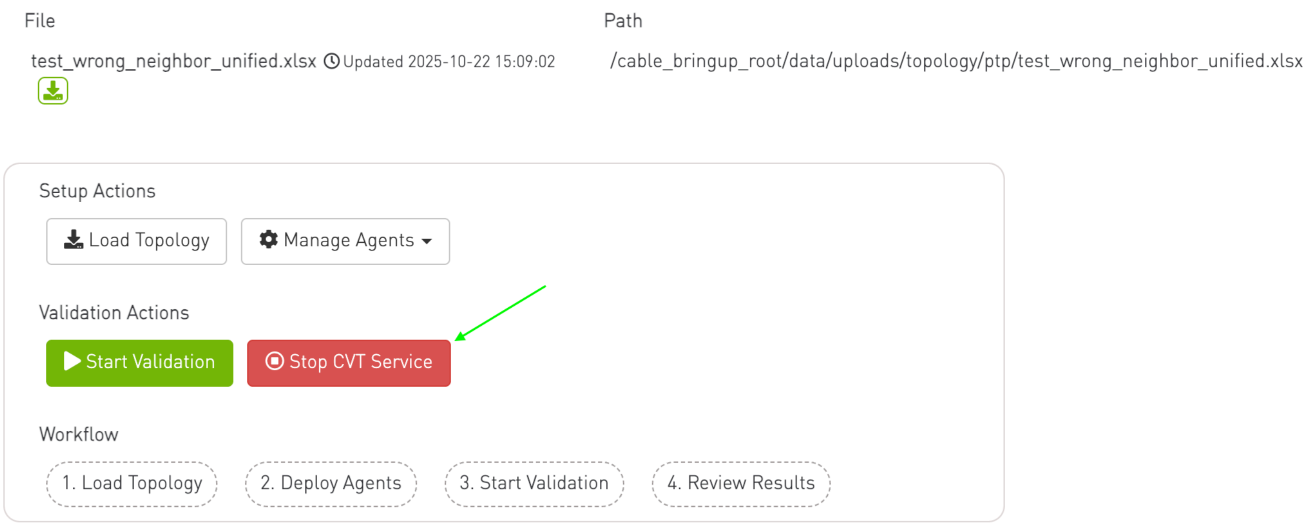
File: test_wrong_neighbor_unified.xlsx Updated 2025-10-22 15:09:02

Path: /cable_bringup_root/data/uploads/topology/ptp/test_wrong_neighbor_unified.xlsx

Users can stop the validation at any time by clicking the **Stop Validation** button.

Stop CVT Service

The **Stop CVT Service** is used to stop the currently running CVT service.


File

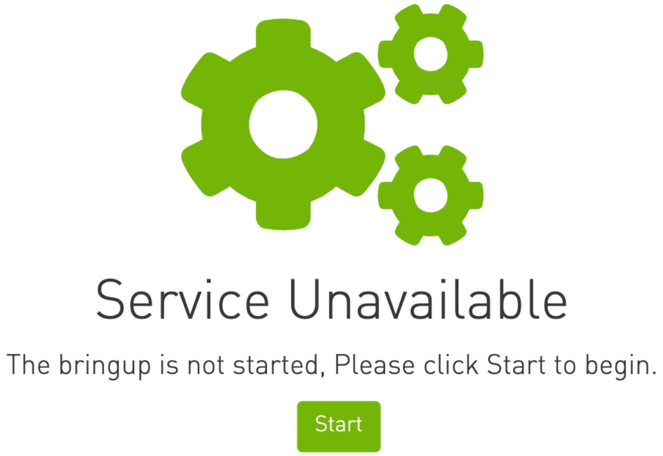
test_wrong_neighbor_unified.xlsx Updated 2025-10-22 15:09:02

Path

/cable_bringup_root/data/uploads/topology/ptp/test_wrong_neighbor_unified.xlsx

When clicked, the service is gracefully stopped, and the user is redirected to a dedicated page that displays the current service status.

From this page, the user can restart the service at any time by clicking the **Start** button.


Service Unavailable

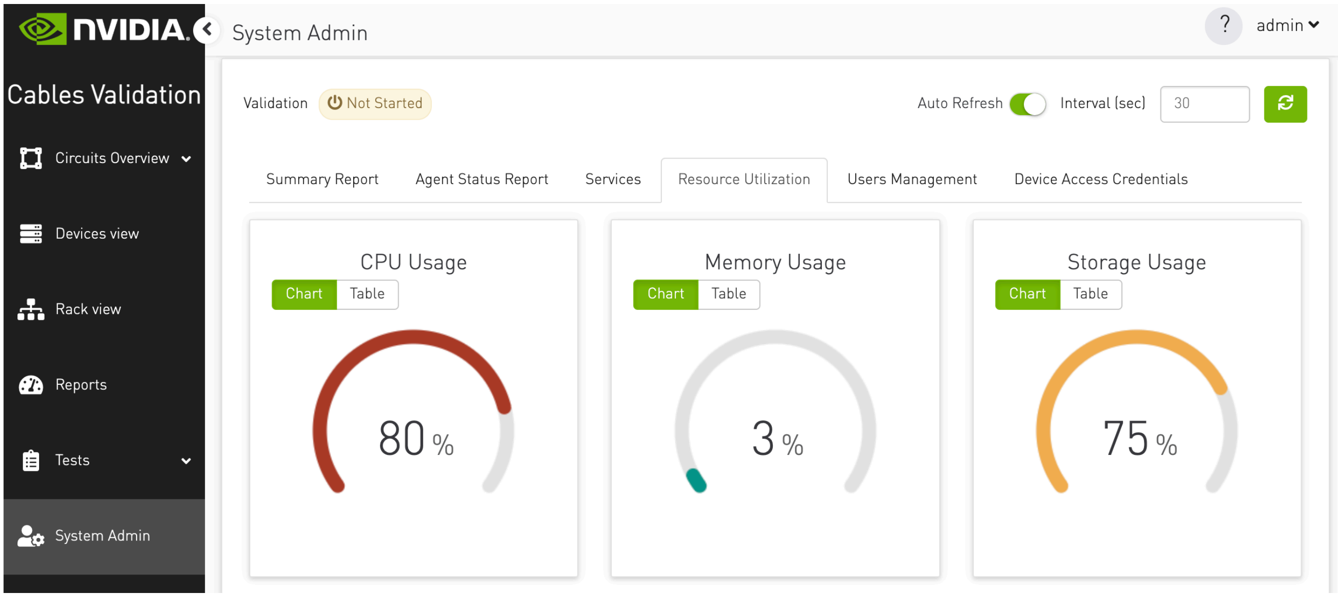
The bringup is not started, Please click Start to begin.

Resource Utilization

Resource utilization is used to track CVT system health and understand how it impacts resource consumption. It will be analyzed across three key areas: storage, memory, and CPU usage.

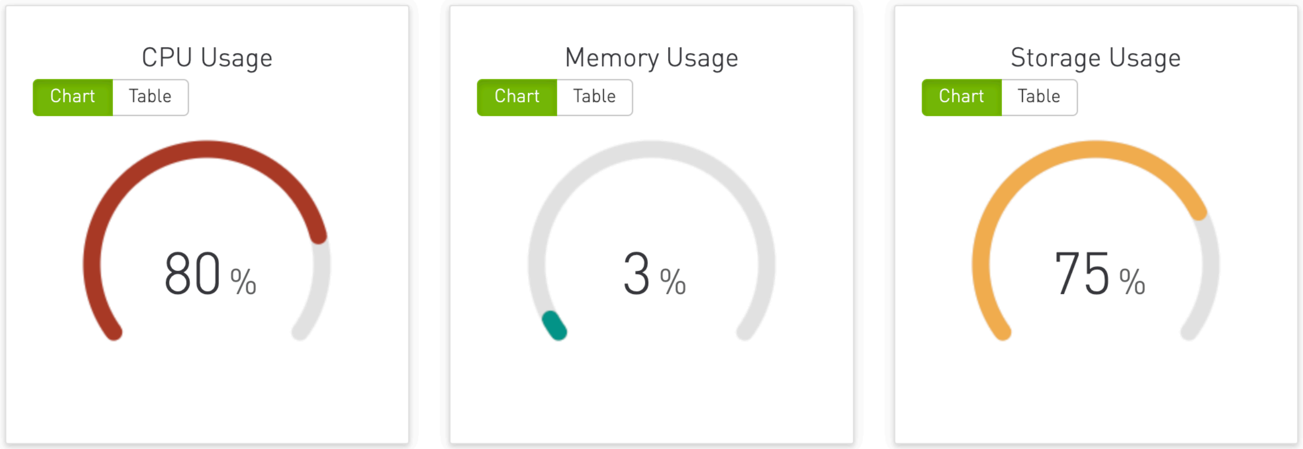
This analysis will help monitor system performance, optimize resource allocation, and ensure efficient operation.

CPU utilization is calculated over a 3-second interval, meaning it will take 3 seconds for the CPU utilization data to be available.

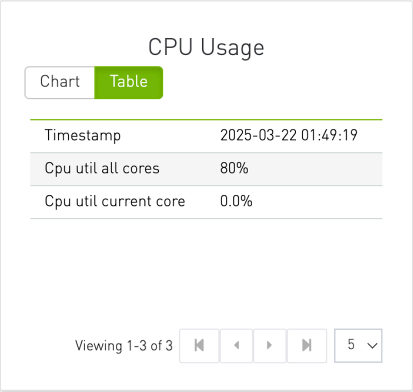


The tab will present resource data in two formats:

1. **Tachometer Chart**: A visual gauge that displays the usage percentage for each resource.

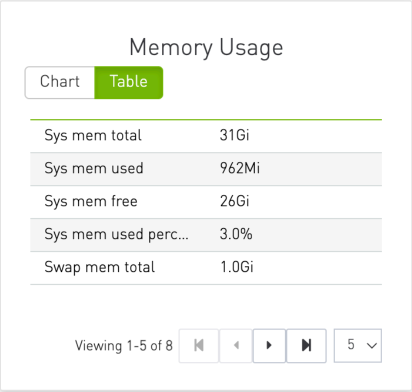


1. **Table**: A structured table displaying resource information in detail



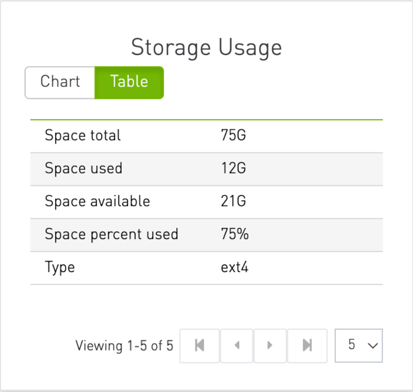
CPU Usage

Timestamp	2025-03-22 01:49:19
Cpu util all cores	80%
Cpu util current core	0.0%



Memory Usage

Sys mem total	31Gi
Sys mem used	962Mi
Sys mem free	26Gi
Sys mem used perc...	3.0%
Swap mem total	1.0Gi



Storage Usage

Space total	75G
Space used	12G
Space available	21G
Space percent used	75%
Type	ext4

To get the latest data, the user can refresh the view manually by clicking the refresh button or enable automatic refresh.

Tachometer Chart Color Thresholds

The color of the tachometer chart will be determined based on the percentage value of each resource, following these predefined thresholds:

CPU Usage Thresholds

0 - 50%	Normal	images/download/thumbnails/4465579677/image-2025-4-6_19-26-2-version-1-modificationdate-1762936875619-api-v2.png Blue
51 - 75%	Warning	□ Orange
76 - 100%	Critical	□ Red

Storage Usage Thresholds

0 - 70%	Normal	images/download/thumbnails/4465579677/image-2025-4-6_19-26-2-version-1-modificationdate-1762936875619-api-v2.png Blue
71 - 90%	Warning	□ Orange
91 - 100%	Critical	□ Red

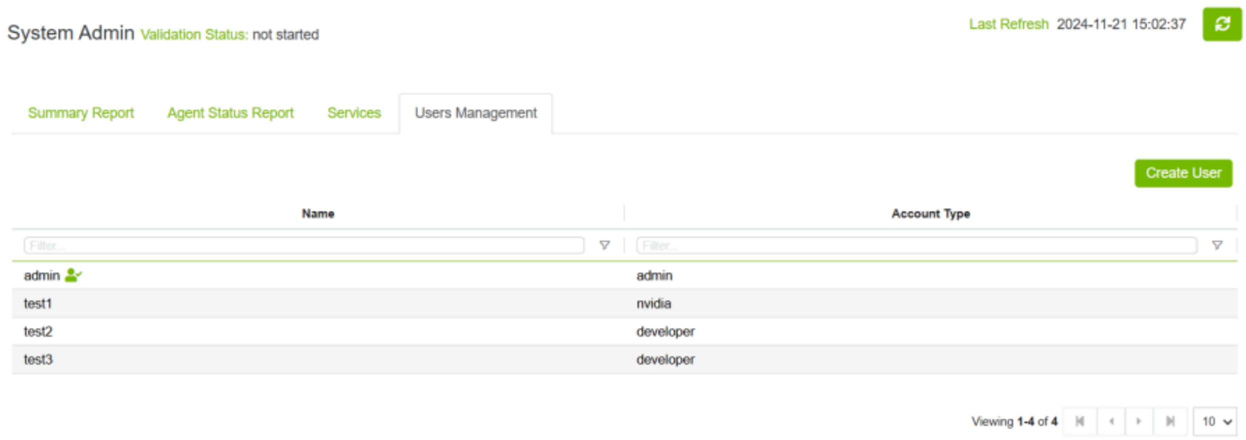
Memory Usage Thresholds

0 - 60%	Normal	images/download/thumbnails/4465579677/image-2025-4-6_19-26-2-version-1-modificationdate-1762936875619-api-v2.png Blue
61 - 80%	Warning	▯ Orange
81 - 100%	Critical	▯ Red

Users Management

The **User Management** view allows admin users to manage user accounts within the application. admin can create, update, or delete user accounts as needed.

The tab displays all users in a table, with a special icon indicating the currently logged-in user.

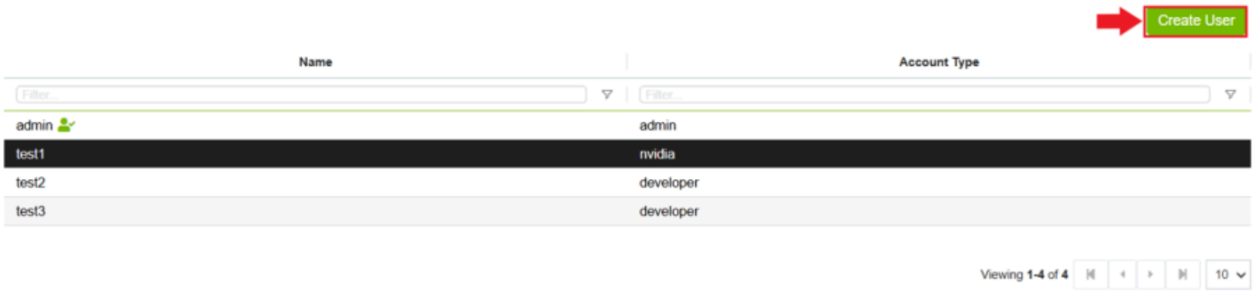


Name	Account Type
admin	admin
test1	nvidia
test2	developer
test3	developer

Create Users

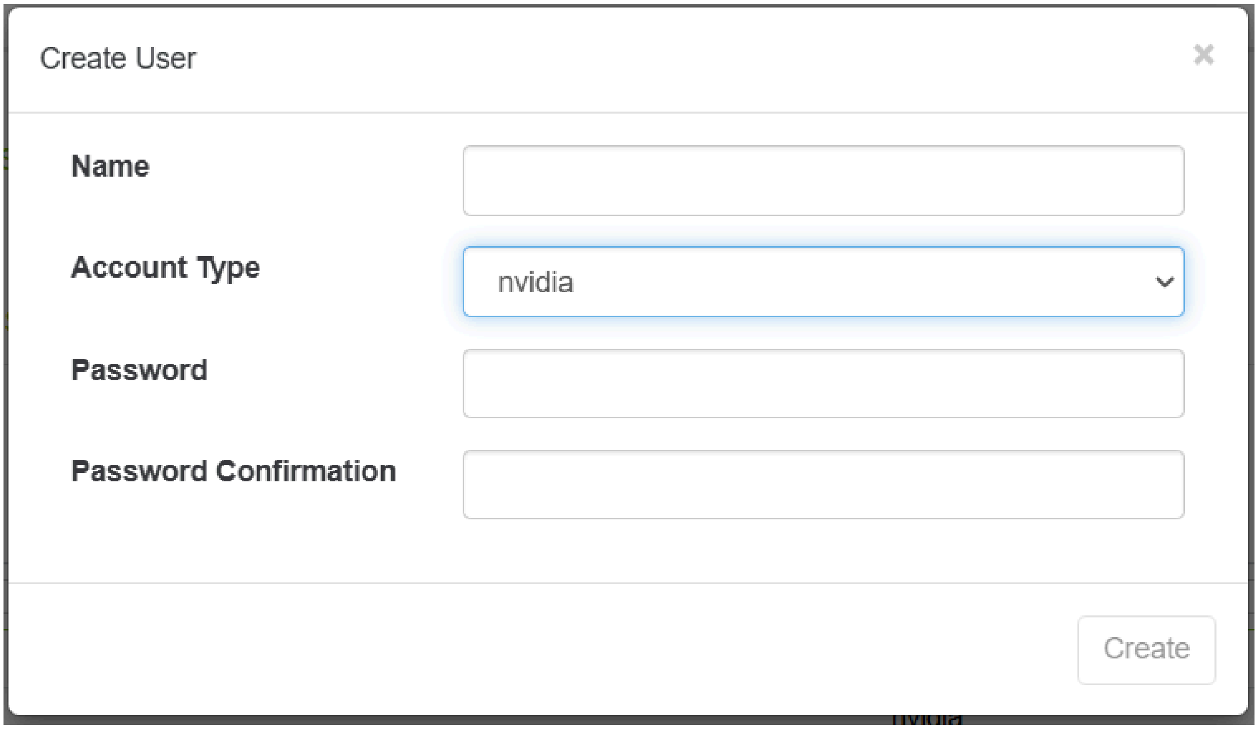
Admin can create new user accounts by following these steps:

1. Accessing the Create User Modal:
   - Click the Create User button to open a modal for adding a new user.



Name	Account Type
admin	admin
test1	nvidia
test2	developer
test3	developer

1. Supported Account Types:
   - The tool supports the following four account types:
     - admin: Full access to all features and settings.
     - nvidia: can access circuits view, rack view and reports only.
     - cabler: can access circuits view, rack view and reports only.
     - developer: can access circuits view, rack view and reports only.
2. Adding User Details:
   - Fill in the required fields in the modal to create the user account.
3. Completing the Process:
   - Click Create to add the new user to the system.

