



## **Enhanced Monitoring and Visibility for LAN Fabric Deployments, Release 11.3(1)**

**First Published:** 2020-05-29

### **Americas Headquarters**

Cisco Systems, Inc.  
170 West Tasman Drive  
San Jose, CA 95134-1706  
USA  
<http://www.cisco.com>  
Tel: 408 526-4000  
800 553-NETS (6387)  
Fax: 408 527-0883

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

All printed copies and duplicate soft copies of this document are considered uncontrolled. See the current online version for the latest version.

Cisco has more than 200 offices worldwide. Addresses and phone numbers are listed on the Cisco website at [www.cisco.com/go/offices](http://www.cisco.com/go/offices).

The documentation set for this product strives to use bias-free language. For purposes of this documentation set, bias-free is defined as language that does not imply discrimination based on age, disability, gender, racial identity, ethnic identity, sexual orientation, socioeconomic status, and intersectionality. Exceptions may be present in the documentation due to language that is hardcoded in the user interfaces of the product software, language used based on standards documentation, or language that is used by a referenced third-party product.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: <https://www.cisco.com/c/en/us/about/legal/trademarks.html>. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1721R)

© 2019 Cisco Systems, Inc. All rights reserved.



## CONTENTS

---

### CHAPTER 1

#### **Endpoint Locator 1**

Endpoint Locator	1
Configuring Endpoint Locator	2
Enabling High Availability	10
Flushing the Endpoint Database	10
Configuring Endpoint Locator in DCNM High Availability Mode	11
Configuring Endpoint Locator in DCNM Cluster Mode	12
Configuring Endpoint Locator for External Fabrics	14
Configuring Endpoint Locator for eBGP EVPN Fabrics	14
EPL Connectivity Options	17
Disabling Endpoint Locator	21
Troubleshooting Endpoint Locator	21
Monitoring Endpoint Locator	24
Endpoint Locator Dashboard	24
Endpoint History	27
Endpoint Search	32
Endpoint Life	33

---

### CHAPTER 2

#### **DCNM Integration with ServiceNow 35**

DCNM Integration with ServiceNow	35
Guidelines and Limitations of DCNM Integration with ServiceNow	36
Installing and Configuring the Cisco DCNM Application on ServiceNow	37
Viewing the Dashboard	40
Troubleshooting DCNM Integration with ServiceNow	42

---

### CHAPTER 3

#### **Template Usage in Cisco DCNM LAN Fabric Deployment 43**

- Policy Template 43
- Fabric Template 47
- Profile Template 47
- Viewing, Editing, and Adding Policies 48
  - Viewing Policies 49
  - Editing Policies 51
  - Adding Policies 52
- Deploying New Configurations 52
- switch\_freeform Template Usage 53
  - Example: Create a switch\_freeform policy 53
- Changing the Contents of a Template in Use 56





## CHAPTER 1

# Endpoint Locator

- [Endpoint Locator](#) , on page 1
- [Monitoring Endpoint Locator](#), on page 24

## Endpoint Locator

The Endpoint Locator (EPL) feature allows real-time tracking of endpoints within a data center. The tracking includes tracing the network life history of an endpoint and getting insights into the trends that are associated with endpoint additions, removals, moves, and so on. An endpoint is anything with at least one IP address (IPv4 and/or IPv6) and MAC address. Starting from Cisco DCNM Release 11.3(1), the EPL feature is also capable of displaying MAC-Only endpoints. By default, MAC-Only endpoints are not displayed. An endpoint can be a virtual machine (VM), container, bare-metal server, service appliance and so on.



### Important

- EPL is supported for VXLAN BGP EVPN fabric deployments only in the DCNM LAN fabric installation mode. The VXLAN BGP EVPN fabric can be deployed as Easy fabric, Easy eBGP fabric, or an External fabric (managed or monitored mode). EPL is not supported for 3-tier access-aggregation-core based network deployments.
- EPL displays endpoints that have at least one IP address (IPv4 and/or IPv6). Starting from Cisco DCNM Release 11.3(1), EPL is also capable of displaying MAC-Only endpoints. Select the **Process MAC-Only Advertisements** checkbox while configuring EPL to enable processing of EVPN Route-type 2 advertisements having a MAC address only. L2VNI:MAC is the unique endpoint identifier for all such endpoints. EPL can now track endpoints in Layer-2 only network deployments where the Layer-3 gateway is on a firewall, load-balancer, or other such nodes.

EPL relies on BGP updates to track endpoint information. Hence, typically the DCNM needs to peer with the BGP Route-Reflector (RR) to get these updates. For this purpose, IP reachability from the DCNM to the RR is required. This can be achieved over in-band network connection to the DCNM eth2 interface.

Some key highlights of the Endpoint Locator are:

- Support for dual-homed and dual-stacked (IPv4 + IPv6) endpoints
- Support for up to two BGP Route Reflectors or Route Servers
- Support real-time and historical search for all endpoints across various search filters such as VRF, Network, Layer-2 VNI, Layer-3 VNI, Switch, IP, MAC, port, VLAN, and so on.

- Support for real-time and historical dashboards for insights such as endpoint lifetime, network, endpoint, VRF daily views, and operational heat map.
- Support for iBGP and eBGP based VXLAN EVPN fabrics. From Release 11.2(1), the fabrics may be created as Easy Fabrics or External Fabrics. EPL can be enabled with an option to automatically configure the spine or RRs with the appropriate BGP configuration (new in DCNM 11.2).
- Starting from Cisco DCNM Release 11.3(1), you can enable the EPL feature for upto 4 fabrics. This is supported only in clustered mode.
- Starting from Cisco DCNM Release 11.3(1), EPL is supported on Multi-Site Domain (MSD).
- Starting from Cisco DCNM Release 11.3(1), IPv6 underlay is supported.
- Support for high availability
- Support for endpoint data that is stored for up to 180 days, amounting to a maximum of 100 GB storage space.
- Support for optional flush of the endpoint data in order to start afresh.
- Supported scale: 50K unique endpoints per fabric. A maximum of 4 fabrics is supported. However, the maximum total number of endpoints across all fabrics should not exceed 100K.

For more information about EPL, refer to the following sections:

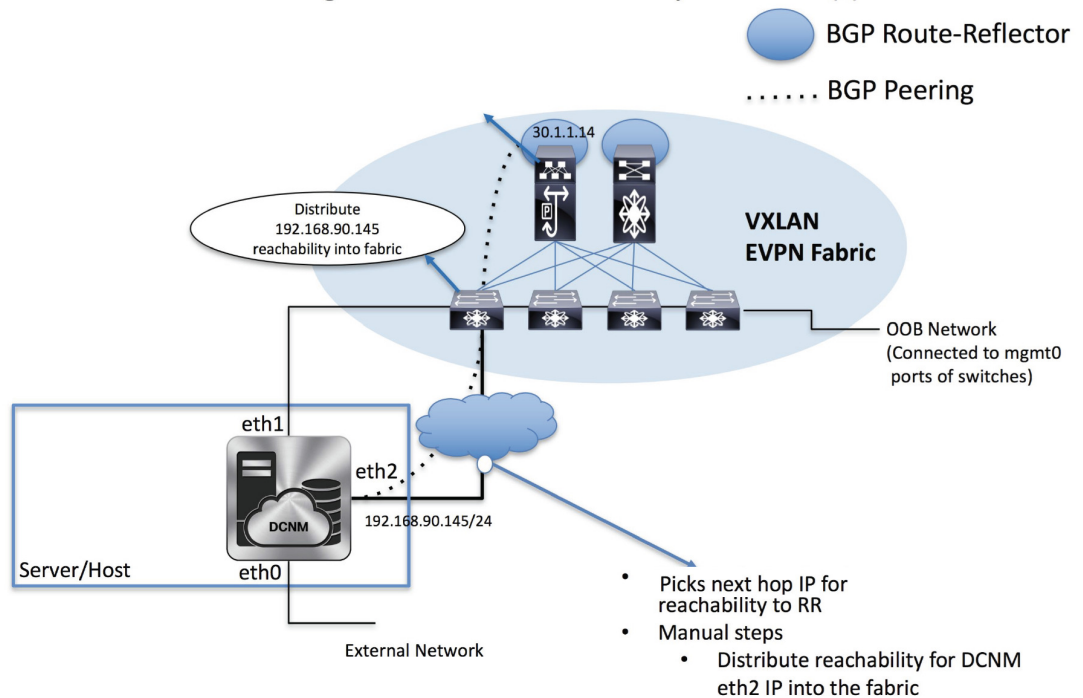
## Configuring Endpoint Locator

The DCNM OVA or the ISO installation comes with three interfaces:

- eth0 interface for external access
- eth1 interface for fabric management (Out-of-band or OOB)
- eth2 interface for in-band network connectivity

## Configuration

The Server Hosting DCNM has IP connectivity to BGP RR(s)



The eth1 interface provides reachability to the devices via the mgmt0 interface either Layer-2 or Layer-3 adjacent. This allows DCNM to manage and monitor these devices including POAP. EPL requires BGP peering between the DCNM and the Route-Reflector. Since the BGP process on Nexus devices typically runs on the default VRF, in-band IP connectivity from the DCNM to the fabric is required. For this purpose, the eth2 interface can be configured using the **appmgr update network-properties** command. Optionally, you can configure the eth2 interface during the Cisco DCNM installation.

If you need to modify the already configured in-band network (eth2 interface), run the **appmgr update network-properties** command again. Refer [Editing Network Properties Post DCNM Installation](#) to run the **appmgr update network-properties** command.

