



Model2 Plugin

Table of contents

Overview

Deployment

Supported APIs

Ports API

Links API

SHARP Reservation APIs

Overview

The Model2 plugin is a component that operates alongside the UFM model, offering a more scalable and efficient method for managing the fabric's inventory and topology. It uses a Redis database to store data received from the SM client consumer and the ibdiagnet collector. Additionally, it provides REST APIs for querying information about ports, systems, links, switches, and SHARP reservations. The Model2 plugin supports persistent inventory, which tracks all devices, ports, and links encountered by UFM, even if they are later disabled or removed. While designed to be backward compatible with the UFM model, there are minor differences in API parameters and the handling of generic node names.

Deployment

- Download the Model2 image from the [NVIDIA License Portal \(NLP\)](#).
- Load the downloaded image onto the UFM server. This can be done either by using the UFM GUI by navigating to the Settings -> Plugins Management tab, or by loading the image via the following instructions:
 - Log in to the [UFM server terminal](#).
 - Run:
 - `docker load -i <path_to_image>`
 - After successfully loading the plugin image, it should appear in the plugin management table within the UFM GUI. To start the plugin, simply right-click on it in the table.

Supported APIs

The following APIs are supported:

Ports API

Retrieves all the ports in the cluster or filters them by the system name. The following attributes are retrieved:

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

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

- GET /resources/ports/?cable_info=true
 - part_number: The part number of the port.
 - serial_number: The serial number of the port.
 - revision: The revision of the port.
 - identifier: The identifier of the port.

- length: The length of the port.
- technology: The technology of the port.
- fw_version: The firmware version of the port.

Links API

Retrieves all the links in the cluster. The following attributes are retrieved:

- GET /resources/links
 - source_guid: The GUID of the source port.
 - source_port: The source port.
 - destination_guid: The GUID of the destination port.
 - destination_port: The destination port.
 - source_port_dname: The display name of the source port.
 - source_port_name: The name of the source port.
 - destination_port_dname: The display name of the destination port.
 - destination_port_name: The name of the destination port.
 - width: The width of the link.
 - severity: The severity of the link.
 - source_port_node_description: The description of the node connected to the source port.
 - destination_port_node_description: The description of the node connected to the destination port.
 - name: The name of the link.

- active: The status of the link.

This API retrieves information about the systems in the cluster. It can retrieve all the systems or filter by the system name. The following attributes are retrieved:

- GET /resources/systems
- GET /resources/systems/<system_name>
- GET /resources/switches
 - name: The display name of the system.
 - system_name: The name of the system.
 - guid: The GUID of the system.
 - sys_guid: The system GUID.
 - ports: A list of ports associated with the system.
 - hostname: The hostname of the system.
 - vendor: The vendor of the system.
 - temperature: The temperature of the system.
 - fw_version: The firmware version of the system.
 - technology: The technology used by the system.
 - level: The level of the system.
 - uptime: The uptime of the system.
 - sw_version: The software version of the system.
 - system_type: The type of the system.
 - description: The description of the system.
 - psid: The PSID of the system.

- active: The status of the system.
- state: The state of the system.
- is_managed: Indicates whether the system is managed.

SHARP Reservation APIs

The following SHARP Reservation APIs are supported and their input and description can be found in the UFM REST API User Manual.

- POST /app/sharp/reservation
- DELETE /app/sharp/reservation
- GET /app/sharp/jobs
- GET /app/sharp/jobs/<job_id>

© Copyright 2024, NVIDIA. PDF Generated on 06/06/2024