Examine the DNA Center Inventory Service and Common Issues

Contents

Introduction

This document describes the Cisco DNA Center Inventory service basic concepts and common issues found in production.

Components Used

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, ensure that you understand the potential impact of any command.

Inventory Service Details

The Cisco DNA Center Inventory service is based in a **Kubernetes (K8s) Pod** which you can find running in the **namespace "fusion"** with name "**apic-em-inventory-manager-service-<id>**" as a Deployment environment type.

Inside of the K8s pod, you can find a **Docker container** called "**apic-em-inventory-manager-service**".

The "**apic-em-inventory-manager-service**" pod main tasks are: Device discovery and device lifecycle management.

This ensures device data is available in Postgres SQL (database used by fusion services).

The **"fusion" namespace (Appstack)** also known as the Network Controller Platform (NCP), provides the Service Provisioning Framework (SPF) services for all network automation requirements.

These include discovery, inventory, topology, policy, Software Image Management (SWIM), Configuration Archive, Network Programmer, Sites, grouping, telemetry, Tesseract integration, template programmer, maps, IPAM, Sensors, Orchestration/Workflow/Scheduling, ISE integration, and similar.

The **inventory pod status** can be checked by running the command:

```
$ magctl appstack status | grep inventory
```

The **inventory service status** can be checked with the command:

```
$ magctl service status <inventory_pod_name>
```

The **inventory service logs** can be checked with the command:

```
$ magctl service logs -r <inventory_pod_name>
```

> **Note**: The inventory service can also consist in two running pods, so you need to specify a single pod in the commands by using the complete inventory pod name, including the pod id.

In this document we can focus in the Inventory device Manageability and Last Syncing status to review the common issues:

Manageability Status

- **Managed with green tick icon:** Device is reachable and is fully managed.
- **Managed with orange error icon:** Device is managed with some error such as unreachable, authentication failure, missing Netconf ports, internal error, and so on. You can hover the cursor over the error message to view more details about the error and the impacted applications.
- **Unmanaged:** Device cannot be reached and no inventory information was collected due to device connectivity issues.

Last Sync Status

- **Managed:** Device is in a fully managed state.
- **Partial Collection Failure:** Device is in a partial collected state and not all the inventory information has been collected. Hover the cursor over theInformation (i)icon to display additional information about the failure.
- **Unreachable:** Device cannot be reached and no inventory information was collected due to device connectivity issues. This condition occurs when periodic collection takes place.
- **Wrong Credentials:** If device credentials are changed after adding the device to the inventory, this condition is noted.
- **In Progress:** Inventory collection is occurring.

> **Note**: For more information about Inventory functions in Cisco DNA Center please review the official guide for version 2.3.5.x: Manage Your Inventory

---

Problems

Internal Error

The Cisco DNA Center Inventory page can display a warning message in the Manageability status for devices with some kind of conflict preventing the data collection:

"**Internal Error: NCIM12024: All information from the device could not be collected successfully or the inventory collection for this device has not yet started. It can be a temporary problem that can resolve automatically. Resync the device, if that does not resolve the problem, please contact Cisco TAC.**"

If the error does not resolves automatically or after a device resync, we can start with initial troubleshooting. That error can be due to multiple reasons, but here, we list just some of the most common:

- Incorrect device credentials for SNMP, SSH and Netconf.
- Network connectivity issues related to SNMP, SSH and Netconf.
- Netconf configuration issues in the device causing Netconf not to work correctly.
- Trigger a device resync while a device syncing is already ongoing.
- Multiple traps been received from the device causing multiple resync triggers in a short period of time.
- Backend issues with inventory database entries in multiple tables related to the device.

> **Tip**: Removing the network device and re-discovering it using the correct CLI, SNMP and NETCONF credentials can help to remove stale database entries that could be causing the Internal Error.

> **Tip**: Reviewing the Inventory service logs and filtering by device IP or Hostname can be helpful to identify the Internal Error root cause.

Device Credentials

In order to review device credentials, navigate to the Cisco DNA Center **Menu -> Provision -> Inventory -> Select Device -> Actions -> Inventory -> Edit Device** and click on **"Validate"** and confirm that the mandatory credentials (CLI and SNMP) are passing the validation with a green check (including netconf if it applies).

If validation fails, please review that the **username and password** that Cisco DNA Center is using to manage the network device are valid directly in the device command line.

If they are locally configured or If they are configured in an AAA server (TACACS or RADIUS) please validate that the username and password are correctly configured in the AAA server.