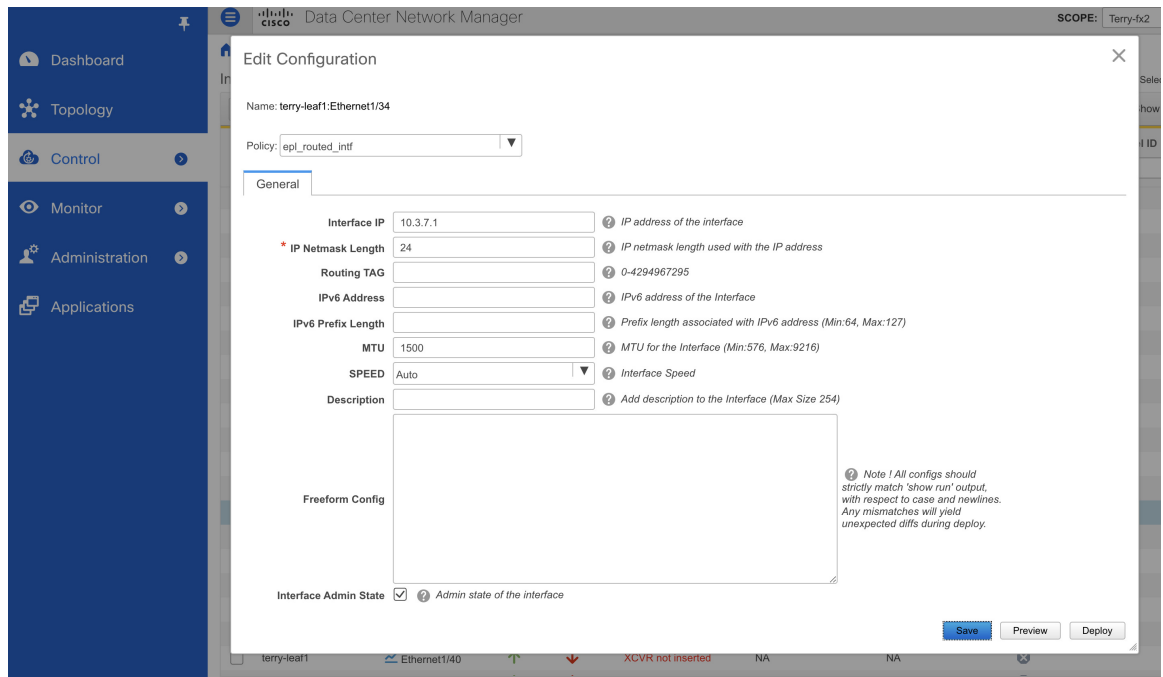


**Note** The setup of eth2 interface on the DCNM is a prerequisite of any application that requires the in-band connectivity to the devices within fabric. This includes EPL and Network Insights Resources (NIR).

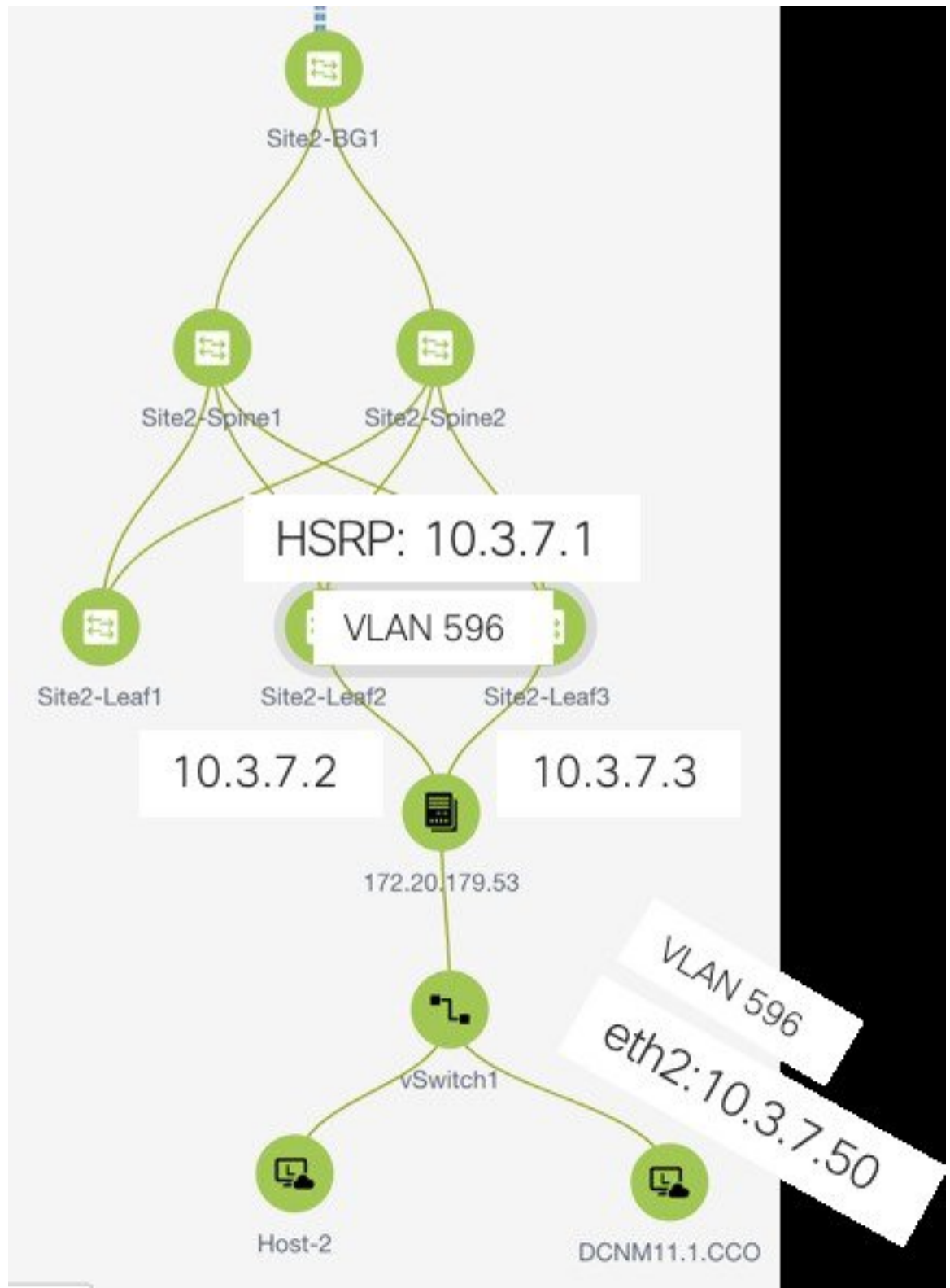


**Note** For configuring EPL in standalone mode, you must add a single neighbor to EPL. DCNM eth2 IP address is EPL IP.

On the fabric side, for a standalone DCNM deployment, if the DCNM eth2 port is directly connected to one of the front-end interfaces on a leaf, then that interface can be configured using the **epl\_routed\_intf** template. An example scenario of how this can be done when IS-IS or OSPF is employed as the IGP in the fabric, is depicted below:



However, for redundancy purposes, it is always advisable to have the server on which the DCNM is installed to be dual-homed or dual-attached. With the OVA DCNM deployment, the server can be connected to the switches via a port-channel. This provides link-level redundancy. To also have node-level redundancy on the network side, the server may be attached to a vPC pair of Leaf switches. In this scenario, the switches must be configured such that the HSRP VIP serves as the default gateway of the eth2 interface on the DCNM. The following image depicts an example scenario configuration:



In this example, the server with the DCNM VM is dual-attached to a vPC pair of switches that are named Site2-Leaf2 and Site2-Leaf3 respectively. VLAN 596 associated with the IP subnet 10.3.7.0/24 is employed for in-band connectivity. You can configure the vPC host port toward the server using the **interface vpc trunk host** policy as shown in the following image:

Add Interface
✕

\* Type:

\* Select a vPC pair:

\* vPC ID:

\* Policy:

Note : PeerOne = Site2-Leaf2 & PeerTwo = Site2-Leaf3

General

Peer-1 Member Interfaces	<input type="text" value="e1/47"/>	<small>? A list of member interfaces for Peer-1 [e.g. e1/5,eth1/7-9]</small>
Peer-2 Member Interfaces	<input type="text" value="e1/47"/>	<small>? A list of member interfaces for Peer-2 [e.g. e1/5,eth1/7-9]</small>
* Port Channel Mode	<input type="text" value="on"/>	<small>? Channel mode options: on, active and passive</small>
* Enable BPDU Guard	<input type="text" value="true"/>	<small>? Enable spanning-tree bpduguard</small>
Enable Port Type Fast	<input checked="" type="checkbox"/>	<small>? Enable spanning-tree edge port behavior</small>
* MTU	<input type="text" value="jumbo"/>	<small>? MTU for the Port Channel</small>
* Peer-1 Trunk Allowed...	<input type="text" value="596"/>	<small>? Peer-1 Trunk Allowed Vlans</small>
* Peer-2 Trunk Allowed	<input type="text" value="596"/>	<small>? Peer-2 Trunk Allowed Vlans</small>

For the HSRP configuration on Site2-Leaf2, the **switch\_freeform** policy may be employed as shown in the following image:

**Edit Policy**

Policy ID: POLICY-237060  
 Entity Type: SWITCH  
 \* Priority (1-1000): 500

Template Name: switch\_freeform\_config  
 Entity Name: SWITCH

General

Variables:

\* Freeform Config CLI

```
feature hsrp
vlan 596
interface vlan 596
ip address 10.3.7.3/24
ip router ospf UNDERLAY area 0.0.0.0
no shutdown
no ip redirects
no ipv6 redirects
hsrp 10
ip 10.3.7.1
```

? Additional CLI not in othe

Save Deploy Cancel

You can deploy a similar configuration on Site2-Leaf3 while using IP address 10.3.7.2/24 for SVI 596. This establishes an in-band connectivity from the DCNM to the fabrics over the eth2 interface with the default gateway set to 10.3.7.1.