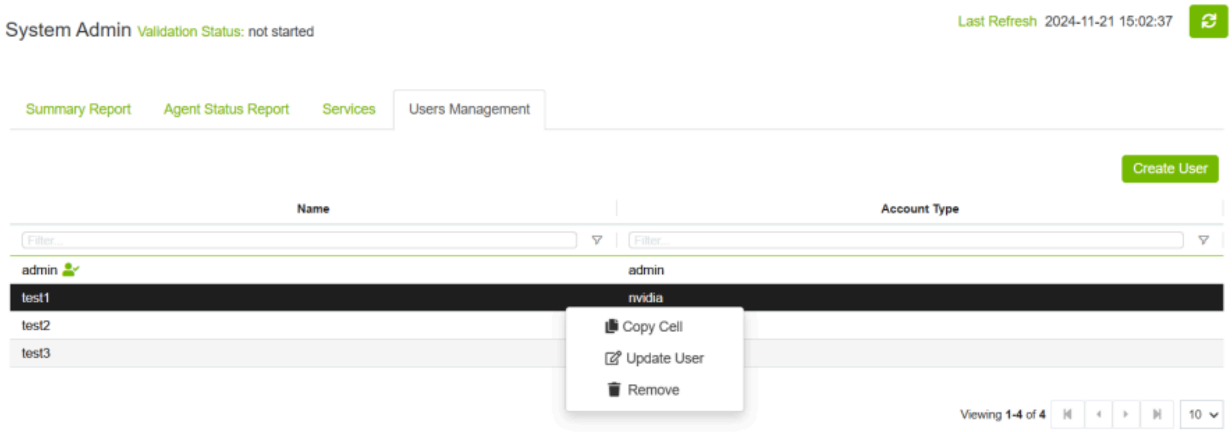


Account Type: nvidia

Update user

Admin user can also update the account type or the password or both for any user he wants

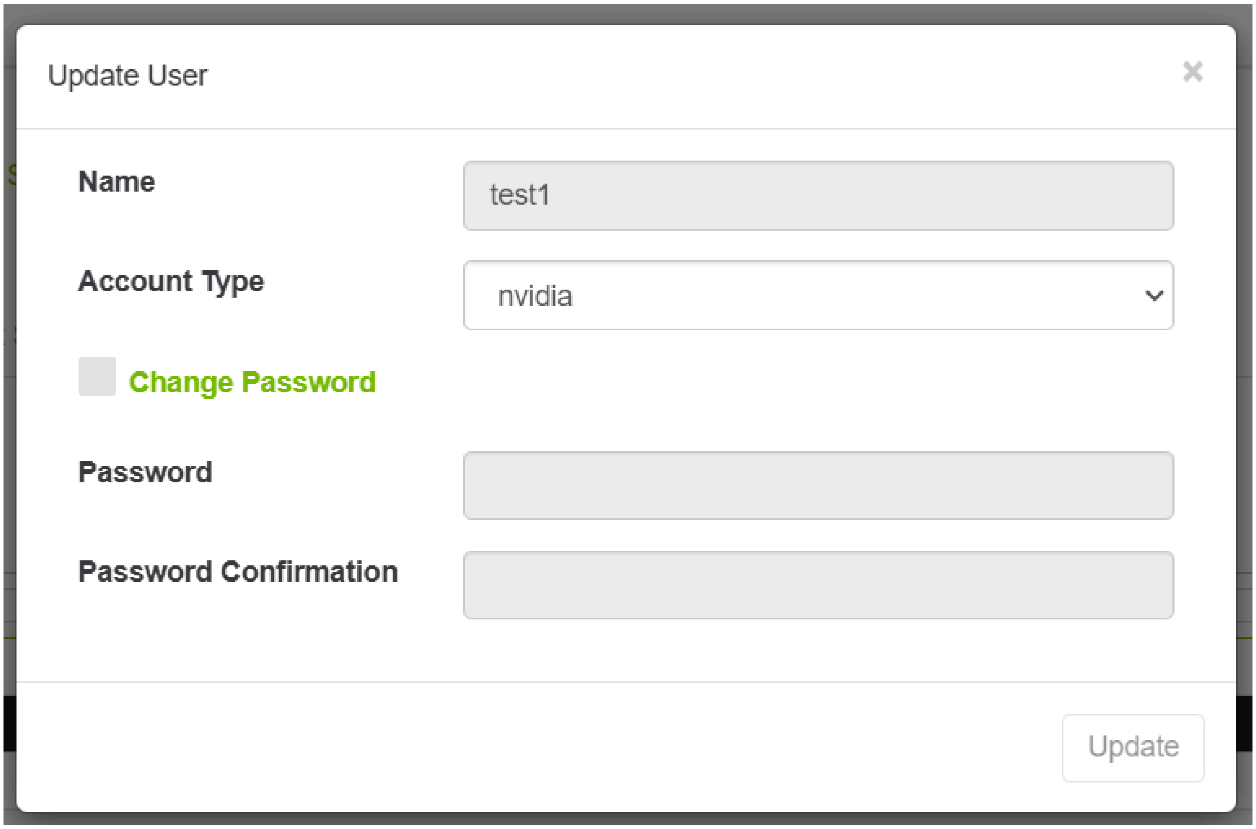
That can be done by right click then click on **Update User**.



Validation Status: not started

Last Refresh 2024-11-21 15:02:37

Name	Account Type
admin	admin
test1	nvidia
test2	
test3	

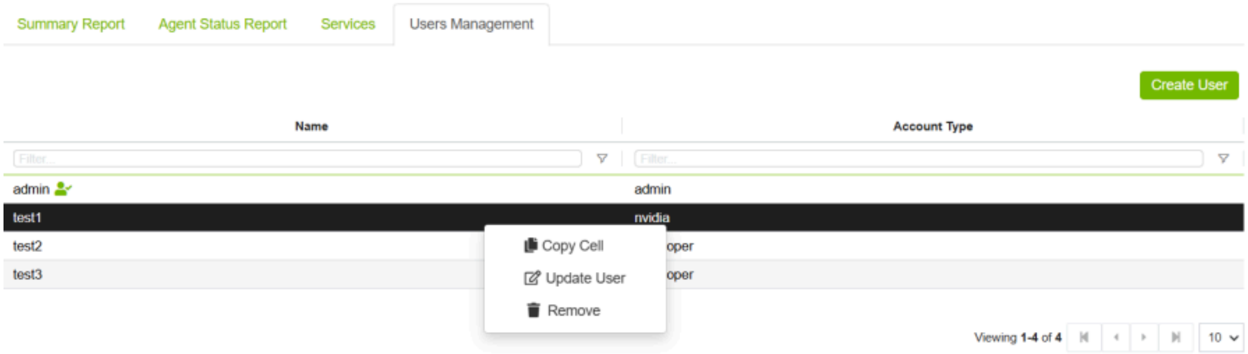


Name	test1
Account Type	nvidia
☐ Change Password	
Password	
Password Confirmation	

To change the password user, need to check change password checkbox then fill the passwords input.

Delete Use

Admin user can also delete the user by right click then click on **Remove User**.



Name	Account Type
admin	admin
test1	nvidia
test2	oper
test3	oper

Device Access Credentials

The Device Access Credentials Management component allows you to securely manage credentials (username, password, etc.) for devices in your system.

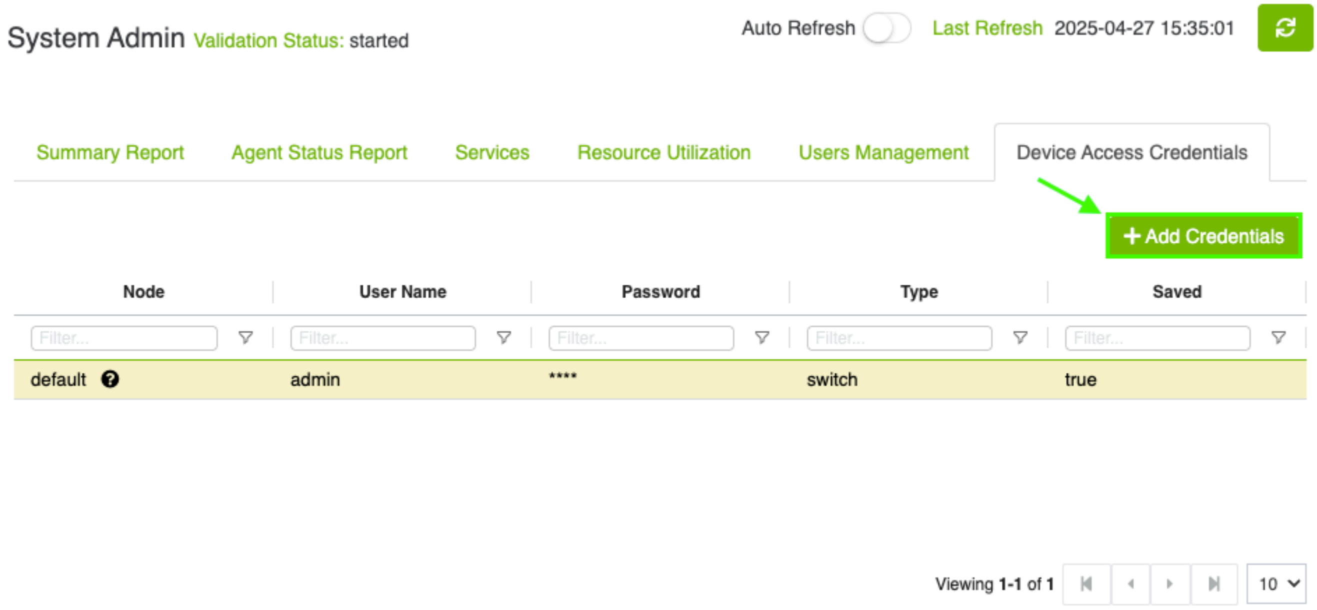
You can add, view, update, and remove device access credentials through a table and modal interface.

Add Credentials

Admin can add new Credentials by following these steps:

1. Accessing the Credentials Modal:

-
    - Click the 'Add Credentials' button to open a modal for adding a new credential.



Node	User Name	Password	Type	Saved
default	admin	****	switch	true

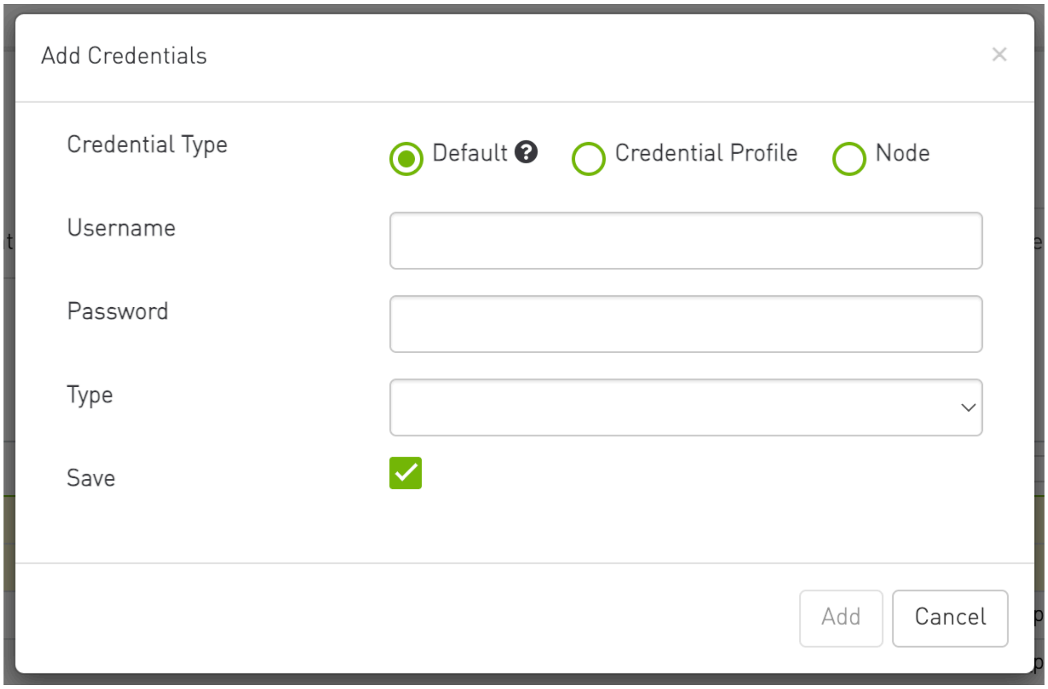
2. Fill in the required fields:

-
    - Credential Type: Select the type of credentials you want to set.

      Precedence:

        - Node (Highest Precedence)
        - Credential Profile
        - Default (Least Precedence)

-
    - Node IP: Enter the device IP (only if `Node`)
    - Profile Name: Enter the credential profile name(only if `Credential Profile`).
-
    - Username: Enter the username for device access.
-
    - Password: Enter the password.
-
    - Type: Select the device type (only if `Default`).
-
    - Save: Check if you want to save these credentials.