Also check if the username privilege requires to have the **"Enable"** password setup in the Device Credentials Settings in Cisco DNA Center Inventory.

Errors in CLI credentials can cause a manageability error message in Inventory: **CLI Authentication Failure**.

Netconf

Netconf is a protocol to remotely manage a compatible network device via Remote Procedure Calls (RPC).

Cisco DNA Center uses Netconf capabilities to push or remove configuration on network devices to enable features like monitoring via Assurance.

Cisco DNA Center Inventory can also validate that the **Netconf requirements** are correct, which includes:

- Netconf default **port 830** to be open and functional in the network.
- User with **privilege 15** with SSH access to the network device (locally or AAA configured).
- Enable Netconf in the network device:

```
<#root>

(config)#

netconf-yang
```

- If aaa new-model is enabled, then you also need to configure the AAA default settings requirements:

```
<#root>

(config)#

aaa authorization exec default <local or radius/tacacs group>

(config)#

aaa authentication login default <local or radius/tacacs group>
```

Errors in Netconf credentials can cause a manageability error message in Inventory: **Netconf Connection Failure**.

Network Checks

We can also validate the network connectivity and protocols settings like **SNMP settings** depending in the version.

For example we can double check community, user, group, engineID, authentication and encryption settings and so on depending on SNMP version.

We also can review SSH and SNMP connectivity using **ping** and **traceroute** commands in device command line and ports for SSH (22) and SNMP (161 and 162) in firewall, proxy or Access Lists.

From Cisco DNA Center, maglev CLI we use the **ip route** commands to validate connectivity to the network device.

SNMP walk can also be used to troubleshoot.

Errors in SNMP credentials can cause a manageability error message in Inventory: **SNMP Authentication Failure** or **Device Unreachable**.

Database tables

As end user, you can use the Cisco DNA Center GUI with **Grafana to execute SQL** queries so you do not need access to the Postgres shell via maglev CLI.

> **Tip**: If you want to learn how to use Grafana please review the official guide: Execute Postgres Queries in Cisco DNA Center GUI

Some postgres database tables to review when having issues with network devices in Inventory are:

- networkdevice
- managedelementinterface
- networkelement
- networkresource
- deviceif
- ipaddress

> **Warning**: Only Cisco TAC is allowed to run show queries in the Postgres Shell and only BU/DE teams are allowed to make modifications to DB tables.

> **Note**: Database issues can also cause the internal error message for devices which can prevent data collection and device provisioning.

> **Tip**: You can review the Postgres logs using Kibana in the Cisco DNA Center System 360 page and look for Constraint Violations when Inventory service is trying to save or update entries in Postgres database tables.

Sync Loop and Traps

Cisco DNA Center is design to execute a device Resycn each time it receives a trap from the device after a major change is performed in the device itself in order to keep the Cisco DNA Center Inventory updated. Sometimes Cisco DNA Center Inventory page keeps your network devices in **"Syncing"** status in the **Manageability** section for a long period of time or for ever.

> **Note**: These kind of sync loops due to massive traps can cause Cisco DNA Center to authenticate multiple times in a short period of time to devices who are sending the traps due to changes detected.

API to Force Device Sync

If your network device keeps in Syncing status for too long, even days, first review the basics checks for reachability and connectivity. Then force the device resync via API call:

1.- Open the Cisco DNA Center **maglev CLI** session.

2.- Get the Cisco DNA Center authentication token via API:

<#root>

```
curl -s -X POST -u admin  https://kong-frontend.maglev-system.svc.cluster.local/api/system/v1/identitymg
```

3.- Use the token from previous step to run the API to Force the device Sync:

<#root>

curl -X PUT -H "X-AUTH-TOKEN:<auth_token>" -H "content-type: application/json" -d '<device_uuid>' https://<cluster-ip>/api/v1/network-device/sync

4.- You can see the device in Syncing once again but this time with a Force Sync option via API.

> **Tip**: You can get the device uuid from the Browser URL (deviceid or id) from the Cisco DNA Center Inventory Device Details page or Device View 360 page.

> **Note**: For more information about APIs in Cisco DNA Center please review the Cisco DevNet API Guide

---

Review Traps

If issue persists after forcing the syncing task in the device, we can review if the Cisco DNA Center **"event-service"** is receiving too many traps and review which type of traps by reading the event-service logs:

1.- Before we read the logs we can just check the total traps with the command:

<#root>

$ echo;echo;eventsId=$(docker ps | awk '/k8s_apic-em-event/ {print $1}'); docker cp $eventsId:/opt/CSCOlumos/logs/ /tmp/;for ip in $(awk -F\: '/ipAddres

2.- Then we attach to the event-service container:

<#root>

```
$ magctl service attach -D event-service
```

3.- Once you get inside the event-service container, change directory to the logs folder:

<#root>

$ cd /opt/CSCOlumos/logs/

4.- If you review the files inside the directory you can see some logs files which their name starts with "ncs".