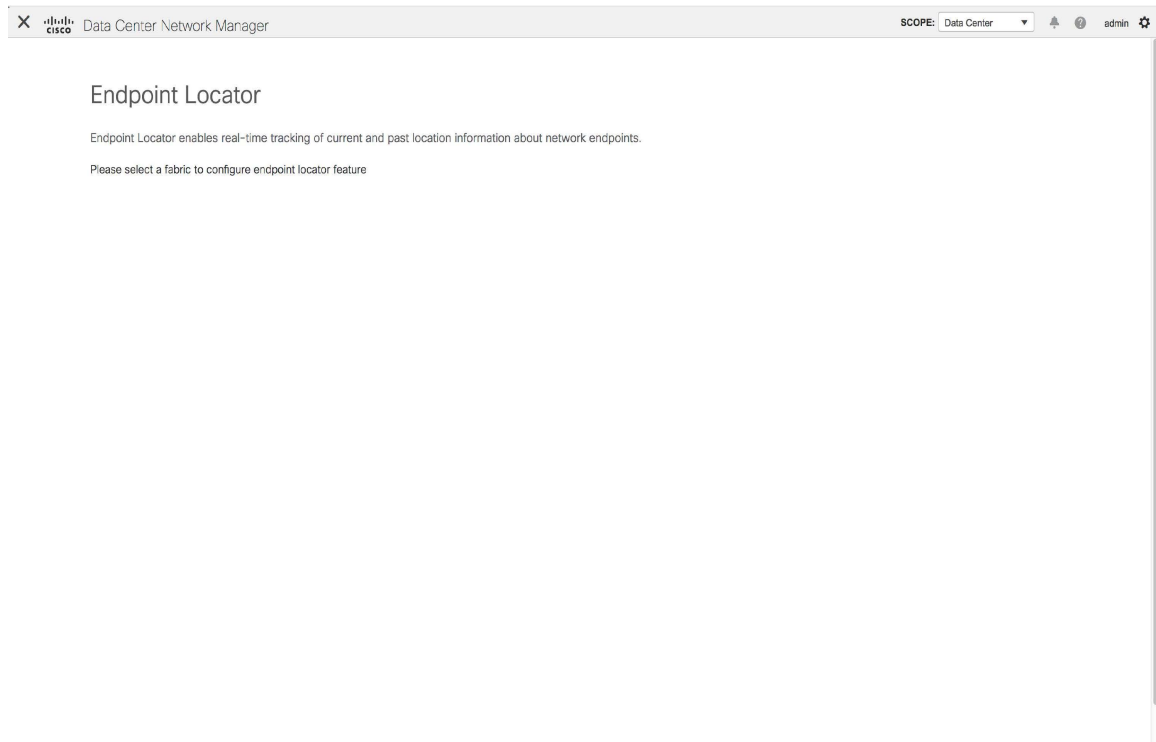
After you establish the in-band connectivity between the physical or virtual DCNM and the fabric, you can establish BGP peering.

During the EPL configuration, the route reflectors (RRs) are configured to accept DCNM as a BGP peer. During the same configuration, the DCNM is also configured by adding routes to the BGP loopback IP on the spines/RRs via the eth2 gateway.

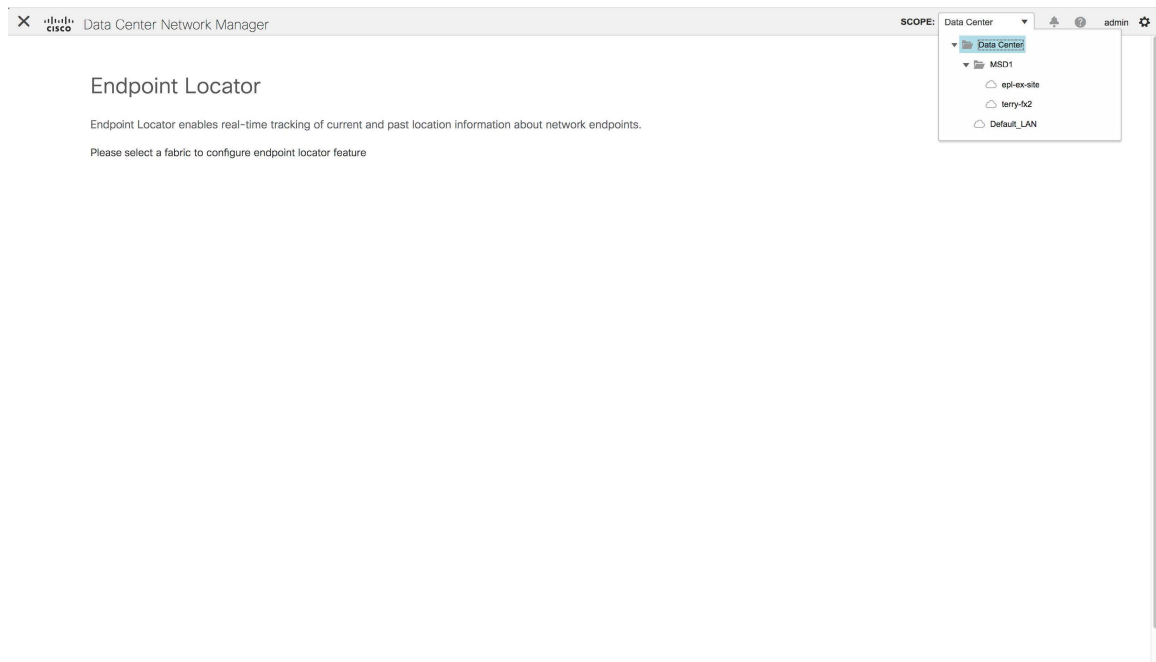


**Note** Cisco DCNM queries the BGP RR to glean information for establishment of the peering, like ASN, RR, IP, and so on.

To configure Endpoint Locator from the Cisco DCNM Web UI, choose **Control > Endpoint Locator > Configure**. The **Endpoint Locator** window appears.

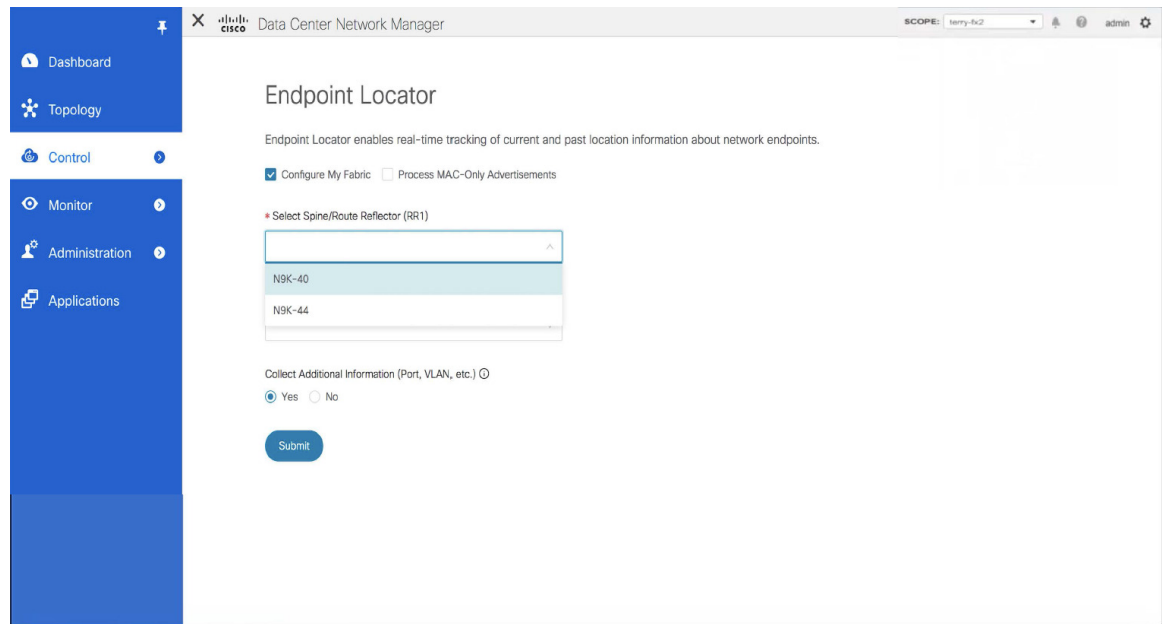


Select a fabric from the **Scope** drop-down list on which the endpoint locator feature should be enabled to track endpoint activity. You can enable EPL for one fabric at a time.



Select the switches on the fabric hosting the RRs from the drop-down list. Cisco DCNM will peer with the RRs.





By default, the **Configure My Fabric** option is selected. This knob controls whether BGP configuration will be pushed to the selected spines/RRs as part of the enablement of the EPL feature. If the spine/RR needs to be configured manually with a custom policy for the EPL BGP neighborhood, then this option should be unchecked. For external fabrics that are only monitored and not configured by DCNM, this option is greyed out as these fabrics are not configured by DCNM.

Select the **Process MAC-Only Advertisements** option to enable processing of MAC-Only advertisements while configuring the EPL feature.

Select **Yes** under **Collect Additional Information** to enable collection of additional information such as PORT, VLAN, VRF etc. while enabling the EPL feature. To gather additional information, NX-API must be supported and enabled on the switches, ToRs, and leafs. If the **No** option is selected, this information will not be collected and reported by EPL.