Add Credentials

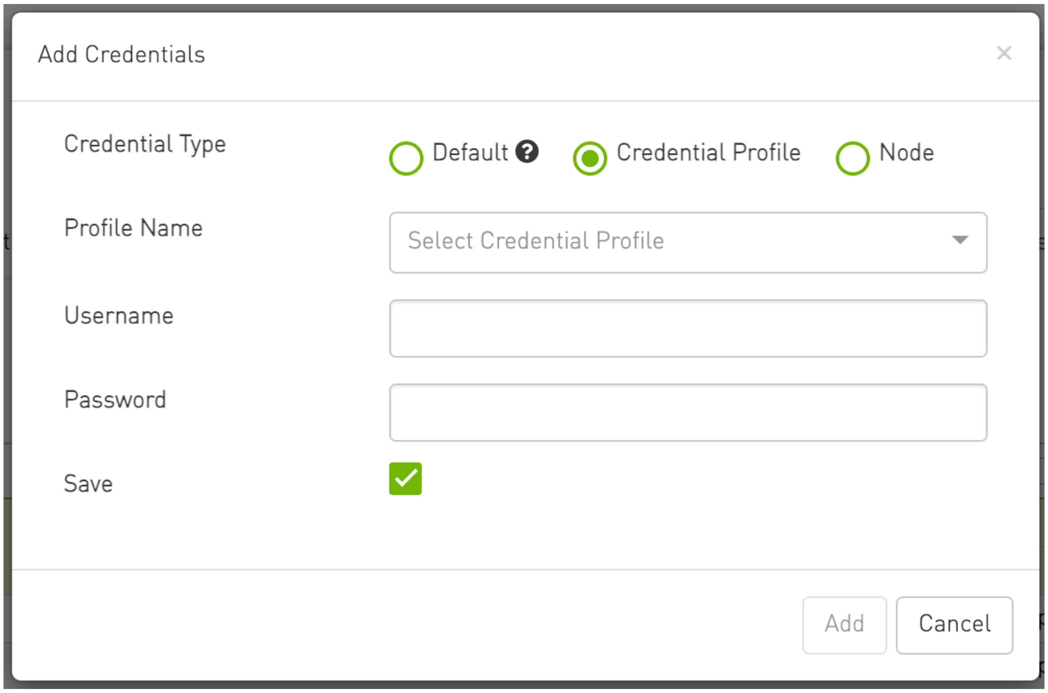
Credential Type ● Default ○ Credential Profile ○ Node

Username

Password

Type

Save ☑



Add Credentials

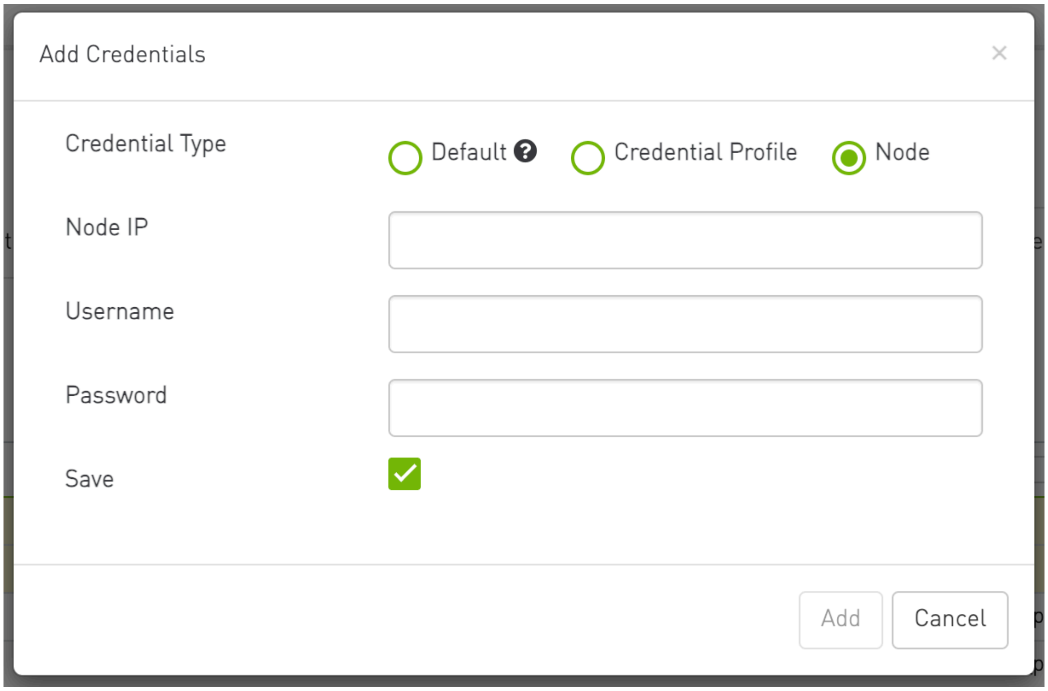
Credential Type: ○ Default ● Credential Profile ○ Node

Profile Name: Select Credential Profile

Username:

Password:

Save: ☑



Add Credentials

Credential Type: ○ Default ○ Credential Profile ● Node

Node IP:

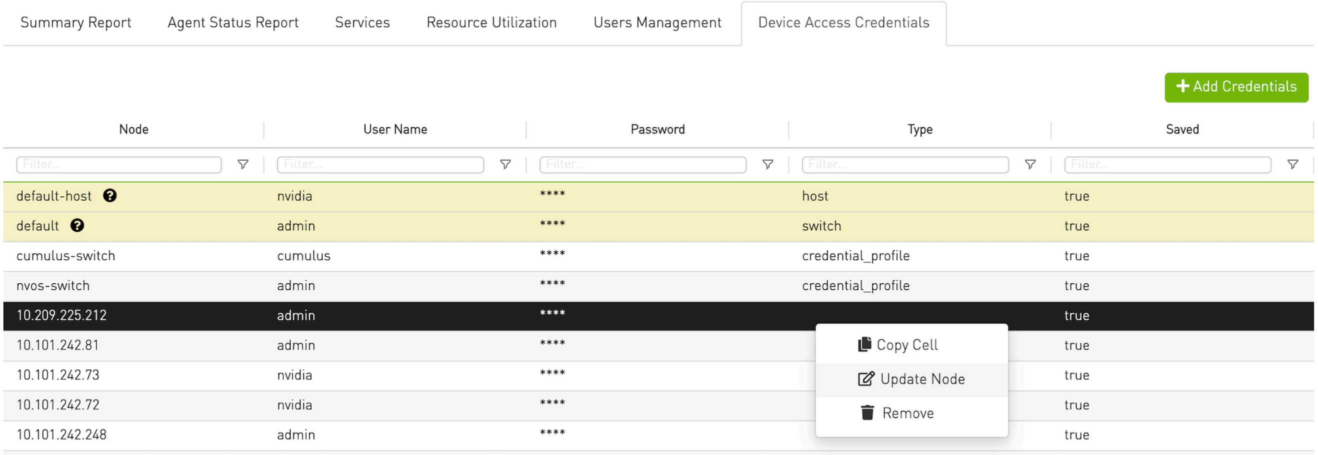
Username:

Password:

Save: ☑

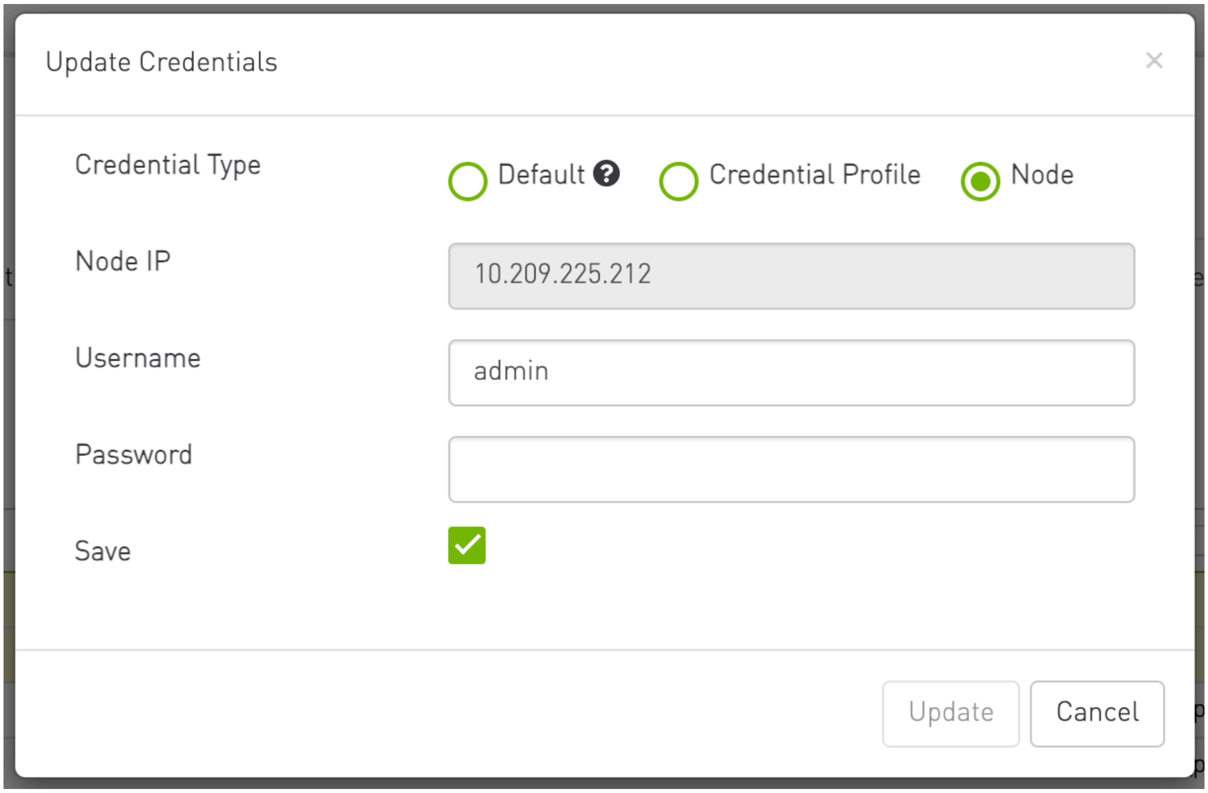
Update Existing Credentials

Admin user can also update the existing credentials by right click then clicking on **Update Node**.



Node	User Name	Password	Type	Saved
default-host	nvidia	****	host	true
default	admin	****	switch	true
cumulus-switch	cumulus	****	credential_profile	true
nvos-switch	admin	****	credential_profile	true
10.209.225.212	admin	****		true
10.101.242.81	admin	****		true
10.101.242.73	nvidia	****		true
10.101.242.72	nvidia	****		true
10.101.242.248	admin	****		true

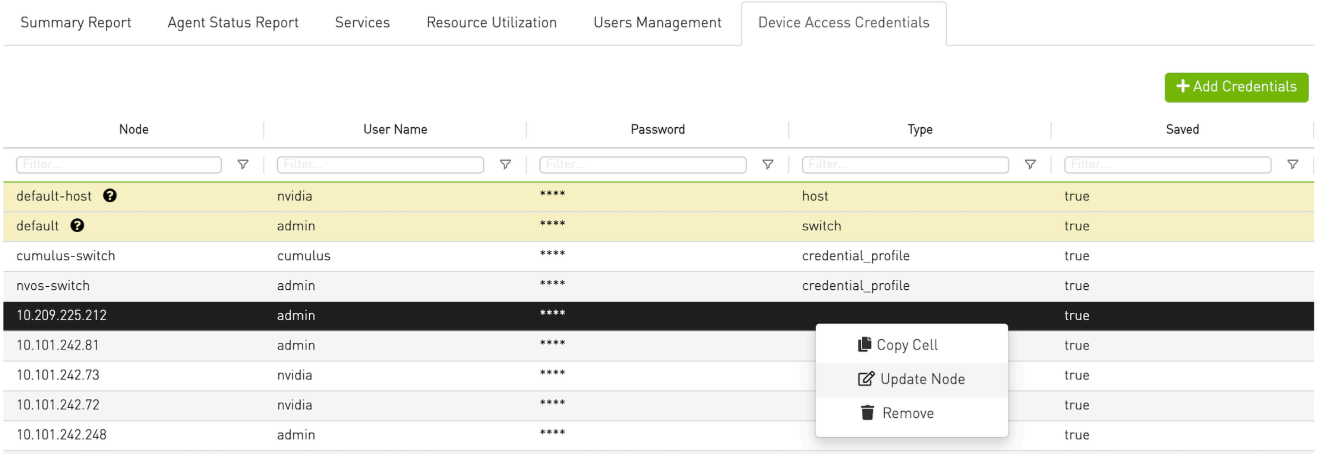
The user can update the username, password, and save status. Updating the password is mandatory when updating the credentials.



Credential Type	○ Default ○ Credential Profile ● Node
Node IP	10.209.225.212
Username	admin
Password	
Save	☑

Remove Credentials

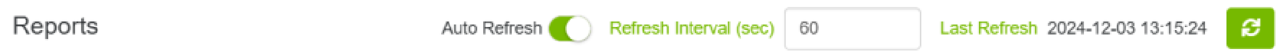
Admin user can also remove the credentials by right click then click on **Remove**. only non-default credentials can be removed



Node	User Name	Password	Type	Saved
default-host	nvidia	****	host	true
default	admin	****	switch	true
cumulus-switch	cumulus	****	credential_profile	true
nvos-switch	admin	****	credential_profile	true
10.209.225.212	admin	****		true
10.101.242.81	admin	****		true
10.101.242.73	nvidia	****		true
10.101.242.72	nvidia	****		true
10.101.242.248	admin	****		true

Auto Refresh

Most pages in the tool contain auto refresh feature to ensure content remains up-to-date.

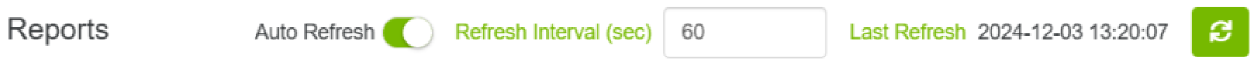


Auto Refresh
Refresh Interval (sec) 60
Last Refresh 2024-12-03 13:15:24

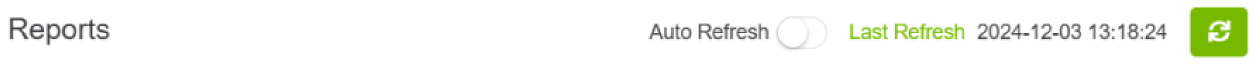
Users can perform the following actions related to the auto-refresh functionality:

Enable or Disable Auto-Refresh

Toggle the auto-refresh feature on or off based on your preferences.



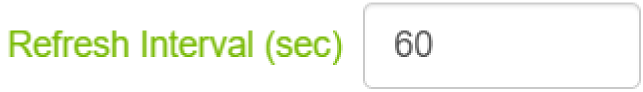
Auto Refresh
Refresh Interval (sec) 60
Last Refresh 2024-12-03 13:20:07



Auto Refresh
Last Refresh 2024-12-03 13:18:24

Adjust the Refresh Interval

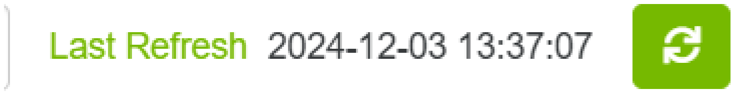
Customize the interval time for auto-refresh to control how frequently the page updates automatically.



Refresh Interval (sec) 60

Manually Refresh the Page

Trigger a manual refresh of the page at any time to view the latest updates immediately.



Last Refresh 2024-12-03 13:37:07

Table

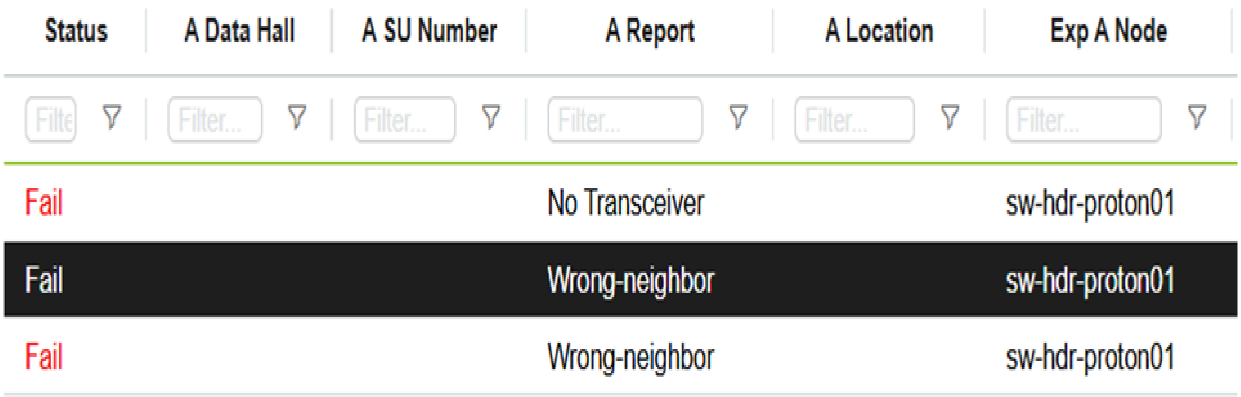
All tables in the tool contain a general functionality, this chapter will describe them.

Filter

The tables provide advanced search capabilities, offering users two primary ways to utilize the search mechanism:

1. **Search Input Field Below the Table Name:**

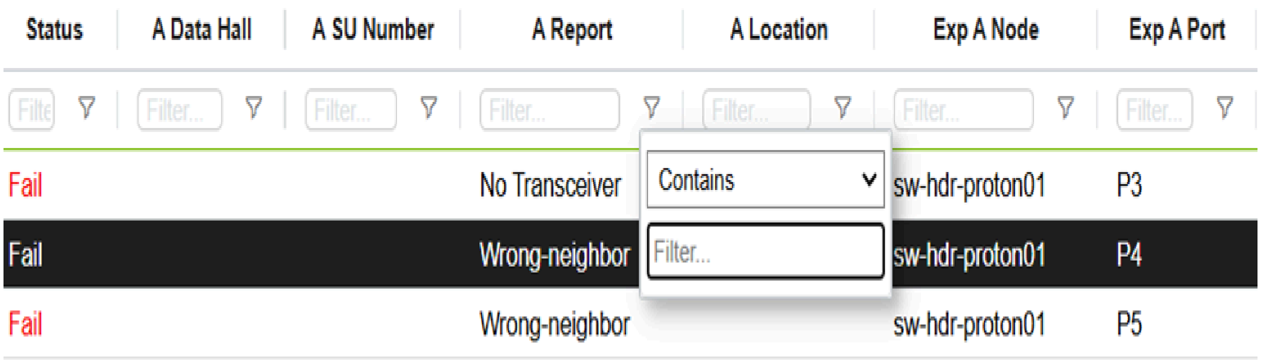
   A straightforward search field located beneath the table name allows users to quickly filter table content based on their input and selected filter type.



Status	A Data Hall	A SU Number	A Report	A Location	Exp A Node
Filte ▽	Filter... ▽	Filter... ▽	Filter... ▽	Filter... ▽	Filter... ▽
Fail			No Transceiver		sw-hdr-proton01
Fail			Wrong-neighbor		sw-hdr-proton01
Fail			Wrong-neighbor		sw-hdr-proton01

2. **Search Menu via the Column Search Icon:**

   Clicking the search icon beside the search input opens a dedicated search menu. This menu provides additional options and customization for filtering data.