Example:

<#root>

root@apic-em-event-service-586df7d4b8-f9c74:/opt/CSCOlumos/logs#

ls -l

```
total 90852
drwxr-xr-x 1 maglev maglev     4096 May  9 21:33 ./
drwxr-xr-x 1 maglev maglev     4096 Apr 29 17:56 ../

-rw-r--r-- 1 root   root    2937478 May  9 21:37 ncs-0-0.log
-rw-r--r-- 1 root   root          0 Apr 29 23:59 ncs-0-0.log.lck
-rw-r--r-- 1 root   root   10002771 May  9 21:33 ncs-1-0.log
-rw-r--r-- 1 root   root   10001432 May  9 21:16 ncs-2-0.log
-rw-r--r-- 1 root   root   10005631 May  9 21:01 ncs-3-0.log
-rw-r--r-- 1 root   root   10000445 May  9 20:47 ncs-4-0.log
-rw-r--r-- 1 root   root   10000507 May  9 20:33 ncs-5-0.log
-rw-r--r-- 1 root   root   10003091 May  9 20:21 ncs-6-0.log
-rw-r--r-- 1 root   root   10001058 May  9 20:06 ncs-7-0.log
-rw-r--r-- 1 root   root   10001064 May  9 19:53 ncs-8-0.log
-rw-r--r-- 1 root   root   10000572 May  9 19:39 ncs-9-0.log

-rw-r--r-- 1 root   root        424 Apr 30 00:01 nms_launchout.log
-rw-r--r-- 1 root   root        104 Apr 30 00:01 serverStatus.log
```

5.- Those "ncs" files are the one we need to analyze which type of traps we are receiving and how many. We can review the logs files filtering them by device hostname or the keyword "trapType":

<#root>

root@apic-em-event-service-586df7d4b8-f9c74:/opt/CSCOlumos/logs#

grep trapType ncs*.log

root@apic-em-event-service-586df7d4b8-f9c74:/opt/CSCOlumos/logs#

grep <hostname> ncs*.log

There are too many type of traps, some of them can trigger the device resync and if they come too much frequently they can cause the Sync loop.

By analyzing the traps we can identify the root cause and make traps to stop, for example an AP in a Rebooting Cycle.

You can save the traps output into a file and share them with escalation team if needed.

Service Crashing Status

If you suspect that the inventory pod is crashing due to an odd behavior in the Cisco DNA Center Inventory page while managing network devices, then you can validate the pod status first:

<#root>

$ magctl appstack status | grep inventory

$ magctl service status <inventory_pod_name>

Reviewing the output of the pod status, if you see a high number of restarts or an error status, then you can attach to the inventory container and collect the heapdump file which can have the data that can help escalation team to analyze and define the root cause of the crashing state:

<#root>

$ magctl service attach -D <inventory_pod_name>

root@apic-em-inventory-manager-service-76f7f8d7f5-427m5:/#

ll /opt/maglev/srv/diagnostics/ | grep heapdump

-rw-r--r-- 1 root root 1804109 Jul 20 21:16

apic-em-inventory-manager-service-76f7f8d7f5-427m5.heapdump

> **Note**: If no heapdump file was found in the container directory then no crashing state was present in the container.

Unable to Delete a Device

In some situations Cisco DNA Center can be unable to delete a network device from the Inventory User Interfcae due to a backend issue.

API to Force Device Delete

If you are unabel to delete the device from Inventory using the Cisco DNA Center GUI, you can use the API to delete the device by id:

1.- Navigate to the Cisco DNA Center **Menu -> Platform -> Developer Toolkit -> APIs Tab** and search for **Devices** in the searchbar, from the results click in **Devices** from the **Know your network** section and search for the **DELETE by Device Id** API.

2.- Click in the **DELETE by Device Id** API, click in **Try** and provide the device id from the desired device to be removed form inventory.

3.- Wait for the API to run and get a 200 OK response, then confirm that the network device is not present in the Inventory page anymore.

Tip: You can get the device uuid from the Browser URL (deviceid or id) from the Cisco DNA Center Inventory Device Details page or Device View 360 page.

Note: For more information about APIs in Cisco DNA Center please review the Cisco DevNet API Guide