You can also watch the video that demonstrates how to configure EPL using Cisco DCNM. See [Configuring Endpoint Locator](#).

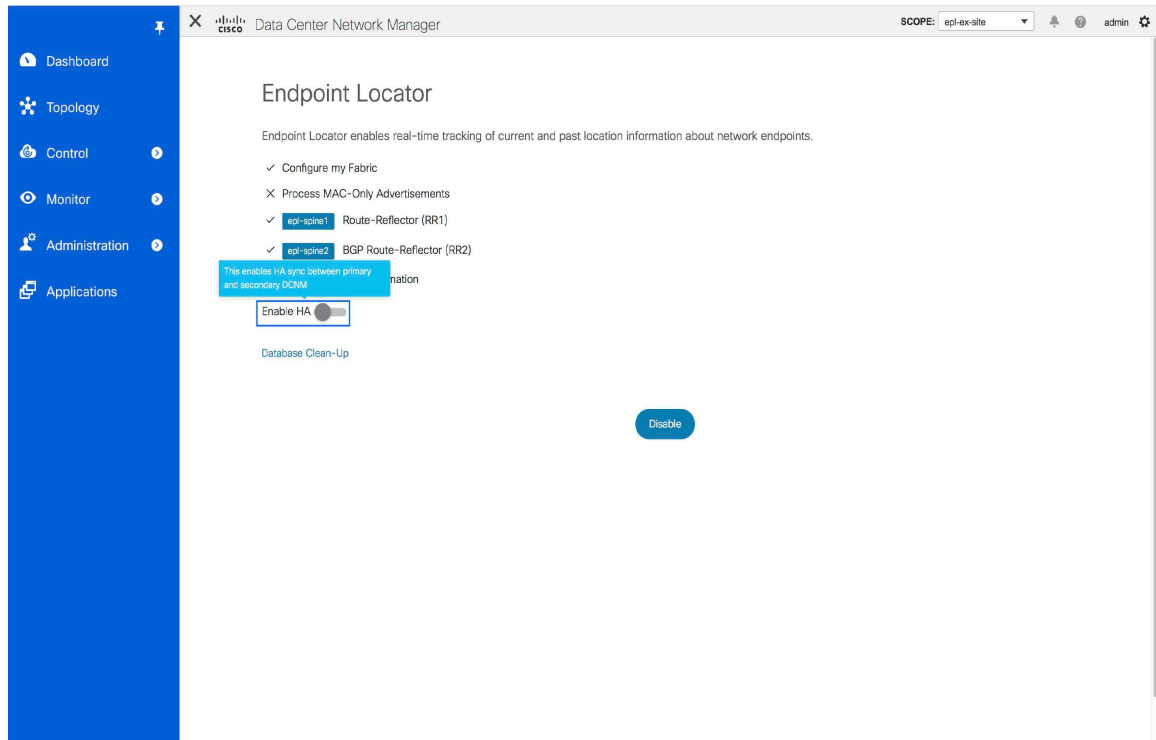
Once the appropriate selections are made and various inputs have been reviewed, click **Submit** to enable EPL. If there are any errors while you enable EPL, the enable process aborts and the appropriate error message is displayed. Otherwise, EPL is successfully enabled.

When the Endpoint Locator feature is enabled, there are a number of steps that occur in the background. DCNM contacts the selected RRs and determines the ASN. It also determines the interface IP that is bound to the BGP process. Also, appropriate BGP neighbor statements are added on the RRs or spines in case of eBGP underlay, to get them ready to accept the BGP connection that will be initiated from the DCNM. For the native HA DCNM deployment, both the primary and secondary DCNM eth2 interface IPs will be added as BGP neighbors but only one of them will be active at any given time. Once EPL is successfully enabled, the user is automatically redirected to the EPL dashboard that depicts operational and exploratory insights into the endpoints that are present in the fabric.

For more information about the EPL dashboard, refer [Monitoring Endpoint Locator](#).

## Enabling High Availability

Consider a scenario in which EPL is enabled on a DCNM deployment that is in non-HA mode and then, DCNM is moved to HA-mode. In such scenarios, the **Enable HA** toggle appears on the **Endpoint Locator** window. Toggle the **Enable HA** knob to enable high availability sync between primary and secondary DCNM.



To enable high availability sync from the Cisco DCNM Web UI, perform the following steps:

### Procedure

**Step 1** Choose **Control > Endpoint Locator > Configure**.

**Step 2** Toggle the **Enable HA** button.

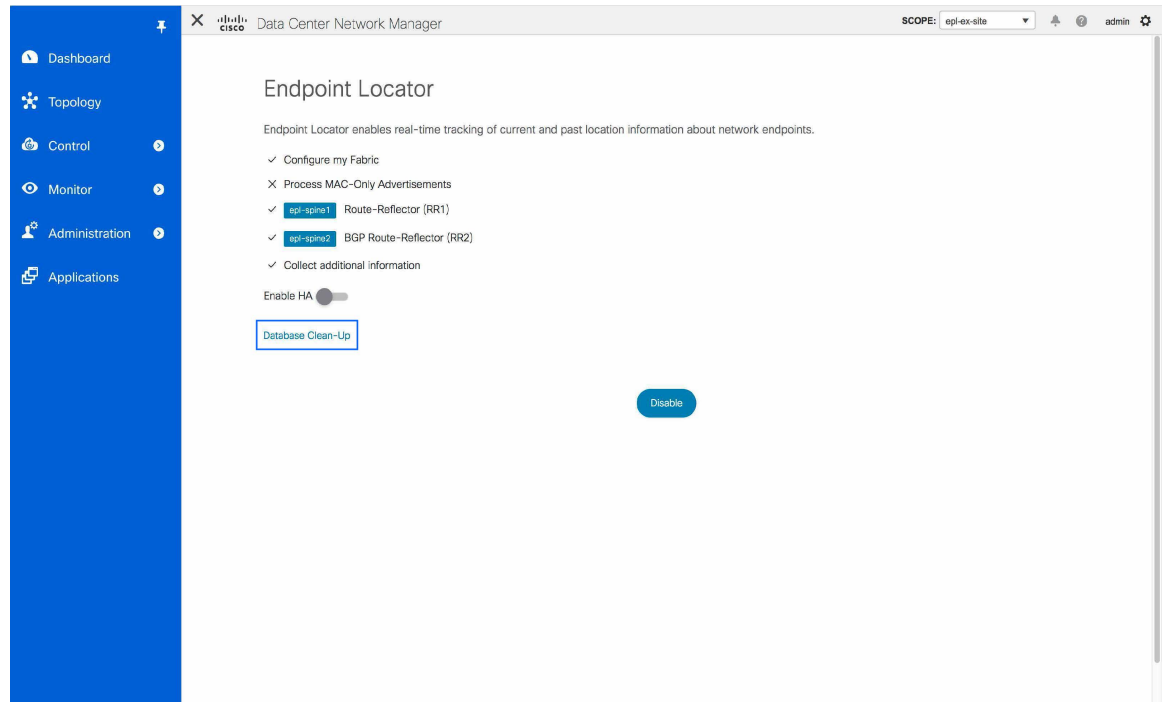
## Flushing the Endpoint Database

After you enable the Endpoint Locator feature, you can clean up or flush all the Endpoint information. This allows starting from a clean-slate with respect to ensuring no stale information about any endpoint is present in the database. After the database is clean, the BGP client re-populates all the endpoint information learnt from the BGP RR.

To flush all the Endpoint Locator information from the Cisco DCNM Web UI, perform the following steps:

## Procedure

**Step 1** Choose **Control > Endpoint Locator > Configure**, and click **Database Clean-Up**.



A warning is displayed with a message indicating that all the endpoint information that is stored in the database will be flushed.

**Step 2** Click **Delete** to continue or **Cancel** to abort.

## Configuring Endpoint Locator in DCNM High Availability Mode



**Note** For configuring EPL in native HA mode, you must add 2 neighbors to EPL. EPL IP being DCNM Primary eth2 and DCNM Secondary eth2 address respectively.

For production deployments, a native HA pair of DCNM nodes is recommended. Since the DCNM active and standby nodes need to be Layer-2 adjacent, their respective eth2 interfaces should be part of the same IP subnet or vlan. In addition, both DCNM nodes should be configured with the same eth2 gateway. The recommended option is to connect the DCNM active and standby nodes to a vPC pair of nexus switches (they may be leafs) so that there is enough fault-tolerance in case of single link failure, single device or a single DCNM node failure.

The following example shows a sample output for the **apmgrp update network-properties** command for a Cisco DCNM Native HA Appliance. In this example, 1.1.1.2 is the primary eth2 interface IP address, 1.1.1.3 is the standby eth2 interface IP address, 1.1.1.1 is the default gateway and 1.1.1.4 is the virtual IP (VIP) for inband.