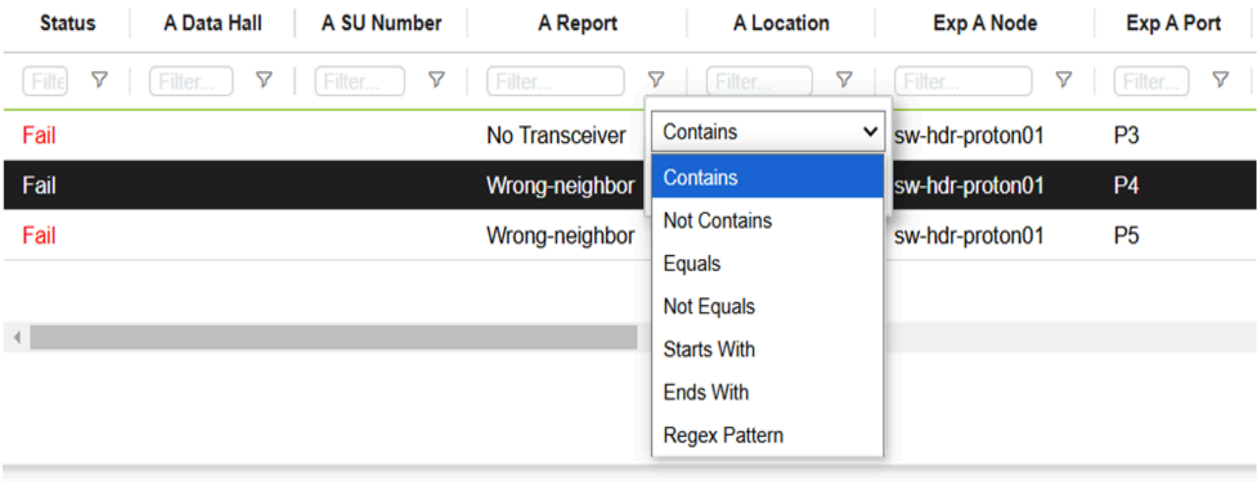
Status	A Data Hall	A SU Number	A Report	A Location	Exp A Node	Exp A Port
Fail			No Transceiver		sw-hdr-proton01	P3
Fail			Wrong-neighbor		sw-hdr-proton01	P4
Fail			Wrong-neighbor		sw-hdr-proton01	P5

Users can change the search type **only** through the search menu accessible via the column search icon.

Supported Filter Types

The table supports the following types of filters to refine search results:

1. **Contains:** Finds records that include the specified text.
2. **Does Not Contain:** Excludes records that include the specified text.
3. **Equals:** Displays records that exactly match the specified value.
4. **Does Not Equal:** Excludes records that exactly match the specified value.
5. **Starts With:** Finds records that begin with the specified text.
6. **Ends With:** Finds records that end with the specified text.
7. **Regular Expression (Regex) Pattern:** Enables advanced searches using custom regular expressions for precise filtering.



Status	A Data Hall	A SU Number	A Report	A Location	Exp A Node	Exp A Port
Fail			No Transceiver		sw-hdr-proton01	P3
Fail			Wrong-neighbor		sw-hdr-proton01	P4
Fail			Wrong-neighbor		sw-hdr-proton01	P5

Resource Filter

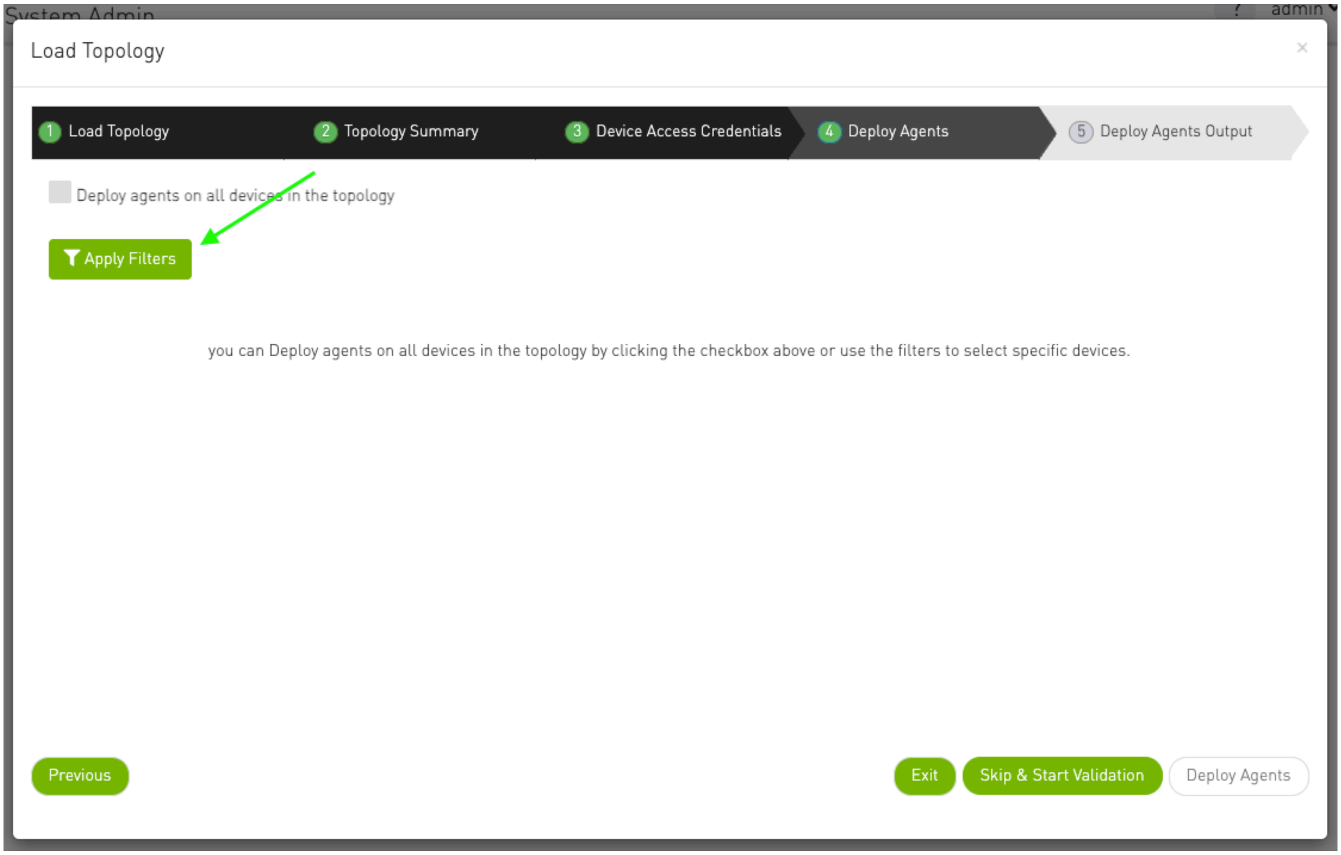
The **Resource Filter** is a shared filtering mechanism within the **Cable Validation Tool (CVT)** that provides a unified and flexible way to refine data across multiple views and workflows. It is designed to enhance usability, consistency, and efficiency in data selection.

This filter is used in several key areas of the application, such as the **Circuits View** and the **Deploy Agents** step of the **Deploy Agent Wizard**.

The main purpose of the Resource Filter is to help users easily narrow down large datasets—such as circuits or agents—based on hierarchical or contextual criteria.

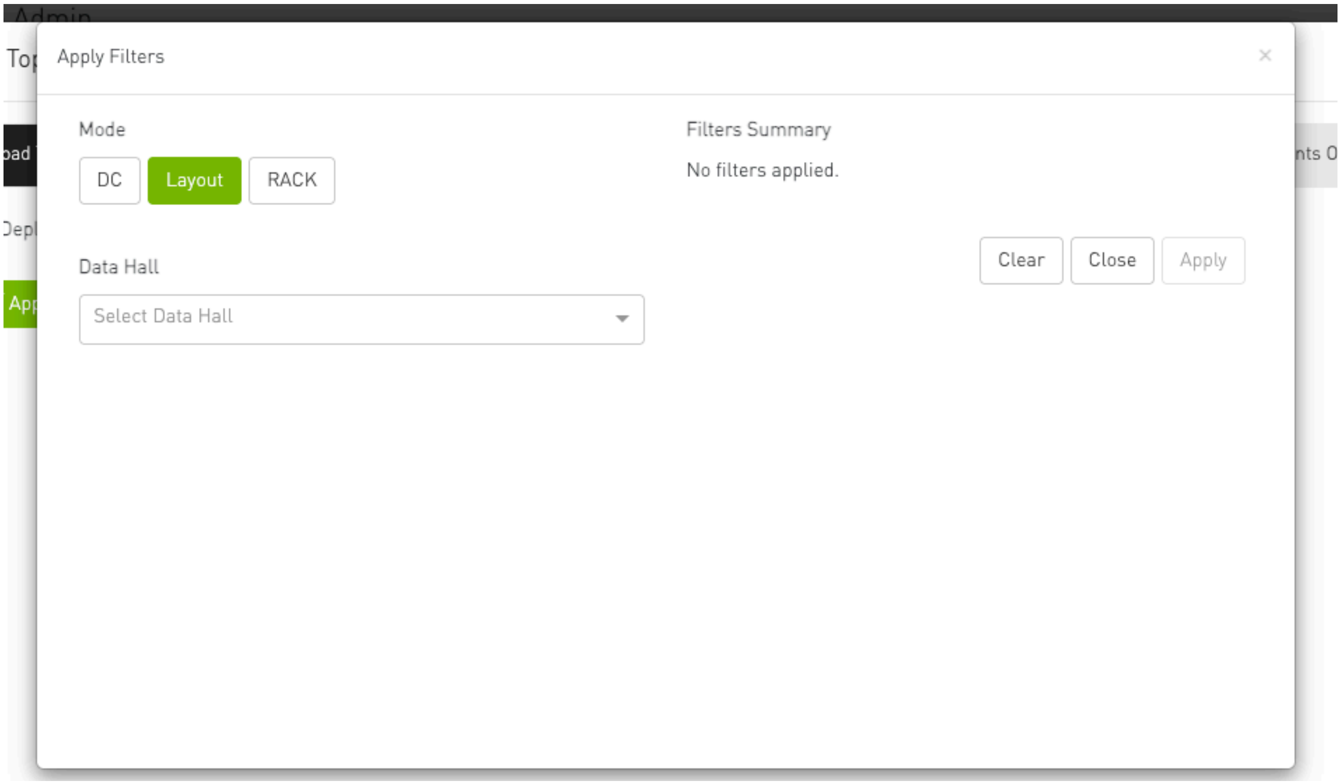
The filter is accessible through a **Filter** button that appears on any page or component supporting this functionality.

For example, in the **Deploy Agents** step of the **Deploy Agent Wizard**, the **Filter** button appears as shown below:



you can Deploy agents on all devices in the topology by clicking the checkbox above or use the filters to select specific devices.

When the user clicks the **Filter** button, a modal window opens, allowing them to define their filtering criteria.



Filters Summary

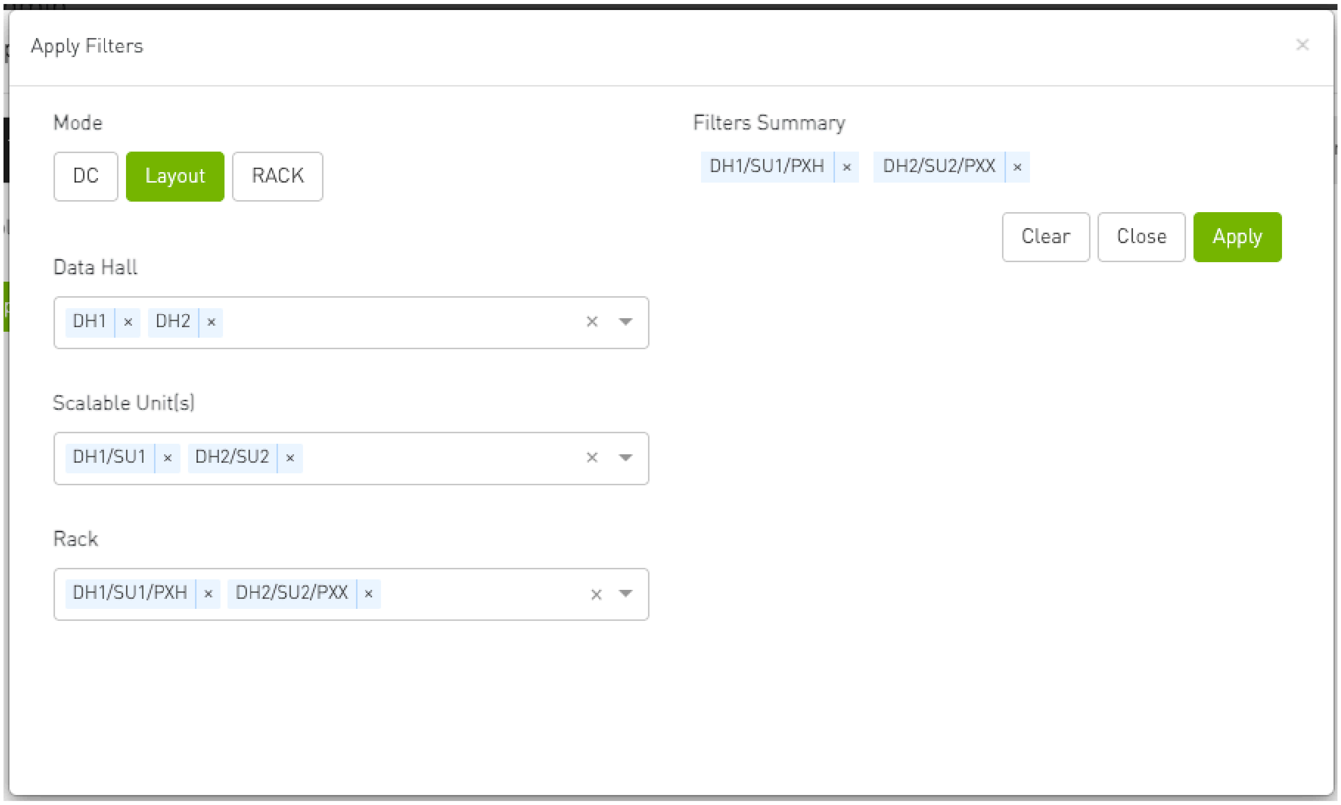
No filters applied.

Filter Modes

The **General Filter Modal** supports three distinct modes, providing flexibility depending on the user's context and workflow:

Layout (Hierarchical Mode)

- Enables drill-down selection through topology layers: **Data Hall → SU → Rack**.
- Dynamically filters devices as the user navigates through each level.
- Provides a clear hierarchical context for every selection step.



Mode: Layout

Data Hall: DH1, DH2

Scalable Unit(s): DH1/SU1, DH2/SU2

Rack: DH1/SU1/PXH, DH2/SU2/PXX

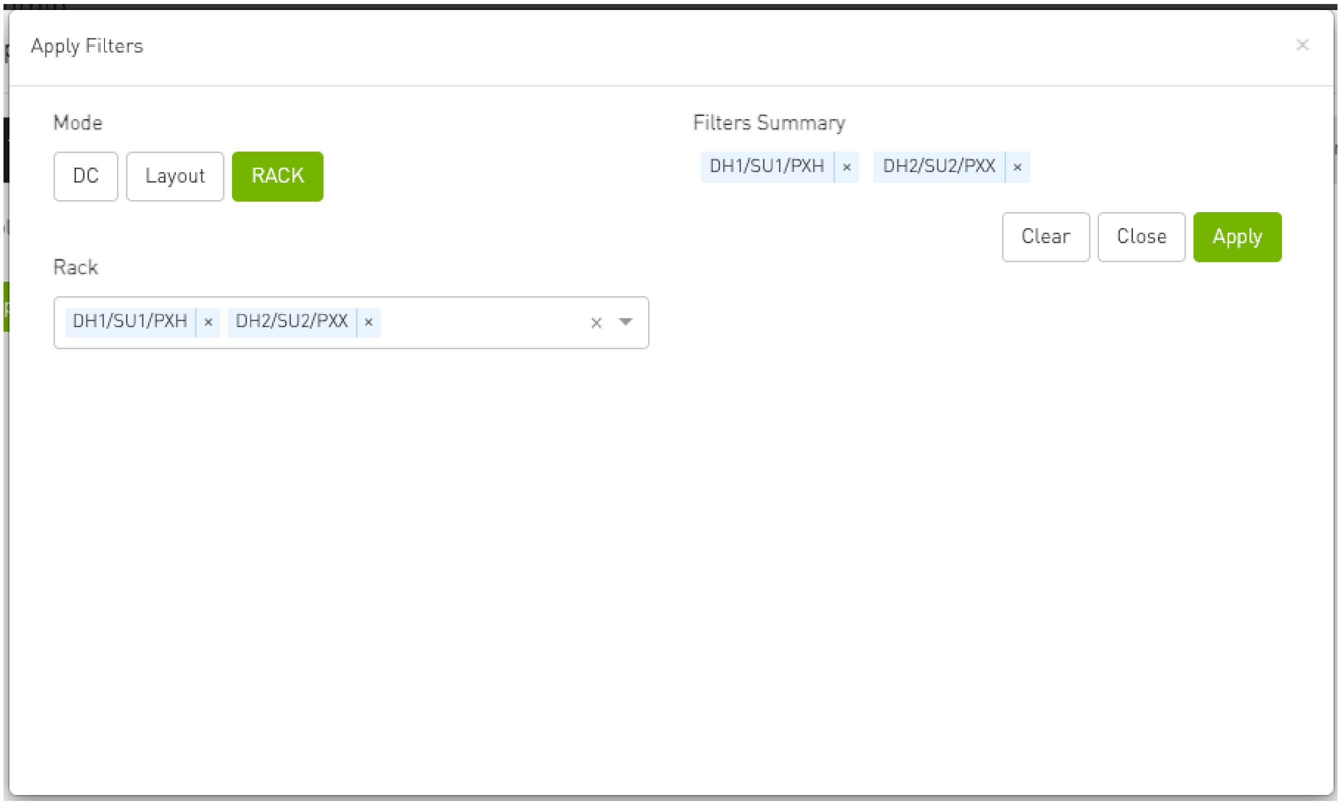
Filters Summary: DH1/SU1/PXH, DH2/SU2/PXX

Data Center (DC Mode)

- Displays all available resources (e.g., devices) within the selected data center.
- Not preferred with large-scale environments.

1.4.1.3. Rack Mode

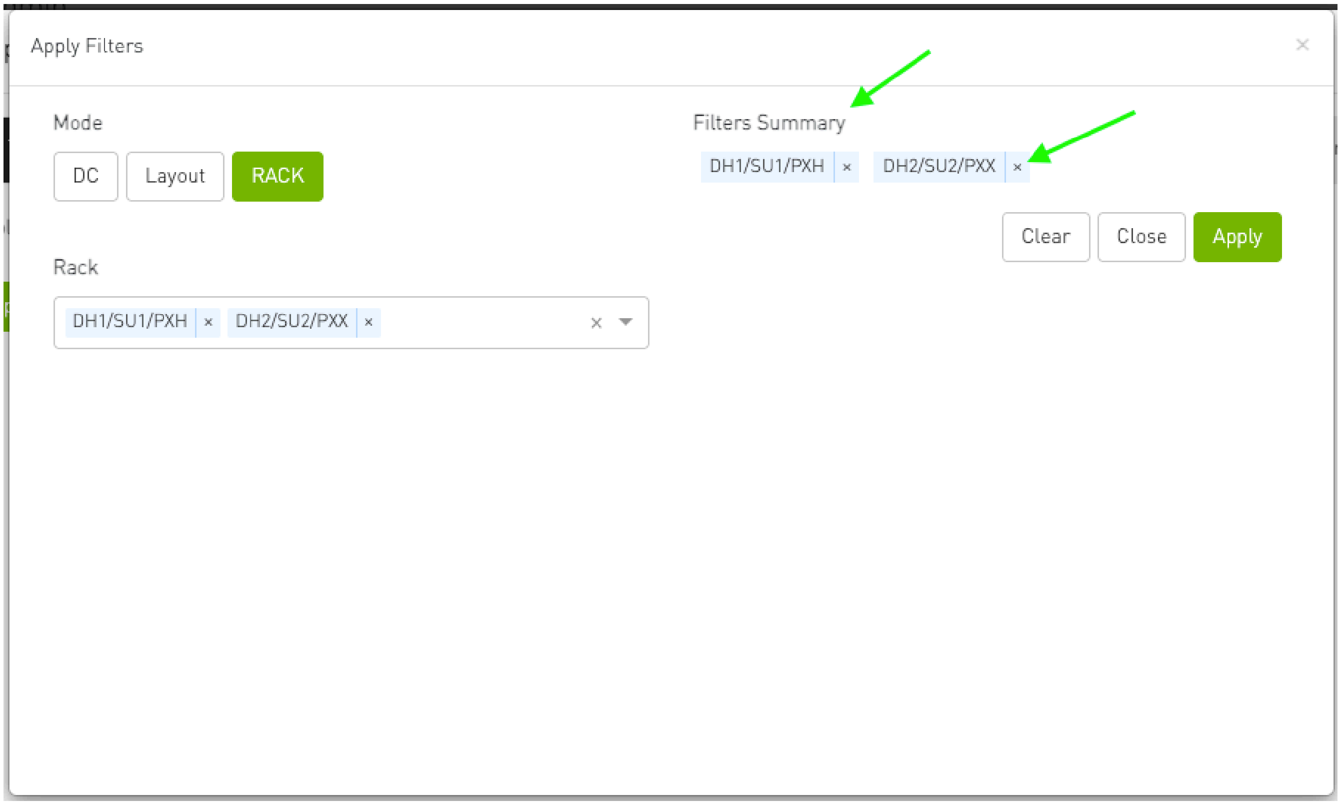
- Provides a flat, searchable list of all racks in the topology.
- Useful for systems that do not include Data Halls or SUs, or when the rack name is already known.



Filter Summary and Results

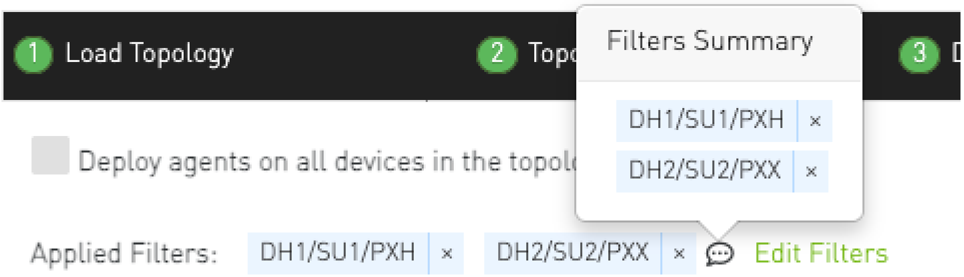
A **summary panel** is displayed on the left side of the modal, showing all selected filters.

Users can review and remove individual filters directly from this panel before applying the filter.



Once **Apply** is clicked:

- The system filters the resources or devices according to the selected criteria.
- The applied filter details appear beside the **Filter** button for visibility, as shown below:



Note

The Resource Filter remembers your last applied filters when you navigate between pages that share the same filtering mechanism.

For example, a filter applied in **Circuit View** will automatically appear in **Flapping Circuit View**.

However, filters that are opened inside a **modal window** (for example, within a wizard or pop-up dialog) **are not remembered**.

Each time you reopen the modal, you’ll need to set your filters again.

Rest APIs

Bringup Server REST API

The collector has a web server listening on two internal port `8251`. This port is not advertized outside the machine. The bringup server is running an *Apache* server which uses the default https port.

Overriding Collector Web server Port

Users can modify the collector web server port as the following:

- **Plugin**

1.
    1. Execute **/opt/ufm/scripts/manage_ufm_plugins.sh add -p cablevalidation** to add the Cable Validation plugin.
    2. Stop the plugin using **/opt/ufm/scripts/manage_ufm_plugins.sh stop -p cablevalidation**
    3. Use **vim /opt/ufm/files/conf/plugins/cablevalidation/config.cfg** to modify the **'BRINGUP_PORT'** variable.
    4. Update and save the file.
    5. Start the plugin again with **/opt/ufm/scripts/manage_ufm_plugins.sh start -p cablevalidation**.

- **Standalone**

1.
    1. Pass the new port as Env by executing **docker run --name cables_bringup -itd --network=host --env BRINGUP_PORT=<new-port> cables_bringup**

With these changes, the new configuration will take effect, and Apache will run with the updated ports.

Reports APIs

Get Validation Report

- Description – Gets the recent cable validation report
- Request URL – `GET /cablevalidation/report/validation`

- Response Content Type – application/json
- Status Codes
    - 200 – OK
    - 404 – NOT FOUND
- Response Example

```
{
   "report" :  "ValidationReport" ,
   "stats" :  {
      "in_progress" :  14 ,
      "no_issues" :  8 ,
      "not_started" :  5
   } ,
   "issues" :  [
      {
         "timestamp" :  1722290755.0392804 ,
         "node_desc" :  "dell001.cm.cluster" ,
         "rack" :  "D01" ,
         "unit" :  2 ,
         "issues" :  [
            [  "Unreachable-device" ]
         ]
      }
   ] ,
   "metadata" :  {
      "file_name" :  "v22_testing_extracable.xlsx" ,
      "file_hash" :
"d2a8ce40b613a17245acb3310fa0b810238dc61613e8f683b95139d19567aa5a" ,
      "load_time" :  1722290305.0386324
   }
}
```

Get Validation Status

- Description – Gets validation status.
- Request URL – `GET /cablevalidation/validation/status`
- Response Content Type – application/json
- Status Codes
  - 200 – OK
  - 500 – INTERNAL SERVER ERROR
- Response Data Example:

```
{"status" :  "not started"}
```

Get Topology Metadata

- Description – Gets topology metadata.
- Request URL – `GET /cablevalidation/topology/metadata`
- Response Content Type – application/json
- Status Codes
  - 200 – OK
  - 400 – BAD REQUEST
- Response Data Example

```
{
"file_name" :  "proton-ptp.xlsx",
"file_path" :  "/cable_bringup_root/data/uploads/topology/ptp/proton-ptp.xlsx",
"dc_layout_file_path" :  null,
"file_hash" :  "c13187caece919c9aa88d2c1e26404fe5e3d0cd56ea801c4c46a7236e42549fb" ,
"load_time" :  1732376613.3298793
}
```

Get Summary Report

- Description – Gets cluster summary by name.
- Request URL – `GET /cablevalidation/report/summary`
- Request Params
    - cluster: Name of the cluster. If set, the response will return the device health for that specific cluster .
        - Type: string
        - Optional: true
        - Default: default
- Response Content Type – application/json
- Status Codes
    - 200 – OK
    - 400 – BAD REQUEST
    - 404 – NOT FOUND
- Response Example

```
[
   {"syndrome" :  "No Transceiver" , "number_of_occurrences" :  0 , "switches_affected" :  0} ,
   {"syndrome" :  "Link Down, No signal" , "number_of_occurrences" :  0 , "switches_affected" :
0} ,
   {"syndrome" :  "Wrong-neighbor" , "number_of_occurrences" :  0 , "switches_affected" :  0} ,
   ……etc
]
```

Get Circuit Report

- Description – Gets circuits information.
- Request URL – `GET /cablevalidation/report/circuits`
- Request Params
    - cluster: - Name of the cluster. If set, the response will return the circuits for that specific cluster.
        - Type: string
        - Optional: true
        - Default: default
    - node - The name of the node. If set, the response will return the circuits for that specific node .
        - Type: string
        - Optional: true
    - port - The name of the port . If set, the response will return the circuits for that specific port.
        - Type: string
        - Optional: true
    - page - The name of the page. It can be one of the following values: circuit, flap, or flap_hist.
        - Type: string
        - Optional: true
    - healthy - If true, only healthy circuits will be returned.

        - Type: bool
        - Optional: true
    - circuit_id - circuit id. If set, the response will return the circuit with that id .
        - Type: string
        - Optional: true
    - report - syndrome name, if set, the response will return the circuits with that syndrome
- Response Content Type – application/json
- Status Codes
    - 200 – OK
    - 400 – BAD REQUEST
    - 404 – NOT FOUND
- Response Example

```
[
    {
        "circuit_id" : "c83df4d68b40" ,
        "a_endpoint" : {
            "node_type" : "Switch" ,
            "data_hall" : "DH0" ,
            "su_number" : "SU01" ,
            "node" : "sw-hdr-proton01" ,
            "port" : "P4" ,
            "rack" : null ,
            "unit" : null ,
            "actual_node" : null ,
            "actual_port" : null ,
            "port_status" : "up" ,
            "plugged" : true ,
            "advanced_stats" :  {…}
            "remediation_action" :  "Check LLDP is enabled on peer; Verify the peer is fully
provisioned and reachable" ,
            "report" :  "Unknown-neighbor"
         } ,
         "z_endpoint" :  {…. } ,
         "healthy" :  false ,
         "status" :  "Fail"
     }
]
```

Get Device Health Report

- Description – Get device health summary report.
- Request URL – `GET /cablevalidation/report/health`
- Request Params
  - cluster - Name of the cluster. If set, the response will return the device health for that specific cluster .
    - Type: string
    - Optional: true
    - Default: default
  - context - dc, dh, su, rack or node
  - items - The report scope, which typically includes a list of data-halls, scalable-unit/data-hall, racks, or nodes, depending on the selected context. *Do not include this item if the context is "dc"*
- Response Content Type – application/json
- Status Codes
  - 200 – OK

    - 400 – BAD REQUEST

- Response Example

```
[
    {
        "health_summary" :  {
            "Power Supply" :  {
                "failed" :  1 ,
                "active" :  1
            } ,
            "Power Supply Fans" :  {
                "failed" :  1 ,
                "active" :  1
            } ,
            "Fans" :  {
                "failed" :  0 ,
                "active" :  12
            } ,
            "Ports" :  {
                "failed" :  0 ,
                "active" :  2
            }
        } ,
        "data_hall" :  null ,
        "su_number" :  null ,
        "rack" :  "PXX"
    }
]
```

Resources APIs

Get Clusters

- Description – Gets all clusters.
- Request URL – GET /cablevalidation/clusters
- Response Content Type – application/json
- Status Codes
    - 200 – OK
- Response Example

```
[ "default" ]
```

Get Data Halls

- Description – Get a list of all data halls
- Request URL – GET /cablevalidation/resources/data_halls
- Response Content Type – application/json
- Status Codes
    - 200 – OK
- Response Example

```
[ "DH1",   "DH2" ]
```

Get Scalable Units

- Description – Get a list of scalable units.
- Request URL – GET /cablevalidation/resources/scalable_units
- Request Params

  - data_hall: If set, the API will return the scalable units for the specified data hall; otherwise, it will return all scalable units in the system.
    - Type - string
    - Optional - yes
- Response Content Type – application/json
- Status Codes
  - 200 – OK
  - 404 - NOT FOUND
- Response Example

```
["SU1", "SU2"]
```

Get All Racks

- Description – Gets a list of rack names.
- Request URL – `GET /cablevalidation/resources/racks`
- Request Params
  - cluster: Name of the cluster. If set, the response will return the racks in that specific cluster.
    - Type - string
    - Optional - true
    - Default - default
- Response Content Type – application/json
- Status Codes
  - 200 – OK

    - 404 – NOT FOUND
- Response Example

```
["PXX", "PXH"]
```

Get Rack

- Description – Gets a single rack by name.
- Request URL – `GET /cablevalidation/resources/racks/<rack_name>`
- Request Params
    - cluster: Name of the cluster. If set, the response will return the rack in that specific cluster.
        - Type - string
        - Optional - true
        - Default - default
- Response Content Type – application/json
- Status Codes
    - 200 – OK
    - 404 – NOT FOUND
- Response Example

```
{
    "name": "AAA",
    "port_type": "ib",
    "dh": "DH1",
    "su": "SU1",
    "units" :  [ {
        "unit" :  "35" ,
        "nodedesc" :  "swx-ray09 mlx5_0" ,
        "nodetype" :  "Switch" ,
        "system_info" :  {
            "Manufacturer" :  "Nvidia" ,
            "Product Name" :  "Q3400_RA" ,
            "Version" :  "V0-F*Tb-L*GcNaEi-P*PaPa-O*Tb" ,
            "Serial Number" :  "MT2421X00988"
        } ,
        "device_health" :  {
            "Power Supply" :  {
                "Failed" :  11 ,
                "Active" :  1
            } ,
            "Power Supply Fans" :  {
                "Failed" :  1 ,
                "Active" :  1
            } ,
            "Fans" :  {
                "Failed" :  0 ,
                "Active" :  8
            }
        } ,
        "ports" :  [
            {
                "port" :  "sw26p1" ,
                "port_name" :  "" ,
                "syndrome" :  "Wrong-neighbor" ,
                "peer_port" :  "sw10p1" ,
                "peer_node" :  "x-spine-1"
            } ]
            } ]
}
```

Get Nodes

- Description – Gets a single rack by name.
- Request URL – `GET /cablevalidation/resources/nodes`
- Request Params
    - cluster: Name of the cluster. If set, the response will return the nodes in that specific cluster.
        - Type - string
        - Optional - true
        - Default - default
- Response Content Type – application/json
- Status Codes
    - 200 – OK
    - 404 – NOT FOUND
- Response Example

```
[
    {
        "node_desc" : "MQM8700 sw-hdr-proton01" ,
        "node_name" : "sw-hdr-proton01" ,
        "type" : "Switch" ,
        "rack" : "PXX" ,
        "unit" : 27 ,
        "cluster" : "default" ,
        "su_number" : null ,
        "data_hall" : null ,
        "health_summary" : {
            "Power Supply" : {
                "failed" :  1 ,
                "active" :  1
            } ,
            "Power Supply Fans" :  {
                "failed" :  1 ,
                "active" :  1
            } ,
            "Fans" :  {
                "failed" :  0 ,
                "active" :  12
            } ,
            "Ports" :  {
                "failed" :  0 ,
                "active" :  2
            }
        } ,
        "is_managed" :  true ,
        "system_info" :  {
            "Manufacturer" :  "Mellanox Technologies Ltd." ,
            "Product Name" :  "MQM8700" ,
            "Version" :  "AF" ,
            "Serial Number" :  "MT2019T11025"
        } ,
        "asset_tag" :  null ,
        "ip" :  "10.209.44.74" ,
        "last_update_time" :  1754222504
    } ]
```

Get Resource Utilization

- Description – Gets resource utilization.
- Request URL – `GET /cablevalidation/resources/resource_utilization`
- Response Content Type – application/json

- Status Codes
    - 200 – OK
- Response Example

```
{
        "storage" :  {
                "space_total" :  "1.8T" ,
                "space_used" :  "156G" ,
                "space_available" :  "1.5T" ,
                "space_percent_used" :  "10%" ,
                "type" :  "ext4"
        } ,
        "memory" :  {
                "sys_mem_total" :  "125Gi" ,
                "sys_mem_used" :  "10Gi" ,
                "sys_mem_free" :  "71Gi" ,
                "sys_mem_used_percent" :  "8.24%" ,
                "swap_mem_total" :  "15Gi" ,
                "swap_mem_used" :  "0B" ,
                "swap_mem_free" :  "15Gi" ,
                "swap_mem_used_percent" :  "0.0%"
        } ,
        "cpu_util" :  {
                "timestamp" :  1743960805.0907748 ,
                "cpu_util_all_cores" :  "1.43%" ,
                "cpu_util_current_core" :  "0.33%"
        }
}
```

Filters APIs

Filter APIs work on a per-user basis, allowing each user to create their own filters.