On Cisco DCNM Primary appliance:

```
appmgr update network-properties session start
appmgr update network-properties set ipv4 eth2 1.1.1.2 255.255.255.0 1.1.1.1
appmgr update network-properties set ipv4 peer2 1.1.1.3
appmgr update network-properties set ipv4 vip2 1.1.1.4 255.255.255.0
appmgr update network-properties session apply
appmgr update ssh-peer-trust
```

On Cisco DCNM Secondary appliance:

```
appmgr update network-properties session start
appmgr update network-properties set ipv4 eth2 1.1.1.3 255.255.255.0 1.1.1.1
appmgr update network-properties set ipv4 peer2 1.1.1.2
appmgr update network-properties set ipv4 vip2 1.1.1.4 255.255.255.0
appmgr update network-properties session apply
appmgr update ssh-peer-trust
```

After the in-band connectivity is established from both the Primary and Secondary nodes to the Fabric, to configure endpoint locator in DCNM HA mode from the Cisco DCNM Web UI, perform the following steps:

### Procedure

- 
- Step 1** Choose **Control > Endpoint Locator > Configure**.  
The **Endpoint Locator** window appears and the fabric configuration details are displayed.
  - Step 2** Select a fabric from the **SCOPE** dropdown list to configure endpoint locator in DCNM HA mode.
  - Step 3** Select the Route-Reflectors (RRs) from the drop-down lists.
  - Step 4** Select **Yes** under **Collect Additional Information** to enable collection of additional information such as PORT, VLAN, VRF etc. while enabling the EPL feature. If the No option is selected, this information will not be collected and reported by EPL.
  - Step 5** Click **Submit**.
- 

### What to do next

After you configure the Endpoint Locator in HA mode, you can view details such as Endpoint Activity and Endpoint History in the Endpoint Locator dashboard. To view these details, navigate to **Monitor > Endpoint Locator > Explore**.

## Configuring Endpoint Locator in DCNM Cluster Mode

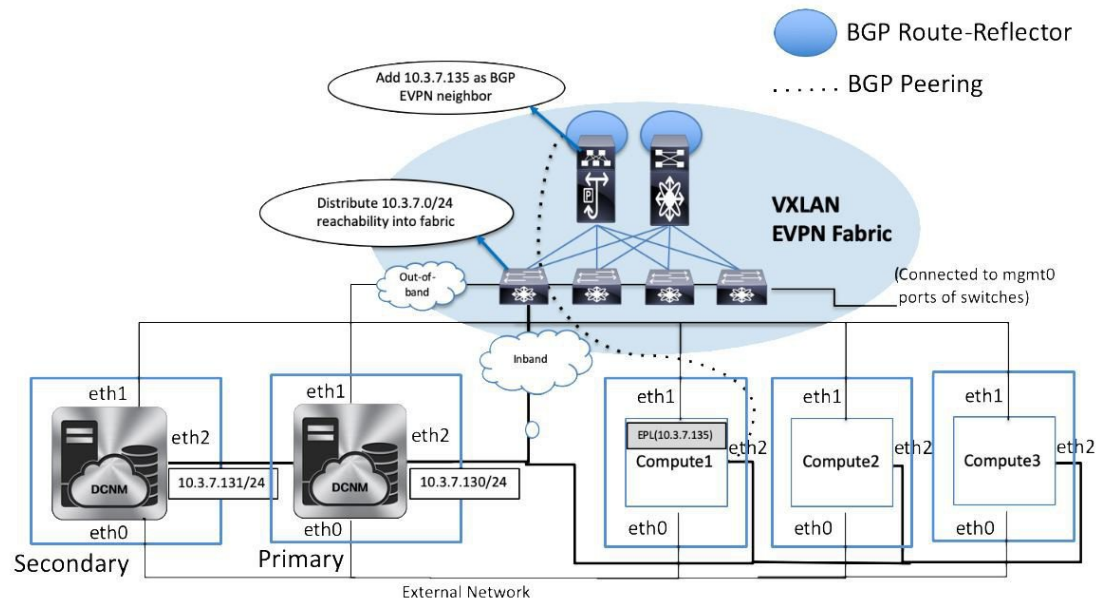



---

**Note** For configuring EPL in cluster mode, you must add a single neighbor to EPL. DCNM EPL container Inband IP address is EPL IP.

---

With the DCNM cluster mode deployment, in addition to the DCNM nodes, an additional 3 compute nodes are present in the deployment. For information about deploying applications in cluster mode, see *Cisco DCNM in Clustered Mode*.



In DCNM Cluster mode, all applications including EPL run on the compute nodes. The DCNM application framework takes care of the complete life cycle management of all applications that run on the compute nodes. The EPL instance runs as a container that has its own IP address allocated out of the inband pool assigned to the compute nodes. This IP address will be in the same IP subnet as the one allocated to the eth2 or inband interface. Using this IP address, the EPL instance forms a BGP peering with the spines/RRs when the EPL feature is enabled. If a compute node hosting the EPL instance will go down, the EPL instance will be automatically respawned on one of the remaining 2 compute nodes. All IP addresses and other properties associated with the EPL instance are retained.

The Layer-2 adjacency requirement of the compute nodes dictates that the compute node eth2 interfaces should be part of the same IP subnet as the DCNM nodes. Again, in this case, connecting the compute nodes to the same vPC pair of switches is the recommended deployment option. Note that for cluster mode DCNM OVA setups, ensure that promiscuous mode is enabled in the port group corresponding to eth2 interface in order to establish inband connectivity as depicted below:

## EPL-Inband - Edit Settings


Properties			
Security	Promiscuous mode	<input checked="" type="checkbox"/> Override	Accept
Traffic shaping	MAC address changes	<input checked="" type="checkbox"/> Override	Accept
Teaming and failover	Forged transmits	<input checked="" type="checkbox"/> Override	Accept

CANCEL

OK

The enablement of the EPL feature for DCNM cluster mode is identical to that in the non-cluster mode. The main difference is that on the spine/RRs, only a single BGP neighborship is required that points to the IP address allocated to the EPL instance. Recall that for the DCNM native HA deployment in the non-cluster mode, all spines/RRs always had 2 configured BGP neighbors, one pointing to the DCNM primary eth2 interface and other one pointing to the DCNM secondary eth2 interface. However, only one neighbor would be active at any given time.

## Configuring Endpoint Locator for External Fabrics

In addition to Easy fabrics, DCNM Release 11.2(1) allows you to enable EPL for VXLAN EVPN fabrics comprising of switches that are imported into the external fabric. The external fabric can be in managed mode or monitored mode, based on the selection of **Fabric Monitor Mode** flag in the **External Fabric Settings**. For external fabrics that are only monitored and not configured by DCNM, this flag is disabled. Therefore, you must configure BGP sessions on the Spine(s) via OOB or using the CLI. To check the sample template, click  to view the configurations required while enabling EPL.

In case the **Fabric Monitor Mode** checkbox in the External Fabric settings is unchecked, then EPL can still configure the spines/RRs with the default **Configure my fabric** option. However, disabling EPL would wipe out the router bgp config block on the spines/RRs. To prevent this, the BGP policies must be manually created and pushed onto the selected spines/RRs.

## Configuring Endpoint Locator for eBGP EVPN Fabrics

From Cisco DCNM Release 11.2(1), you can enable EPL for VXLAN EVPN fabrics, where eBGP is employed as the underlay routing protocol. Note that with an eBGP EVPN fabric deployment, there is no traditional RR similar to iBGP. The reachability of the in-band subnet must be advertised to the spines that behave as Route

Servers. To configure EPL for eBGP EVPN fabrics from the Cisco DCNM Web UI, perform the following steps:

## Procedure

### Step 1 Choose **Control > Fabric Builder**.

Select the fabric to configure eBGP on or create eBGP fabric with the **Easy\_Fabric\_eBGP** template.

Add Fabric
✕

\* Fabric Name :

\* Fabric Template :

General	EVPN	vPC	Advanced	Manageability	Bootstrap	Configuration Backup
<p>* BGP ASN for Spines <input type="text" value="65535"/> <small>? 1-4294967295   1-65535[.0-65535]</small></p> <p>* BGP AS Mode <input type="text" value="Multi-AS"/> <small>? Multi-AS: Unique ASN per Leaf/Border Dual-AS: One ASN for all Leafs/Borders</small></p> <p>* Routing Loopback Id <input type="text" value="0"/> <small>? 0-512</small></p> <p>* Underlay Subnet IP Mask <input type="text" value="30"/> <small>? Mask for Underlay Subnet IP Range</small></p> <p>Manual Underlay IP Address Allocation <input type="checkbox"/> <small>? Checking this will disable Dynamic Underlay IP Address Allocations</small></p> <p>* Underlay Routing Loopback IP Range <input type="text" value="10.2.0.0/22"/> <small>? Typically Loopback0 IP Address Range</small></p> <p>* Underlay Subnet IP Range <input type="text" value="10.4.0.0/16"/> <small>? Address range to assign Numbered and Peer Link SVI IPs</small></p> <p>* Subinterface Dot1q Range <input type="text" value="2-511"/> <small>? Per Border Dot1q Range For VRF Lite Connectivity (Min:2, Max:511)</small></p> <p>NX-OS Software Image Version <input type="text"/> <small>? If Set, Image Version Check Enforced On All Switches. Images Can Be Uploaded From Control:Image Upload</small></p>						

### Step 2 Use the **leaf\_bgp\_asn** policy to configure unique ASNs on all leaves.

View/Edit Policies for leaf1 ( FDO23070AC0 )

Add Policy ✕

\* Priority (1-1000):

\* Policy:

General

\* Leaf BGP AS #  ? Leaf BGP Autonomous System number

Variables:

- Step 3** Add the **ebgp\_overlay\_leaf\_all\_neighbor** policy to each leaf.  
 Fill **Spine IP List** with the spines' BGP interface IP addresses, typically the loopback0 IP addresses.  
 Fill **BGP Update-Source Interface** with the leaf's BGP interface, typically loopback0.

View/Edit Policies for leaf1 ( FDO23070AC0 )

Add Policy ✕

\* Priority (1-1000):

\* Policy:

General

\* Spine IP List  ? list of spine IP address for peering list e.g. 10.2.