Get Page Filters

- Description – Gets all filters in the given page.
- Request URL – `GET /cablevalidation/report/filters/<page>`
- Response Content Type – application/json
- Status Codes
  - 200 – OK
  - 404 – NOT FOUND
- Response Example

```
[ "filter1", "filter2" ]
```

Get Filter

- Description – Gets the filter by name in the specified page.
- Request URL – `GET /cablevalidation/report/filters/<page>/<filter_name>`
- Response Content Type – application/json
- Status Codes
  - 200 – OK
  - 404 – NOT FOUND
  - 500 – INTERNAL SERVER ERROR
- Response Example

```
{
    "filter_name" :  "filter1" ,
    "columns" :  {
        "a_endpoint_cable_pn" :  {
        "type" :  "Contains" ,
        "filter" :  "22"
        }
    }
}
```

Add Filter

- Description – Adds filter to the given page.
- Request URL – P`OST /cablevalidation/report/filters/<page>`
- Response Content Type – application/json
- Status Codes
    - 201 – Created
    - 400 – BAD REQUEST
    - 404 – NOT FOUND
- Request Data Example

```
{
    "filter_name" : "test2" ,
    "columns" : {
    "a_endpoint_cable_pn" : {"type" : "Contains" , "filter" : "22"}
    }
}
```

- Response - 201: Created

Delete Filter

- Description – Deleted the given filter in the specified page.
- Request URL – `DELET /cablevalidation/report/filters/<page>/<filter_name>`
- Response Content Type – application/json
- Status Codes
  - 204 – NO CONTENT
  - 404 – NOT FOUND
- Response – No Content

Commands APIs

Get Supported Commands

- Description – Returns supported commands that could be executed via web.
- Request URL – `GET /cablevalidation/commands`
- Response Content Type – application/json
- Status Codes
  - 200 – OK
- Response Data Example

```
{
"load_topo" :  {
"args" :  {
"dns" :  {
"type" :  "bool" ,
"mandatory" :  false
},
"files" : {
"type" : "list",
"mandatory" : true
},
"cluster" : {
"type" : "str",
"mandatory" : false
}
},
"is_async" : false
},
"load_ip" : {
"args" : {
"files" : {
"type" : "list",
"mandatory" : true
},
"cluster" : {
"type" : "str",
"mandatory" : false
}
},
"is_async" : false
},
…. etc.
}
```

Get Command Help

- Description – Gets command help.
- Request URL – `GET /cablevalidation/commands/<command_name>/help`
- Response Content Type – application/json

- Status Codes
    - 200 – OK
    - 404 – NOT FOUND
- Response Data Example - It depends on the CLI command you need to run. The following is an example of running load_ptp

```
{
    "command" :  "load_topo" ,
    "help" :  [
        "" ,
        "       load_topo filename [dns=true/false] [type=topo/dot] [cluster=<cluster name>] [servers=true/false]" ,
        "       loads a topology file" ,
        "" ,
        "       Parameters" ,
        "       1. dns [optional, boolean(true/false), default is true]:" ,
        "         if dns=true, hostname will be used to communication, otherwise user have to load IP file." ,
        "       2. cluster [optional, string, default is 'default`] cluster name." ,
        "       3. servers [optional, boolean(true/false), default varies]: flag whether to manage agents on servers." ,
        "         default is true for ETH clusters, not supported for IB clusters." ,
        "       4. type [optional, string(topo/dot), default varies]" ,
        "         for IB clusters, the default is `topo`, and for ETH clusters the default is `dot`." ,
        ""
    ]
}
```

Get Commands Status

- Description – Returns the status of commands processing, the status is either 'Idle' of 'Executing <cmd_name>' .

- Request URL – `GET /cablevalidation/commands/status`
- Response Content Type – application/json
- Status Codes
    - 200 – OK
- Response Data Example

```
{
"status" :  "Idle"  or  "Executing <cmd_name>"
}
```

Run Command

- Description – Processes a Bringup CLI command.
- Request URL – `POST /cablevalidation/commands/<command_name>`
- Request Data Example – It depends on the CLI command you need to run. The following is an example of running load_ptp.

```
{
"file" : "/cable_bringup_root/data/uploads/topology/ptp/proton-ptp.xlsx"
}
```

- Response Content Type – application/json
- Status Codes
    - 200 – OK
    - 400 – BAD REQUEST
    - 404 – NOT FOUND

- Response – text message e.g. Command 'load_ptp' completed successfully

Get Command Output

- Description – Gets the output of the given command name.
- Request URL – `GET /cablevalidation/commands/<command_name>/output?timestamp=<timestamp>`
- Response Content Type – application/json
- Status Codes
    - 200 – OK
    - 404 – NOT FOUND
- Response Data Example

```
{
        "command" :  "load_ptp" ,
        "request_ts" :  0 ,
        "last_ts" :  1732376613333429 ,
        "status" :  "Completed" ,
        "content" :  [
                "Warning: Rack unit mismatch for MQM8700 sw-hdr-proton01: 30, 27" ,
                "Loaded 1 switches, 2 hosts, 2 links." ,
                "Loaded IP addresses of 1 switches/hosts!"
        ]
}
```

Users Management APIs

Get All Users

- Description – Gets all users.
- Request URL – `GET /cablevalidation/users`
- Response Content Type – application/json
- Status Codes
    - 200 – OK
    - 500 – INTERNAL SERVER ERROR
- Response Data Example

```
[
    {
        "name" :  "admin",
        "account_type" :  "admin"
    }
]
```

Get User

- Description – Gets the user by the given name.
- Request URL – `GET /cablevalidation/users/<user_name>`
- Response Content Type – application/json
- Status Codes
    - 200 – OK
    - 404 – NOT FOUND
- Response Data Example

```
{
```

```
"name" :  "admin" ,
"account_type" :  "admin"
}
```

Create User

- Description – Creates a user.
- Request URL – `POST /cablevalidation/users`
- Response Content Type – application/json
- Request Data Example

```
{"account_type" : "cabler" , "password" : "user1" , "name" : "user1"}
```

- Status Codes
    - 201 – CREATED
    - 500 – INTERNAL SERVER ERROR
    - 400 – BAD REQUEST
- Response - 201: Created

Update User

- Description – Updates account type or password or both.
- Request URL – `PATCH /cablevalidation/users/<user_name>`
- Response Content Type – application/json
- Request Data Example

```
{"account_type" : "nvidia" , "password" : "test"}
```

- Status Codes
  - 204 – NO CONTENT
  - 500 – INTERNAL SERVER ERROR
  - 400 – BAD REQUEST
- Response - NA

Delete User

- Description – Deletes user by the given name.
- Request URL – `DELETE /cablevalidation/users/<user_name>`
- Response Content Type – application/json
- Status Codes
  - 204 – NO CONTENT
  - 500 – INTERNAL SERVER ERROR
- Response - NA

Get Account Types

- Description – Gets user account types.
- Request URL – `GET /cablevalidation/users/account_types`
- Response Content Type – application/json
- Status Codes
  - 200 – OK

    - 500 – INTERNAL SERVER ERROR
- Response Data Example

```
["nvidia", "admin", "cabler", "developer"]
```

Files Management APIs

Get Files

- Description – Gets the files inside a specified folder, if accessible via web.
- Request URL – `GET /cablevalidation/files_manager/<folder_name>/files?full_path=<true|false>`
- Params
    - `full_path`: if set to true the files will return with the absolute path
        - Type - bool
        - Optional - yes
- Response Content Type – application/json
- Status Codes
    - 200 – OK
    - 400 – BAD REQUEST
    - 404 – NOT FOUND
- Response Data Example

```
[
        "cable_bringup_root/data/uploads/topology/ptp/proton-ptp.xlsx" ,
        "cable_bringup_root/data/uploads/topology/ptp/proton-ptp-2.xlsx"
]
```

Upload File

- Description – Uploads the provided file to the specified folder, only if the folder is accessible via web.
- Request URL – `POST /cablevalidation/files_manager/{folder_name}/files`
- Response Content Type – application/json
- Request Data Example

```
  ------WebKitFormBoundarybxAm2tY7u5Lrhs4q
Content-Disposition: form-data; name="file"; filename="dc_layout.csv"
Content-Type: text/csv
 ------WebKitFormBoundarybxAm2tY7u5Lrhs4q--
```

- Status Codes
    - 201 – CREATED
    - 404 – NOT FOUND
    - 400 – BAD REQUEST
- Response - 201: Created

Delete File

- Description – Deletes the specified file, only if it is accessible via web.

- Request URL – `DELETE /cablevalidation/files_manager/<folder_name>/files/<file_name>`
- Response Content Type – application/json
- Status Codes
  - 204 – NO CONTENT
  - 404 – NOT FOUND
  - 400 – BAD REQUEST
- Response - NA

Exclude Devices APIs

Get Excluded Devices

- Description – Gets the excluded devices.
- Request URL – `GET /cablevalidation/devices/excluded`
- Response Content Type – application/json
- Status Codes
  - 200 – OK

Exclude Devices

- Description – Excludes devices
- Request URL – `POST /cablevalidation/devices/exclude`
- Request Data Example

```
["sw-hdr-proton01", "sw-hdr-proton02"]
```

- Response Content Type – application/json
- Status Codes
    - 200 – OK
    - 400 – BAD REQUEST
- Response - OK

Un-exclude Devices

- Description – Unexcludes devices.
- Request URL – `DELETE /cablevalidation/devices/unexclude`
- Request Data Example

```
["sw-hdr-proton01", "sw-hdr-proton02"]
```

- Response Content Type – application/json
- Status Codes
    - 200 – OK
    - 400 – BAD REQUEST
- Response – OK

Get Version

- Description – Gets the bringup server and cables agent versions.

- Request URL – `GET /cablevalidation/version`
- Response Content Type – application/json
- Status Codes
  - 200 – OK
- Response Data Example

```
{"bringup_version": "1.3.5-28", "agent_version": "1.3.5-28"}
```

Agent Status APIs

Get Agents Summary

- Description – Gets the agent's summary.
- Request URL – `GET /cablevalidation/nodes/agents/summary`
- Request Params
  - cluster: Name of the cluster. If set, the response will return the agent details for that specific cluster.
    - Type - string
    - Optional - true
    - Default - default
- Response Content Type – application/json
- Status Codes
  - 200 – OK
- Response Data Example

```
{
        "total_device_count" :  1 ,
        "correct_agent_version_installed" :  1 ,
        "no_ping" :  0 ,
        "no_json_api" :  0 ,
        "no_agent" :  0 ,
        "wrong_agent_version" :  0 ,
        "no_reports" :  0 ,
        "late_reports" :  0 ,
        "stale_reports" :  0 ,
        "expected_agent" :  "1.6.0-3" ,
        "comment" :  ""
}
```

Get Agents Details

- Description – Gets the agent's summary.
- Request URL – `GET /cablevalidation/nodes/agents/details`
- Request Params
    - cluster: Name of the cluster. If set, the response will return the agent details for that specific cluster.
        - Type - string
        - Optional - True
        - Default - default
    - context: one of the following: **dc, dh, su, rack, or node.**
        - Type - string
        - Optional - False

  - items - The report scope, which typically includes a list of data-halls, scalable-unit/data-hall, racks, or nodes, depending on the selected context. *Do not include this item if the context is "dc".*
    - Type - comma-separated string
    - Optional - False (not included for **DC** context).
    - Examples**:**
      - `context=su&items=DH1/SU1,DH1/SU2`
      - `context=dh&items=DH1,DH2`
      - `context=rack&items=PXH,PXX`
- Response Content Type – application/json
- Status Codes
  - 200 – OK
  - 404 - Not Found
- Response Data Example

```
[
    {
        "hostname" : "sw-hdr-proton01.mtr.labs.mlnx" ,
        "ip" : "10.209.44.74" ,
        "ping" : true ,
        "json_api" : "True" ,
        "agent" : "1.6.0-3" ,
        "node_type" : "Switch" ,
        "comment" : "" ,
        "last_updated" : 1754220100 ,
        "time_since_update" : 3.36 ,
        "dh" : null ,
        "su" : null ,
        "rack" :  "PXX",
        "unit" :  27
    }
]
```

Golden BER Test APIs

Get Tests

- Description – Gets golden BER tests.
- Request URL – `GET /cablevalidation/tests/golden_ber`
- Response Content Type – application/json
- Status Codes
    - 200 – OK
- Response Data Example

```
[
 {
    "timestamp" :  1674151365.5055714,
    "name" :  "Test1",
    "scope" :  "",
    "status" :  "Passed",
    "user" :  "admin",
    "duration_time" :  3,
    "test_time" :  125  ,
    "result" :  null,
    "number_of_failed_circuits" :  0,
    "best_circuit_ber" :  {
        "effective_ber" :  0.0,
        "raw_ber" :  0.0
        },
        "worst_circuit_ber" :  {
            "effective_ber" :  0.0,
            "raw_ber" :  1e-10
        }
    }
]
```

Get Test Circuits

- Description – Gets golden BER test circuits.
- Request URL – `GET /cablevalidation/tests/golden_ber/<test_name>/circuits`
- Response Content Type – application/json
- Status Codes
    - 200 – OK
    - 404 – NOT FOUND
- Response Data Example

```
[
  {
      "a_endpoint" :  {
          "node" :  "sw-hdr-proton01",
          "node_type" :  "Switch",
          "port" :  "P1",
          "rack" :  null,
          "unit" :  null,
          "advanced_stats" :  {
              "metadata" :  {
                  "read_time" :  1727982822.4693458,
                    "file_name" :
"/opt/ufm/cablevalidation/src/cablevalidation/cablesagent/data/amber.csv" ,
                    "file_timestamp" :  1727982822.4331076
                } ,
                "time_since_last_clear" :  0.9 ,
                "power_stats" :  {
                    "rx_power_lane_0" :  1.0 ,
                       . . . . . . .
                    "rx_power_lane_7" :  0.82 ,
                    "rx_power_high_th" :  4.0 ,
                    "rx_power_low_th" :  -5.0 ,
                    "tx_power_lane_0" :  1.27 ,
                        ...
                    "tx_power_lane_7" :  1.33 ,
                    "tx_power_high_th" :  4.0 ,
                    "tx_power_low_th" :  -3.47
                } ,
                "temp_stats" :  {
                    "module_temperature" :  49.0 ,
                    "temperature_low_th" :  -10.0 ,
                    "temperature_high_th" :  80.0
                } ,
                "ber_stats" :  {
                    "effective_ber" :  1.5e-254 ,
                    "raw_ber" :  2e-08 ,
                    "ber_status" :  "Poor" ,
                    "ber_performance" :  "Improved"
                } ,
                "carrier_transition_counter" :  2 ,
                "ib_interface_stats" :  null ,
                "eth_interface_stats" :  {
                    "in_bytes" :  null ,
                    "in_drops" :  null ,
                    "in_errors" :  null ,
                    "out_bytes" :  null ,
                    "out_drops" :  null ,
                    "out_errors" :   null
                } ,
                "flapping_counters" :   {
                    "flap_30_sec" :   0 ,
                    "flap_1_min" :   1 ,
                    "flap_5_min" :   1 ,
                    "flap_1_hr" :   1 ,
                    "flap_12_hrs" :   1 ,
                    "flap_24_hrs" :   1
                }
            }
        } ,
        "z_endpoint" :   {
            . . . . .
        }
    ]
```

Create Test

- Description – Creates a golden BER test.
- Request URL – `POST /cablevalidation/tests/golden_ber`
- Request Data Example
    - context - dc, dh, su, rack or node
    - items - The test scope, which typically includes a list of DH<n>, SU<n>/DH<n>, racks, or nodes, depending on the selected context. *Do not include this item if the context is "dc"*

```
{"name" : "test" , "duration" : 5 , "context" : "rack" , "items" : [ "AAA" ] }
```

- Response Content Type – application/json
- Status Codes

    - 201 – CREATED
    - 409 – CONFLICT
- Response - 201: Created

Stop Test

- Description – Stops the given test.
- Request URL – `DELETE /cablevalidation/tests/golden_ber/{test_name}`
- Response Content Type – application/json
- Status Codes
    - 204 – NO CONTENT
    - 404 – NOT FOUND
- Response – NA

Download The Amber File For A Specific Node

- Description – Download the amber file generated by a specific Golden BER test for a specific node.
- Request URL – `GET /cv_tests/golden_ber/<test_name>/amber_<node_name>.csv.gz`
- Response Content Type – text/html
- Status Codes
    - 200 – OK
    - 404 – NOT FOUND
- Response – File

Download The Amber Files For The Failed Circuits

- Description – Download amber files for the failed circuits generated by the golden ber test.
- Request URL – `GET /cv_tests/golden_ber/<test_name>/<test_name>_failed_circuits_ambe`
- Response Content Type – text/html
- Status Codes
    - 200 – OK
    - 404 – NOT FOUND
- Response – File

Amber Collection Test APIs

Get Tests

- Description – Gets amber tests.
- Request URL – `GET /cablevalidation/tests/amber`
- Response Content Type – application/json
- Status Codes
    - 200 – OK
- Response Data Example

```
[
    {
        "timestamp" :  1674151365.5055714,
        "scope" :  "DH0/SU1",
        "status" :  "Running",
        "name" :  "amber1"
    } ,
    {
        "timestamp" : 1674151365.5055714 ,
        "scope" : "DH0/SU1" ,
        "status" : "Finished" ,
        "name" : "amber2"
    }
]
```

Get Test Details

- Description – Gets amber test details.
- Request URL – `GET /cablevalidation/tests/amber/<test_name>`
- Response Content Type – application/json
- Status Codes
    - 200 – OK
    - 404 – NOT FOUND
- Response Data Example

```

[
    {
        "hostname" : "sw-hdr-proton01" ,
        "ip" : "10.209.44.74" ,
        "node_type" : "Switch" ,
        "status" : "Finished" ,
        "dh" : "DH1" ,
        "su" : "SU2" ,
        "rack" : "AAA" ,
        "unit" : "1"
    }
]
```

Create Test

- Description – Creates amber test.
- Request URL – `POST /cablevalidation/tests/amber`
- Request Data Example
  - context - dc, dh, su, rack or node
  - items - The test scope, which typically includes a list of DH<n>, SU<n>/DH<n>, racks, or nodes, depending on the selected context. *Do not include this item if the context is "dc"*

```
{"name":"test",  "context":"rack",  "items":["AAA"]}
```

- Response Content Type – application/json
- Status Codes
  - 201 – CREATED
  - 409 – CONFLICT
- Response - 201: Created

Download Amber tar File

- Description – Downloads amber tar file.
- Request URL – `GET /cv_tests/amber/<test_name>/<test_name>.tar.gz`
- Response Content Type – text/html
- Status Codes
  - 200 – OK

    - 404 – NOT FOUND
- Response - NA

Advanced Flapping Test APIs

Get Tests

- Description – Gets advanced flapping tests.
- Request URL – `GET /cablevalidation/tests/flapping`
- Response Content Type – application/json
- Status Codes
    - 200 – OK
- Response Data Example

```
[
    {
        "timestamp" :  1674151365.5055714 ,
        "scope" :  "DataCenter" ,
        "status" :  "Finished" ,
        "name" :  "flapping1" ,
        "total_flapping" :  5 ,
        "test_time" :  11 ,
        "duration_time" :  11 ,
        "user" :  "admin"
    } ,
    {
        "timestamp" :  1674151365.5055714 ,
        "scope" :  "DataCenter" ,
        "status" :  "Finished" ,
        "name" :  "flapping2" ,
        "total_flapping" :  5 ,
            "test_time": 11,
            "duration_time": 11,
            "user": "admin"
    }
]
```

Get Test Circuits

- Description – Gets advanced flapping test circuits.
- Request URL – `GET /cablevalidation/tests/flapping/<test_name>/circuits`
- Response Content Type – application/json
- Status Codes
    - 200 – OK
    - 404 – NOT FOUND
- Response Data Example

```
[
    {
        "a_endpoint": {
            "data_hall": "",
            "su_number": "",
            "node": "sw-hdr-proton01",
            "node_type": "Switch",
            "port": "P1",
            "rack": null,
            "unit": null,
            "advanced_stats": {
                "metadata": {
                    "read_time": 1727982822.4693458,
                    "file_name" :
"/opt/ufm/cablevalidation/src/cablevalidation/cablesagent/data/amber.csv" ,
                    "file_timestamp" :  1727982822.4331076
                } ,
                "time_since_last_clear" :  0.9 ,
                "power_stats" :  {
                    "rx_power_lane_0" :  1.0 ,
                      . . .
                    "tx_power_lane_7" :  1.33 ,
                    "tx_power_high_th" :  4.0 ,
                    "tx_power_low_th" :  -3.47
                } ,
                "temp_stats" :  {
                    "module_temperature" :  49.0 ,
                    "temperature_low_th" :  -10.0 ,
                    "temperature_high_th" :  80.0
                } ,
                "ber_stats" :  {
                    "effective_ber" :  1.5e-254 ,
                    "raw_ber" :  2e-08 ,
                    "ber_status" :  "Good" ,
                    "ber_performance" :  "Constant"
                } ,
                "carrier_transition_counter" :  2 ,
                "total_flapping" :  1 ,
                "flapping_counters" :  {
                    "flap_30_sec" :  0 ,
                    "flap_1_min" :  1 ,
                    "flap_5_min" :  1 ,
                    "flap_1_hr" :  1 ,
                    "flap_12_hrs" :  1 ,
                    "flap_24_hrs" :  1 ,
                    "status" :  "Flapped"
                }
            }
        } ,
    "z_endpoint": { ... }}]
```

Create Test

- Description – Creates advanced flapping test.
- Request URL – `POST /cablevalidation/tests/flapping`
- Response Content Type – application/json
- Request Data Example
    - context - dc, dh, su, rack or node
    - items - The test scope, which typically includes a list of DH<n>, SU<n>/DH<n>, racks, or nodes, depending on the selected context. *Do not include this item if the context is "dc"*

```
{"context":"rack", "items":["AAA"], "name":"test", "duration":125}
```

- Status Codes
    - 201 – CREATED
    - 409 – CONFLICT
- Response - 201: Created

Stop Advanced Flapping Test

- Description – Stop the given test.
- Request URL – `DELETE /cablevalidation/tests/flapping/<test_name>`
- Response Content Type – application/json
- Status Codes
    - 204 – NO CONTENT

    - 404 – NOT FOUND
- Response – NA

Cable Validation Controller APIs

Start Bringup

- Description – Starts the bringup.
- Request URL – `POST /cv_controller/start`
- Response Content Type – application/json
- Status Codes
    - 200 – OK
    - 404 – NOT FOUND
- Response - 200: OK

Stop Bringup

- Description – Stops bringup.
- Request URL – `POST /cv_controller/stop`
- Response Content Type – application/json
- Status Codes
    - 200 – OK
    - 404 – NOT FOUND
- Response - 200: OK

Get Devices Credentials

- Description – Gets devices' credentials.
- Request URL – `GET /cablevalidation/creds`
- Response Content Type – application/json
- Status Codes
    - 200 – OK
- Response Data:

```
[
    {
        "node" :  "default" ,
        "user" :  "admin" ,
        "pwd" :  "****" ,
        "type" :  "switch" ,
        "save" :  true
    } ,
    {
        "node" :  "sw-hdr-proton01.mtr.labs.mlnx" ,
        "user" :  "admin" ,
        "pwd" :  "****" ,
        "type" :  "" ,
        "save" :  false
    }
]
```

Help API

- Description – Print supported Rest APIs.
- Request URL – GET /cablevalidation/help
- Response Content Type – text/html

- Status Codes
    - 200 – OK
- Response Data:

```
Bringup server REST API

Command                        Method    Description
/images/{image_name}    GET          For agent use: return the
agent docker image file.
/report/validation              GET         return the recent
cable validation report.
/clusters                            GET          Get all
clusters
/report/summary               GET          Get all clusters
summary or a single cluster summary by name
/report/circuits                   GET          Get circuits
information
```

Amber Simulation APIs

Get Amber Simulation

- Description – Gets amber simulation.
- Request URL – `GET /cablevalidation/simulation/amber`
- Request Params:

Parameter	Description	Type	Optional/Required
node_name	Node name. If set, gets a simulation info for that node	string	Optional

- Response Content Type – application/json

- Status Codes
    - 200 – OK
- Response Data Example

```
[{"node_name": "sw-hdr-proton01", "device": null}]
```

Load Amber Simulation

- Description – Load amber simulation.
- Request URL – `POST /cablevalidation/simulation/amber/load`
- Request Data

Parameter	Description	Type	Optional/Required
node_name	Node name	string	Required
device	Device's name, if the node contains multiple devices	string	Optional
file	The binary file	Binary	Required

- Response Content Type – application/json
- Status Codes
    - 201 – CREATED
    - 400 - Bad Request
    - 409 - CONFLICT
- Response Data Example

```
{
    "status": "success",
    "node_name": "sw-hdr-proton01",
    "saved_file" :  "/tmp/amber_simulation_sw-hdr-proton01_device1_1754227317.csv" ,
    "size_bytes" :  185578
}
```

Stop Amber Simulation

- Description – Load amber simulation.
- Request URL – `DELETE /cablevalidation/simulation/amber/stop`
- Request Parameters:

Parameter	Description	Type	Optional/Required
node_name	Node name	string	Required
device	Device's name, if the node contains multiple devices	string	Optional

- Response Content Type – no content
- Status Codes
    - 204 – NO CONTENT
    - 400 - BAD REQUEST
    - 404 - NOT FOUND

Cable Validation Tool - Prometheus Metrics Endpoint

Overview

The Cable Validation Tool (CVT) now provides a Prometheus-compatible metrics endpoint that exposes real-time cable health and performance data for monitoring and alerting. This endpoint enables integration with modern monitoring stacks like Prometheus, Grafana, and other observability tools.

Features

▯ Key Capabilities

- **Real-time Metrics**: Live cable validation data from network switches and hosts
- **Multi-format Support**: Prometheus, JSON, and CSV output formats
- **Rich Labeling**: Complete network topology context with peer relationships
- **High Performance**: Multi-level caching optimized for frequent scraping
- **Production Ready**: Handles 100K+ ports with memory-adaptive optimizations

▯ Metrics Categories

- **Power Metrics**: RX/TX optical power per lane (up to 8 lanes per port)
- **BER Metrics**: Effective and Raw Bit Error Rates
- **Temperature Metrics**: Module temperature and thresholds
- **Counter Metrics**: Transceiver reinsert/swap events
- **Validation Metrics**: Port validation status with issue descriptions
- **Threshold Metrics**: Power and temperature alarm thresholds
- **Timestamp Metrics**: Data collection and report timestamps

API Endpoints

Base URL

`https://<cvt-server>/cablevalidation/metrics`

Available Endpoints

Endpoint	Format	Content-Type	Description
`/cablevalidation/metrics`	Prometheus	`text/plain`	Standard Prometheus exposition format
`/cablevalidation/metrics/json`	JSON	`application/json`	Structured JSON for programma access
`/cablevalidation/metrics/csv`	CSV	`text/plain`	Comma-separated values for spreadshee import

Authentication

- **No authentication required** for metrics endpoints
- **HTTPS enforced** for security
- **Bypasses session handling** for automated scraping

Sample Output

Prometheus Format

```
# HELP effective_ber Effective Bit Error Rate

# TYPE effective_ber gauge

# HELP validation_status Port validation status with issue descriptions in labels (value = issue count)

# TYPE validation_status gauge
# HELP port_info Port information with status and validation details in labels

# TYPE port_info gauge

# Healthy port with performance metrics

effective_ber{node="ufm-host38",port="enp3s0f0np0",peer_node="r-ufm-sw-
eth01",peer_port="swp2",node_type="Host",su_number="SU1",data_hall="DH1"} 1.5e-254 1759345924622

module_temperature{node="ufm-host38",port="enp3s0f0np0",peer_node="r-ufm-sw-
eth01",peer_port="swp2",node_type="Host",su_number="SU1",data_hall="DH1"} 65.2 1759345924622

validation_status{node="ufm-host38",port="enp3s0f0np0",peer_node="r-ufm-sw-
eth01",peer_port="swp2",node_type="Host",su_number="SU1",data_hall="DH1"} 0 1759345924622

# Unplugged port with validation issue (power/temp metrics excluded due to NA values)

validation_status{node="ufm-host38",port="enp3s0f1np1",peer_node="r-ufm-sw-
eth01",peer_port="swp3",node_type="Host",su_number="SU1",data_hall="DH1",MediaUnplugged="Insert; Reseat
or Replace Cable/Transceiver"} 1 1759345923752

effective_ber{node="ufm-host38",port="enp3s0f1np1",peer_node="r-ufm-sw-
eth01",peer_port="swp3",node_type="Host",su_number="SU1",data_hall="DH1"} 0.0 1759345923752

time_since_last_clear{node="ufm-host38",port="enp3s0f1np1",peer_node="r-ufm-sw-
eth01",peer_port="swp3",node_type="Host",su_number="SU1",data_hall="DH1"} 320035.6 1759345923752

# Port info with detailed status

port_info{node="ufm-host38",port="enp3s0f1np1",peer_node="r-ufm-sw-
eth01",peer_port="swp3",node_type="Host",su_number="SU1",data_hall="DH1",phy_manager_state="Disable",modu
1 1759345923752
```

JSON Format

```
{

  "ufm-host38:enp3s0f0np0": {

    "timestamp": 1757524769.645,

    "port_info": {

      "node_name": "ufm-host38",
      "port_name": "enp3s0f0np0",
      "peer_node_name": "r-ufm-sw-eth01",
      "peer_port_name": "swp2",
      "node_type": "Host",
      "su_number": "SU1",
      "data_hall": "DH1"
    },
    "port_labels": {
      "cable_sn": "ABC123",
      "cable_pn": "DEF456",
      "protocol": "400G"
    },
    "port_stats": {
      "effective_ber": 1.5e-254,
      "module_temperature": 65.2,
      "rx_power_lane_0": -2.5
    },
    "validation_data": {
      "issues_count": 1,
      "last_report_time": 1757524769.645,
      "issues": {
        "WrongNeighbor": "Check cable connection to switch2"
      }
    }
```

```
 }

}
```

Metrics Reference

Power Metrics

Metric	Type	Description	Unit
`rx_power_lane_N`	gauge	RX optical power for lane N (0-7, not all lanes may be present)	dBm
`tx_power_lane_N`	gauge	TX optical power for lane N (0-7, not all lanes may be present)	dBm
`rx_power_high_th`	gauge	RX power high threshold	dBm
`rx_power_low_th`	gauge	RX power low threshold	dBm

BER Metrics

Metric	Type	Description
`effective_ber`	gauge	Effective Bit Error Rate
`raw_ber`	gauge	Raw Bit Error Rate

Temperature Metrics

Metric	Type	Description	Unit
`module_temperature`	gauge	Current module temperature	Celsius
`temperature_high_th`	gauge	Temperature high threshold	Celsius
`temperature_low_th`	gauge	Temperature low threshold	Celsius

Status Metrics

Metric	Type	Description	Values
`port_status`	gauge	Port plugged status	1=Up, 0=Down
`port_oper_status`	gauge	Port operational status	1=Up, 0=Down

Counter Metrics

Metric	Type	Description
`transceiver_reinsert_cnt`	counter	Number of transceiver reinsert events
`transceiver_swap_cnt`	counter	Number of transceiver swap events
`time_since_last_clear`	gauge	Time since last counter clear (seconds)

Validation Metrics

Metric	Type	Description	Special Features
`validation_status`	gauge	Port validation status with issue descriptions	**Value = issue count, descriptions in labels**
`last_report_time`	gauge	Timestamp of last validation report	Unix timestamp

Validation Status Labels

The `validation_status` metric includes dynamic labels for each type of validation issue:

- `WrongNeighbor`: "Check cable connection to correct switch"
- `MediaUnplugged`: "Insert; Reseat or Replace Cable/Transceiver"

- `AnomalousPort`: "Temperature exceeds threshold"
- `FlappingLink`: "Reseat transceiver; Check Fiber"
- `UnknownNeighbor`: "Verify neighbor device connectivity"
- `WrongPort`: "Check port mapping in topology"
- `ExtraCable`: "Remove unexpected cable connection"
- `UnreachableDevice`: "Check device connectivity and power"
- `LinkDown_NoSignal`: "Check physical connection"
- `ErrDisable_Flap`: "Port disabled due to flapping"
- `AdminDown`: "Port administratively disabled"
- `ErrDisable_Rx`: "RX error disable condition"
- `NegotiationFail`: "Check autonegotiation settings"
- `NicNameMismatch`: "Verify NIC provisioning"
- `ModulePnMismatch`: "Replace with compatible module"

Note: Commas in descriptions are automatically converted to semicolons to maintain Prometheus label format compatibility.