\* BGP Update-Source Interface  ? Source of BGP session and updates

Enable Tenant Routed Multicast  ? For Overlay Multicast Support In VXLAN Fabrics

Enable BGP Authentication  ? BGP Authentication needs to match the fabric setting

Variables:

- Step 4** Add the **ebgp\_overlay\_spine\_all\_neighbor** policy to each spine.  
 Fill **Leaf IP List** with the leaves' BGP interface IPs, typically the loopback0 IPs.



Fill **Leaf BGP ASN** with the leaves' ASNs in the same order as in **Leaf IP List**.

Fill **BGP Update-Source Interface** with the spine's BGP interface, typically loopback0.

View/Edit Policies for spine ( FDO231003AG )

Add Policy ✕

\* Priority (1-1000):

\* Policy:  ▼

General

---

\* Leaf IP List  ? *list of leaf IP address for peering list e.g. 10.2.0.*

\* Leaf BGP ASN  ? *BGP ASN of each leaf, separated by ,*

\* BGP Update-Source Interface  ? *Source of BGP session and updates*

Enable Tenant Routed Multicast  ? *Tenant Routed Multicast setting needs to match the fabric setting*

Enable BGP Authentication  ? *BGP Authentication needs to match the fabric setting*

Variables:

After the in-band connectivity is established, the enablement of the EPL feature remains identical to what is listed so far. EPL becomes a iBGP neighbor to the Route Servers running on the spines.

## EPL Connectivity Options

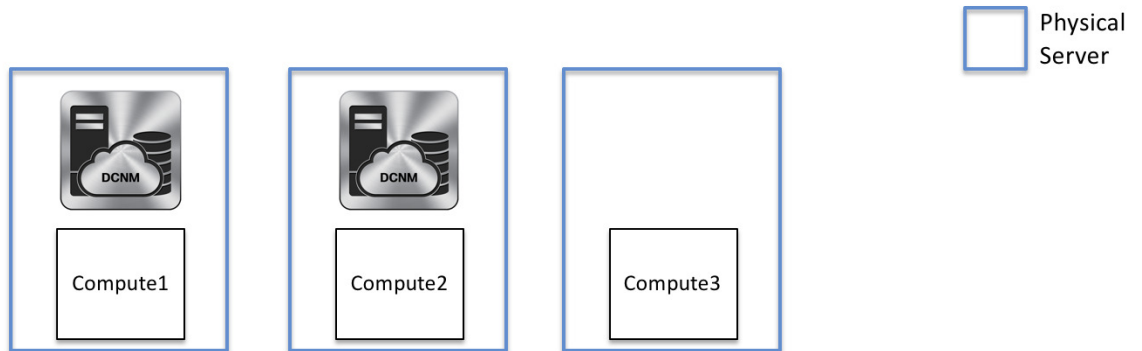
Sample topologies for the various EPL connectivity options are as given below.

Cisco DCNM supports the following web browsers:

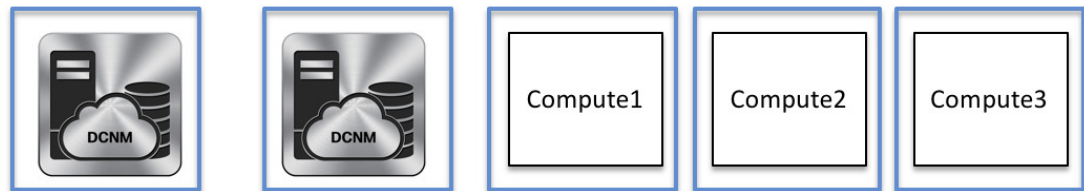
### DCNM Cluster Mode: Physical Server to VM Mapping

We recommend a minimum of 3 physical servers, or a maximum of 5 physical servers in which each DCNM and compute is located on an individual physical server.

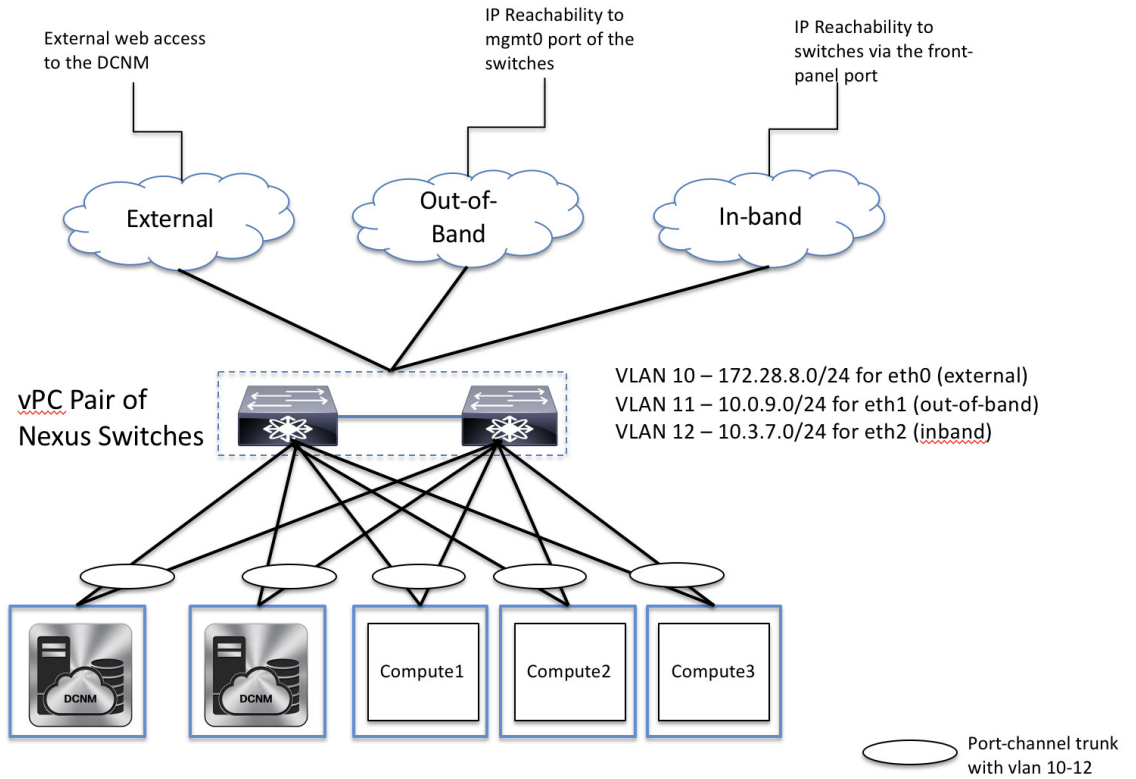
*Figure 1: A minimum of 3 physical servers*



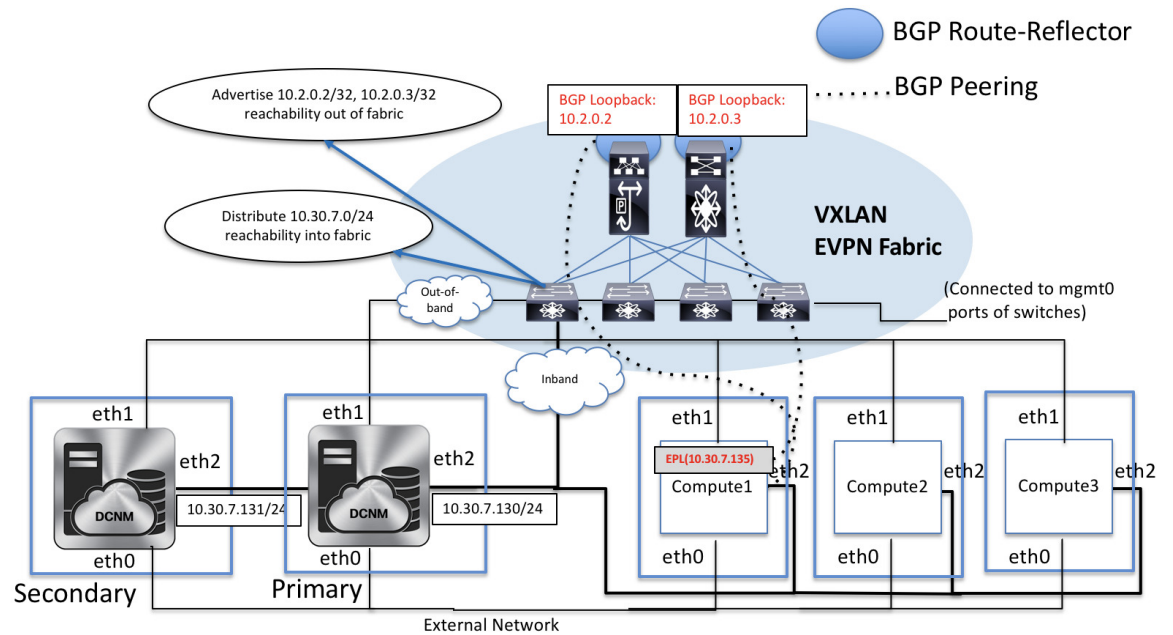
*Figure 2: A maximum of 5 physical servers*