Examples:

# Port with validation issues (unplugged cable)

validation_status{node="ufm-host38",port="enp3s0f1np1",peer_node="r-ufm-sw-eth01",peer_port="swp3",node_type="Host",su_number="SU1",data_hall="DH1",MediaUnplugged="Insert; Reseat or Replace Cable/Transceiver"} 1

# Port without issues (healthy connection)

validation_status{node="ufm-host38",port="enp3s0f0np0",peer_node="r-ufm-sw-

```
eth01",peer_port="swp2",node_type="Host",su_number="SU1",data_hall="DH1"} 0

# Port with multiple validation issues

validation_status{node="switch1",port="1/1",WrongNeighbor="Check cable connection to
switch2",AnomalousPort="Temperature exceeds threshold"} 2
```

Labels

Topology Labels (All Metrics)

- `node`: Switch or host name
- `port`: Port identifier
- `peer_node`: Connected peer node name
- `peer_port`: Connected peer port identifier
- `node_type`: Node type (Switch, Host, etc.)
- `su_number`: Scalable Unit identifier
- `data_hall`: Data hall location

Cable Labels (Status Metrics Only)

- `cable_sn`: Cable serial number
- `cable_pn`: Cable part number
- `protocol`: Cable protocol (400G, InfiniBand, etc.)
- `port_status`: Port status (Up, Down, etc.)
- `plugged`: Module plugged status

Performance Characteristics

Update Frequency

- **Agent Data**: Updated every 10 minutes (configurable)
- **Metrics Cache**: Invalidated on data changes
- **Prometheus Scraping**: Recommended 15-30 second intervals

Performance Metrics

Deployment Size	Response Time	Memory Usage	Caching Strategy
< 10K ports	< 50ms	~20MB	Full caching enabled
10K-50K ports	< 200ms	~100MB	Collection cache only
50K+ ports	< 500ms	~200MB	No caching, real-time generation

Optimization Features

- **Multi-level Caching**: Port, collection, and label caching
- **Memory Adaptive**: Automatically adjusts for large deployments
- **Smart Change Detection**: Only updates when cable/module data changes
- **Zero Value Handling**: Includes all values for complete visibility

Troubleshooting

Common Issues

1. No Metrics Data

Symptoms: Empty response or no metrics **Causes**:

- CVT service not running
- No topology loaded
- No advanced stats collection

Solutions:

```
# Check service status

# Check topology loading

# Check agent connectivity
```

2. Missing Port Data

Symptoms: Some ports not appearing in metrics **Causes**:

- Port not in loaded topology
- Agent not deployed on switch
- Advanced stats not collected

Solutions:

- Verify topology includes all expected ports
- Deploy agents on missing switches
- Check agent connectivity and data collection

3. Stale Timestamps

Symptoms: Old timestamps in metrics **Causes**:

- Agent not sending updates

- Network connectivity issues

Solutions:

- Check agent logs for errors
- Verify network connectivity to switches
- Restart agents if necessary

4. Missing Validation Data

Symptoms: validation_status metrics missing or always 0 **Causes**:

- Validation reports not being generated
- Agent data filtering (switch not in topology)
- Report processing errors

Solutions:

- Verify validation is started on agents
- Check switch IP exists in topology
- Review agent and collector logs for errors

5. Inconsistent Issue Counts

Symptoms: validation_status count doesn't match expected issues **Causes**:

- Issues filtered by port
- Report data synchronization issues
- Processing errors

Solutions:

- Check that report data includes port-specific issues

- Verify advanced stats and reports arrive together
- Review validation report structure

Performance Tuning

Environment Variables

TBD: not supported yet.

```
# Adjust caching thresholds
PROMETHEUS_MAX_CACHED_PORTS=10000
# Disable detailed metrics for very large deployments
PROMETHEUS_ENABLE_DETAILED_METRICS=false
# Adjust cache TTL
PROMETHEUS_CACHE_TTL=60
```

Memory Optimization

For deployments > 50K ports:

- Collection-level caching automatically disabled
- Port-level caching automatically disabled
- Real-time generation used (acceptable 200-500ms response time)

Security Considerations

Access Control

- **HTTPS Required**: All access must be over HTTPS
- **No Authentication**: Designed for automated monitoring tools
- **Network Restrictions**: Consider IP-based access control

Sensitive Data

- **Network Topology**: Metrics expose network structure
- **Cable Information**: Serial numbers and part numbers included
- **Performance Data**: Could reveal network capacity information

Recommended Security

```
<Location /cablevalidation/metrics>
    Use BringupProxy
    SSLRequireSSL

    # Restrict to monitoring networks
    <RequireAll>
        Require ip 10.0.0.0/8      # Internal networks
        Require ip 172.16.0.0/12   # Container networks
        Require ip 192.168.0.0/16  # Private networks
    </RequireAll>
</Location>
```

Integration Examples

Python Client Example

```
import requests
import json
# Get metrics in different formats
def get_cvt_metrics(server: str, port: int, response_format='prometheus'):
    endpoints = {
        'prometheus': '/cablevalidation/metrics',
        'json': '/cablevalidation/metrics/json',
        'csv': '/cablevalidation/metrics/csv'
    }
    url = f"https://{server}:{port}{endpoints[format]}"
    response = requests.get(url, verify=False)

    if format == 'json':
        return response.json()
    return response.text
# Usage
metrics = get_cvt_metrics('cvt-server.example.com', 'json')
for port_key, port_data in metrics.items():
    if port_data['port_stats']['effective_ber'] > 1e-12:
        print(f"High BER on {port_key}: {port_data['port_stats']['effective_ber']}")
```

Validation Monitoring Examples

```
# Find all ports with validation issues
validation_status > 0
# Count issues by syndrome type
sum by (node) (validation_status{WrongNeighbor!=""})
sum by (node) (validation_status{MediaUnplugged!=""})
sum by (node) (validation_status{LinkDown_NoSignal!=""})
# Find specific issue types
validation_status{MediaUnplugged!=""} > 0  # Unplugged cables
validation_status{AdminDown!=""} > 0       # Administratively disabled ports
validation_status{ModulePnMismatch!=""} > 0 # Hardware compatibility issues
# Ports with multiple issue types
validation_status{WrongNeighbor!="",AnomalousPort!=""}
# Correlation with performance metrics
(validation_status > 0) and (effective_ber > 1e-12)
# Port status correlation
validation_status{MediaUnplugged!=""} and on() port_info{module_oper_status="unplugged"}
```

Architecture

Data Flow

```
Network Agents   CVT Collector   Advanced Stats + Report Data
Prometheus Collector   Metrics Endpoint<p></p>
                                                  <p></p>  (10 min)
(Real-time)        (Synchronized)              (Multi-level Cache)
(GET request)
```

Enhanced Data Processing

1. **Agent Data Validation**: Switch IP validated against topology before processing
2. **Synchronized Processing**: Advanced stats and validation reports processed together
3. **Optimized Issue Processing**: Report data pre-processed to group issues by port (O(n+m) complexity)
4. **Independent Validation Cache**: PortValidationStatus class with hash-based change detection
5. **Robust Syndrome Handling**: Automatic fallback for unknown syndromes with developer warnings
6. **Smart Data Quality**: NA values properly excluded, counters preserve semantics

Caching Strategy

1. **Port-level Cache**: Individual port metrics cached until data changes
2. **Collection-level Cache**: Aggregated output cached for fast retrieval
3. **Label Cache**: Stable topology/cable labels cached separately
4. **Validation Cache**: Independent cache for validation status with hash-based change detection
5. **Metadata Cache**: Static TYPE/HELP comments cached permanently

Performance Optimizations

- **Push-based Updates**: Metrics updated when advanced stats arrive
- **Smart Change Detection**: Only cable/module changes invalidate caches
- **Memory Adaptive**: Caching disabled automatically for large deployments
- **String Manipulation**: Efficient JSON aggregation using string operations
- **Validation Processing**: O(n+m) complexity with report preprocessing

- **Hash-based Cache**: Validation cache only invalidated when issue content changes
- **Agent Data Filtering**: Invalid switch IPs filtered early to prevent unnecessary processing

Monitoring Best Practices

Prometheus Configuration

- **Scrape Interval**: 15-30 seconds (matches CVT data update frequency)
- **Timeout**: 10 seconds (allows for cache generation)
- **Retention**: Configure based on historical analysis needs

Alerting Guidelines

- **BER Thresholds**: Alert when effective_ber > 1e-12
- **Temperature Limits**: Alert when module_temperature approaches temperature_high_th
- **Validation Issues**: Alert when validation_status > 0
- **Critical Issues**: Alert on specific syndromes (MediaUnplugged, UnreachableDevice, ModulePnMismatch)
- **Infrastructure Issues**: Alert on LinkDown_NoSignal, ErrDisable conditions
- **Counter Anomalies**: Alert on rapid increases in transceiver_reinsert_cnt

Sample Alerting Rules

```
# Validation issues alert
- alert: PortValidationIssues
  expr: validation_status > 0
  labels:
    severity: warning
  annotations:
    summary: "Port {{ $labels.node }}:{{ $labels.port }} has validation issues"
    description: "{{ $value }} validation issues detected"
# Critical validation issues
- alert: CriticalPortIssues
  expr: validation_status{MediaUnplugged!=""} > 0 or validation_status{UnreachableDevice!=""} > 0
  labels:
    severity: critical
  annotations:
    summary: "Critical issues on {{ $labels.node }}:{{ $labels.port }}"
    description: "{{ if $labels.MediaUnplugged }}Cable unplugged: {{ $labels.MediaUnplugged }}{{ end }}{{ if $labels.UnreachableDevice }}Device unreachable: {{ $labels.UnreachableDevice }}{{ end }}"
# Infrastructure issues
- alert: InfrastructureIssues
  expr: validation_status{LinkDown_NoSignal!=""} > 0 or validation_status{ModulePnMismatch!=""} > 0
  labels:
    severity: warning
  annotations:
    summary: "Infrastructure issue on {{ $labels.node }}:{{ $labels.port }}"
# Administrative issues
- alert: AdminIssues
  expr: validation_status{AdminDown!=""} > 0 or validation_status{NicNameMismatch!=""} > 0
  labels:
```

```
  severity: info
 annotations:
  summary: "Administrative issue on {{ $labels.node }}:{{ $labels.port }}"
```

Version Information

Release Notes

- **Version**: 1.1.0
- **Release Date**: October 2025
- **Compatibility**: CVT 1.7.0 and later
- **Dependencies**: Requires advanced stats collection enabled

New in Version 1.1.0

- ▯ **Validation Metrics Integration**: Port validation status with actionable issue descriptions
- ▯ **Synchronized Data Processing**: Advanced stats and validation reports processed together
- ▯ **Performance Optimizations**: O(n+m) validation processing, hash-based change detection
- ▯ **Enhanced Security**: Agent data validation prevents processing from unknown switches
- ▯ **Improved Data Quality**: None-based initialization for gauges, proper counter semantics
- ▯ **Better Caching**: Independent validation cache with content-based invalidation
- ▯ **Comprehensive Syndrome Coverage**: 15+ validation issue types with fallback handling

- ▯ **Real-world Validation**: Successfully tested with production data and unplugged ports

API Stability

- **Metric Names**: Stable (no breaking changes planned)
- **Label Names**: Stable (additions possible, no removals)
- **Output Format**: Prometheus standard compliance maintained
- **Endpoint URLs**: Stable API contract

Data Quality Improvements

Enhanced Counter Semantics

- **Gauges default to None**: Missing sensor data excluded instead of showing false zeros
- **Counters preserve values**: No unexpected resets when data temporarily unavailable
- **Proper NA handling**: Invalid data marked as NA and excluded from metrics
- **Temperature accuracy**: Fixed zero temperature issue by using actual amber timestamps

Validation Integration Benefits

- **Synchronized processing**: Performance metrics and validation issues always in sync
- **Rich context**: Issue descriptions provide actionable corrective actions
- **Efficient processing**: O(n+m) complexity prevents performance degradation
- **Smart caching**: Validation cache independent of performance metrics cache

Support

Troubleshooting

1. **Check CVT Service**: Ensure Cable Validation service is running
2. **Verify Topology**: Confirm network topology is loaded
3. **Agent Status**: Check that agents are deployed and collecting data
4. **Network Connectivity**: Verify switch/host accessibility

Performance Monitoring

```
# Check metrics endpoint response time

time curl -k https://cvt-server/cablevalidation/metrics > /dev/null
```

Contact Information

- **Development Team**: Cable Validation Engineering
- **Documentation**: [Internal Wiki Link]
- **Support**: [Support Channel/Email]

This endpoint provides comprehensive cable validation metrics for modern monitoring and observability workflows, enabling proactive network health management and automated alerting.

Document Revision History

Version	Date	Description
1.7.1	Nov 5, 2025	Updated: • Introduction • P2P File - Updated the "Unified Topology Format" • Deploying the Module • Bringup CLI • Check if Cable Agent is Running • Collecting Amber File • Circuits Overview • Rack View • System Admin Added: • CVT as a Service • CVT Configuration • High Availability (HA) Mode Support for CVT Plugin • Agent Deployment and SSH Configuration • Cluster Sizing Guide • Open Agent Port • Resource Filter • Agent Status APIs
1.6.0	Aug 7, 2025	Documentation History in UFM Cable Validation Tool v1.6.0
1.5.0	May 5, 2025	Documentation History in UFM Cable Validation Tool v1.5.0
1.4.0	Feb 10, 2025	Documentation History in UFM Cable Validation Tool v1.4.0
1.3.1	Aug 12, 2024	Documentation History in UFM Cable Validation Tool v1.3.1
1.2.0	Nov 5, 2023	Documentation History in UFM Cable Validation Tool v1.2.0
1.1.0	Aug 10, 2023	Documentation History in UFM Cable Validation Tool v1.1.0
1.0.0	May 8, 2023	First release

Notice

This document is provided for information purposes only and shall not be regarded as a warranty of a certain functionality, condition, or quality of a product. NVIDIA Corporation ("NVIDIA") makes no representations or warranties, expressed or implied, as to the accuracy or completeness of the information contained in this document and assumes no responsibility for any errors contained herein. NVIDIA shall have no liability for the consequences or use of such information or for any infringement of patents or other rights of third parties that may result from its use. This document is not a commitment to develop, release, or deliver any Material (defined below), code, or functionality.

NVIDIA reserves the right to make corrections, modifications, enhancements, improvements, and any other changes to this document, at any time without notice.

Customer should obtain the latest relevant information before placing orders and should verify that such information is current and complete.

NVIDIA products are sold subject to the NVIDIA standard terms and conditions of sale supplied at the time of order acknowledgement, unless otherwise agreed in an individual sales agreement signed by authorized representatives of NVIDIA and customer ("Terms of Sale"). NVIDIA hereby expressly objects to applying any customer general terms and conditions with regards to the purchase of the NVIDIA product referenced in this document. No contractual obligations are formed either directly or indirectly by this document.

NVIDIA products are not designed, authorized, or warranted to be suitable for use in medical, military, aircraft, space, or life support equipment, nor in applications where failure or malfunction of the NVIDIA product can reasonably be expected to result in personal injury, death, or property or environmental damage. NVIDIA accepts no liability for inclusion and/or use of NVIDIA products in such equipment or applications and therefore such inclusion and/or use is at customer's own risk.

NVIDIA makes no representation or warranty that products based on this document will be suitable for any specified use. Testing of all parameters of each product is not necessarily performed by NVIDIA. It is customer's sole responsibility to evaluate and determine the applicability of any information contained in this document, ensure the product is suitable and fit for the application planned by customer, and perform the necessary testing for the application in order to avoid a default of the application or the product. Weaknesses in customer's product designs may affect the quality and reliability of the NVIDIA product and may result in additional or different conditions and/or requirements beyond those contained in this document. NVIDIA accepts no liability related to any default, damage, costs, or problem which may be based on or attributable to: (i) the use of the NVIDIA product in any manner that is contrary to this document or (ii) customer product designs.

No license, either expressed or implied, is granted under any NVIDIA patent right, copyright, or other NVIDIA intellectual property right under this document. Information published by NVIDIA regarding third-party products or services does not constitute a license from NVIDIA to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property rights of the third party, or a license from NVIDIA under the patents or other intellectual property rights of NVIDIA.

Reproduction of information in this document is permissible only if approved in advance by NVIDIA in writing, reproduced without alteration and in full compliance with all applicable export laws and regulations, and accompanied by all associated conditions, limitations, and notices.

THIS DOCUMENT AND ALL NVIDIA DESIGN SPECIFICATIONS, REFERENCE BOARDS, FILES, DRAWINGS, DIAGNOSTICS, LISTS, AND OTHER DOCUMENTS (TOGETHER AND SEPARATELY, "MATERIALS") ARE BEING PROVIDED "AS IS." NVIDIA MAKES NO WARRANTIES, EXPRESSED, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE MATERIALS, AND EXPRESSLY DISCLAIMS ALL IMPLIED WARRANTIES OF NONINFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE. TO THE EXTENT NOT PROHIBITED BY LAW, IN NO EVENT WILL NVIDIA BE LIABLE FOR ANY DAMAGES, INCLUDING WITHOUT LIMITATION ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, PUNITIVE, OR CONSEQUENTIAL DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF

Trademarks