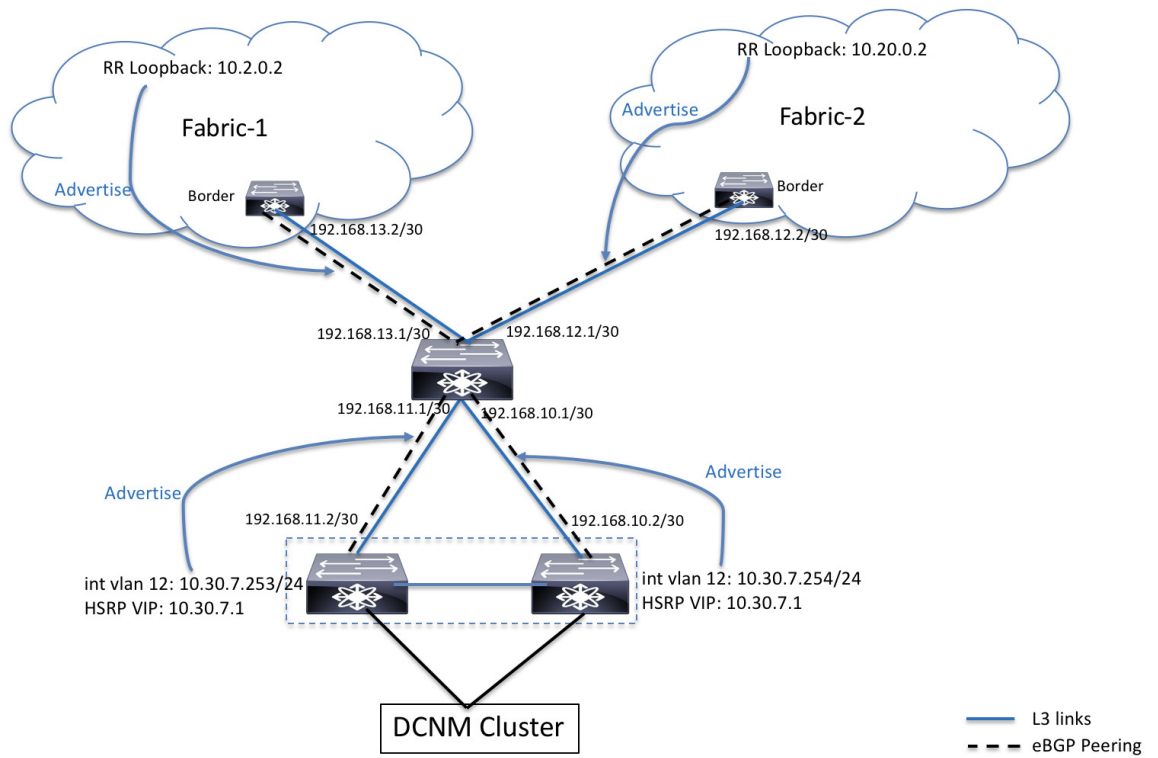
### DCNM/Compute VM Physical Connectivity



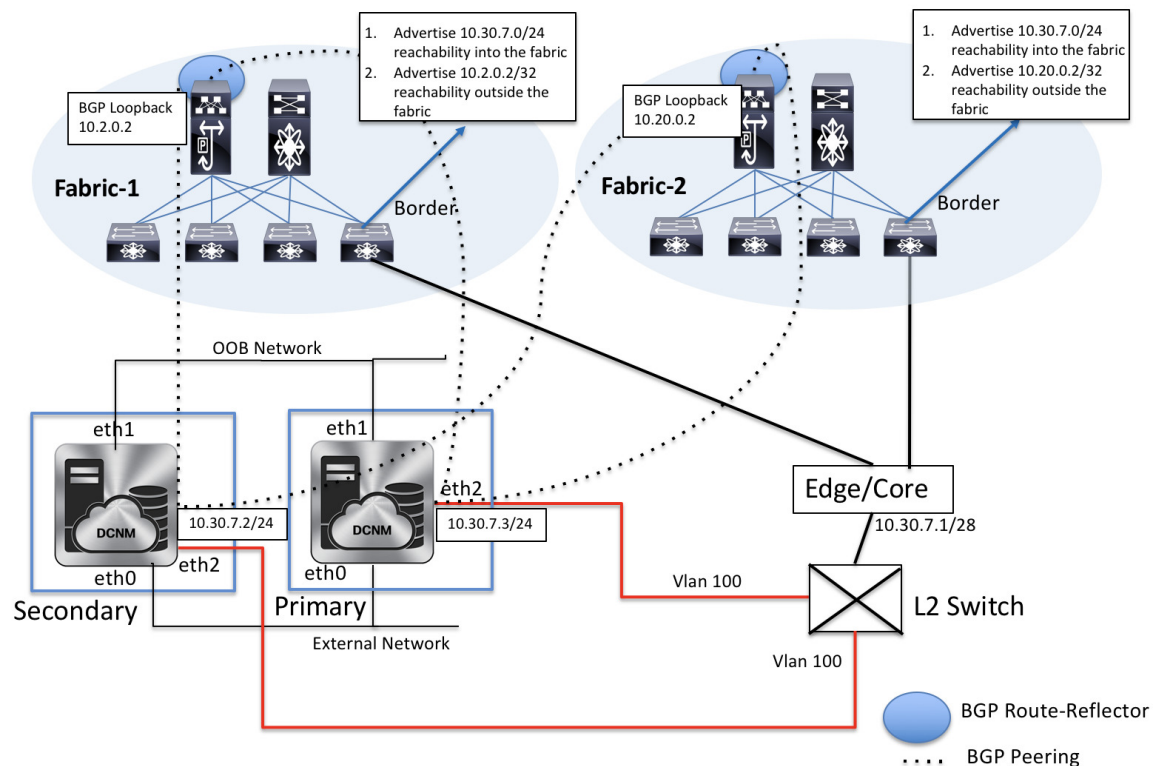
### DCNM Cluster Mode



### DCNM Multi-Fabric Connectivity



### EPL Connectivity for Native HA



## Disabling Endpoint Locator

To disable endpoint locator from the Cisco DCNM Web UI, perform the following steps:

### Procedure

**Step 1** Choose **Control > Endpoint Locator > Configure**.

The **Endpoint Locator** window appears. Select the required fabric from the **SCOPE** dropdown list. The fabric configuration details are then displayed for the selected fabric.

**Step 2** Click **Disable**.

## Troubleshooting Endpoint Locator

There may be multiple reasons why enabling the Endpoint Locator feature may fail. Typically, if the appropriate devices are selected and the IP addresses to be used are correctly specified, the connectivity of the DCNM to the BGP RR may not be present due to which the feature cannot be enabled. This is a sanity check that is present to ensure that basic IP connectivity is available. The following image shows an example error scenario that was encountered during an attempt to enable the EPL feature.

The log that provides details on what occurred when the EPL feature is enabled or disabled, is present in the file `epl.log` at the location: `/usr/local/cisco/dcm/fm/logs/epl.log`. The following example provides a snapshot of the `epl.log` that shows the EPL configuration progress for a fabric.

```

2019.12.05 12:18:23 INFO [epl] Found DCNM Active Inband IP: 192.168.94.55/24
2019.12.05 12:18:23 INFO [epl] Running script: [sudo, /sbin/appmgr, setup, inband-route,
--host, 11.2.0.4]
2019.12.05 12:18:23 INFO [epl] Getting EPL configure progress for fabric 4
2019.12.05 12:18:23 INFO [epl] EPL Progress 2
2019.12.05 12:18:23 INFO [epl] [sudo, /sbin/appmgr, setup, inband-route, --host, 11.2.0.4]
command executed, any errors? No
2019.12.05 12:18:23 INFO [epl] Received response:
2019.12.05 12:18:23 INFO [epl] Validating host route input
2019.12.05 12:18:23 INFO [epl] Done configuring host route
2019.12.05 12:18:23 INFO [epl] Done.
2019.12.05 12:18:23 INFO [epl] Running script: [sudo, /sbin/appmgr, setup, inband-route,
--host, 11.2.0.5]
2019.12.05 12:18:23 INFO [epl] [sudo, /sbin/appmgr, setup, inband-route, --host, 11.2.0.5]
command executed, any errors? No
2019.12.05 12:18:23 INFO [epl] Received response:
2019.12.05 12:18:23 INFO [epl] Validating host route input
2019.12.05 12:18:23 INFO [epl] Done configuring host route
2019.12.05 12:18:23 INFO [epl] Done.
2019.12.05 12:18:23 INFO [epl] Running command: sudo /sbin/appmgr show inband
2019.12.05 12:18:24 INFO [epl] Received response: Physical IP=192.168.94.55/24
Inband GW=192.168.94.1
No IPv6 Inband GW found

2019.12.05 12:18:26 INFO [epl] Call:
http://localhost:35000/afw/apps?imagetag=cisco:epl:2.0&fabricid=epl-ex-site, Received
response:
2019.12.05 12:18:26 INFO [epl] Epl started on AFW

```

After the EPL is enabled successfully, all the debug, error, and info logs associated with endpoint information are stored in `/var/afw/applogs/` under the directory for the associated fabric. For example, if EPL is enabled for the `test` fabric, the logs will be in `/var/afw/applogs/epl_cisco_test_afw_log/epl/` starting with filename `afw_bgp.log.1`. Depending on the scale of the network and the number of endpoint events, the file size will increase. Therefore, there is a restriction on the maximum number and size of `afw_bgp.log`. Up to 10 such files will be stored with each file size of maximum of 100 MB.



**Note** EPL creates a symlink in this directory inside the docker container, hence it appears broken when accessed natively.

The EPL relies on BGP updates to get endpoint information. In order for this to work, the switch loopback or VTEP interface IP addresses must be discovered on the DCNM for all switches that have endpoints. To validate, navigate to the Cisco DCNM **Web UI > Dashboard > Switch > Interfaces** tab, and verify if the IP address and the prefix associated with the corresponding Layer-3 interfaces (typically loopbacks) are displayed correctly.

In a Cisco DCNM Cluster deployment, if EPL cannot establish BGP peering and the active DCNM is able to ping the loopback IP address of the spine, while the EPL container cannot, it implies that the eth2 port group for Cisco DCNM and its computes does not have Promiscuous mode set to **Accept**. After changing this setting, the container can ping the spine and EPL will establish BGP.

In a large-scale setup, it may take more than 30 seconds (default timer set in Cisco DCNM) to get this information from the switch. If this occurs, the `ssh.read-wait-timeout` property (in the **Administration > DCNM Server > Server Properties**) must be changed from 30000 (default) to 60000 or a higher value.

The endpoint data displayed on the dashboard may be slightly inaccurate in a large-scale setup. An approximately 1% accuracy tradeoff is made at higher endpoint counts for performance. If the dashboard greatly differs from what is expected, the validity can be checked with a verifier script that is packaged in DCNM. As root, run the `epl-rt-2.py` script in `/root/packaged-files/scripts/`. This script needs the RR/spine IP and the associated username and password. Note that the `/root/packaged-files/scripts/` directory is read only, so the script needs to be run outside that directory. For example, to run the script for a spine with IP 10.2.0.5, username admin, and password cisco123, run **`/root/packaged-files/scripts/epl-rt-2.py -s 10.2.0.5 -u admin -p cisco123`** while the working directory is `/root/`. If the EPL dashboard still does not display expected numbers and the `epl-rt-2.py` script output differs significantly from the dashboard, please contact tech support.

In cluster mode, BGP is not established between the spines/RRs and DCNM. Check that the **Promiscuous mode** setting for the port group corresponding to the eth2 DCNM interface is set to **Accept**. If a connection is still not established, perform the following steps to check the connectivity between DCNM's BGP client and the spine/RR:

1. Open a shell on the active DCNM and run the following commands:

a. `docker service ls`

\*Note the ID for the EPL service

b. `docker service ps $ID`

\*Note the NODE field

c. `afw compute list -b`

\*Note the HostIp matching the HostName (NODE) from before. This is the compute that the EPL service is currently running on.

2. Open a shell on the compute noted from Step 1 - c and run the following commands:

a. `docker container ls`

\*Note the CONTAINER ID for EPL. If there are multiple EPL containers check the container name to see which one corresponds to which fabric. The naming scheme is `epl_cisco_${FabricName}_afw.*`

b. `docker container inspect $CONTAINER_ID`

\*Note the value of SandboxKey

c. `nsenter --net=$SandboxKey`

This command enters the network namespace of the EPL container. Now network commands such as `ifconfig`, `ip`, and `ping` will act as if they're being ran from inside the container until "exit" is issued in the shell.

3. Try pinging the spine/RR. Make sure that the Inband IP Pool that the DCNM cluster is configured with does not conflict with any switch loopback IPs.

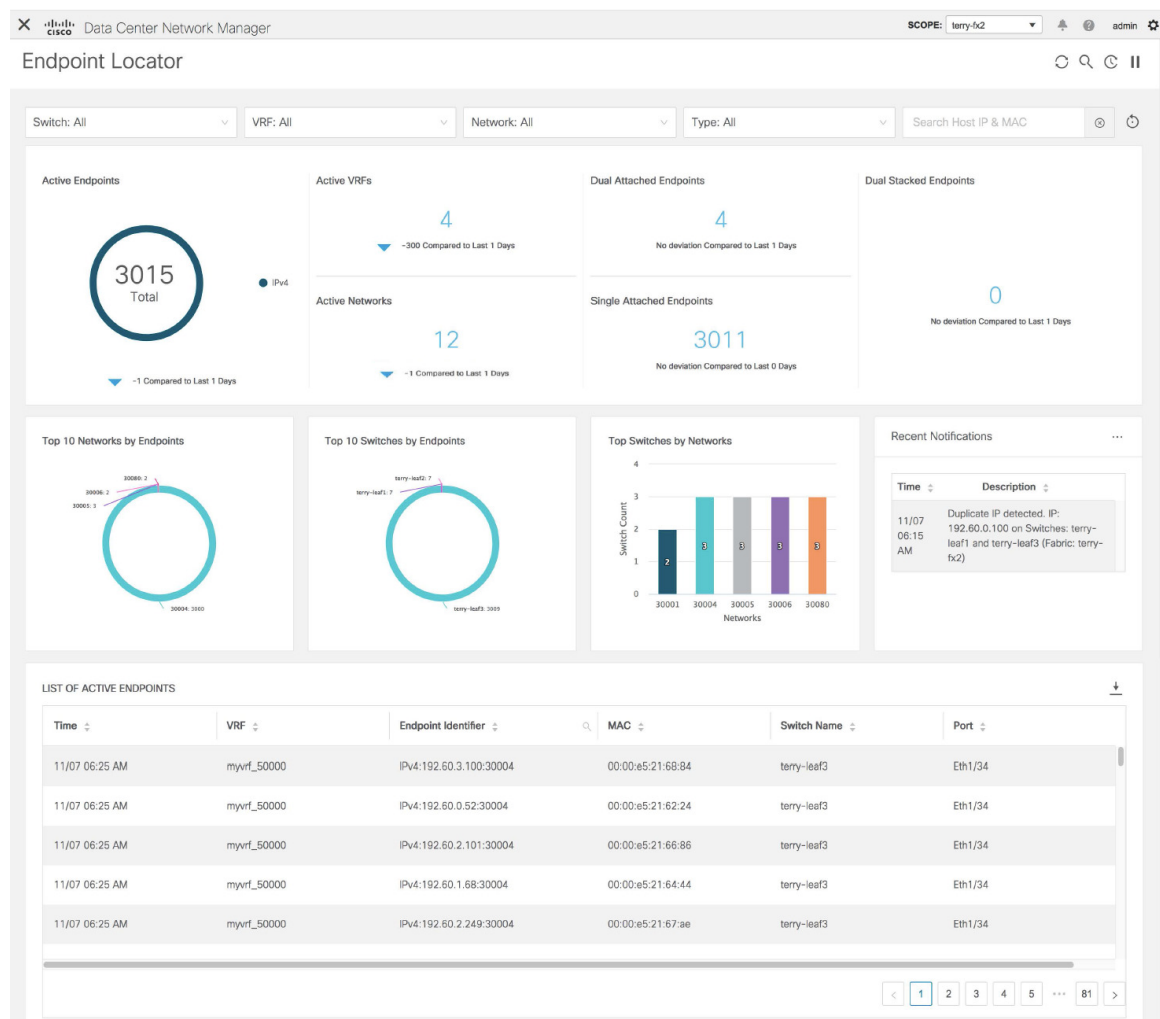
# Monitoring Endpoint Locator

Information about the Endpoint Locator is displayed on a single landing page or dashboard. The dashboard displays an almost real-time view of data (refreshed every 30 seconds) pertaining to all the active endpoints on a single pane. The data that is displayed on this dashboard depends on the scope selected by you from the **SCOPE** drop-down list. The DCNM scope hierarchy starts with the fabrics. Fabrics can be grouped into a Multi-Site Domain (MSD). A group of MSDs constitute a Data Center. The data that is displayed on the Endpoint Locator dashboard is aggregated based on the selected scope. From this dashboard, you can access Endpoint History, Endpoint Search, and Endpoint Life.

You can also watch the video that demonstrates how to monitor EPL using Cisco DCNM. See [Monitoring EPL](#).

## Endpoint Locator Dashboard

To explore endpoint locator details from the Cisco DCNM Web UI, choose **Monitor > Endpoint Locator > Explore**. The **Endpoint Locator** dashboard is displayed.







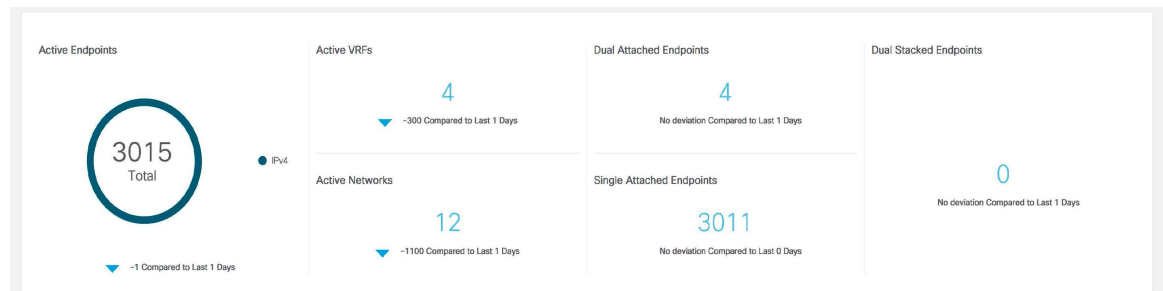
**Note** Due to an increase in scale from Cisco DCNM Release 11.3(1), the system may take some time to collect endpoint data and display it on the dashboard. Also, on bulk addition or removal of endpoints, the endpoint information displayed on the EPL dashboard takes a few minutes to refresh and display the latest endpoint data.

You can also filter and view the endpoint locator details for a specific **Switch**, **VRF**, **Network**, and **Type**, by using the respective drop-down lists. Starting from Cisco DCNM Release 11.3(1), you can select MAC type of endpoints as a filter attribute. By default, the selected option is **All** for these fields. You can also display endpoint data for a specific host by entering the IP and MAC address of a host in the **Search Host IP and MAC** field.

You can reset the filters to the default options by clicking the **Reset Filters** icon.



The 'top pane' of the window displays the number of active endpoints, active VRFs, active networks, dual attached endpoints, single attached endpoints and dual stacked endpoints, for the selected scope. Support for displaying the number of dual attached endpoints, single attached endpoints and dual stacked endpoints has been added from Cisco DCNM Release 11.3(1). A dual attached endpoint is an endpoint that is behind at least two switches. A dual stacked endpoint is an endpoint that has at least one IPv4 address and one IPv6 address.



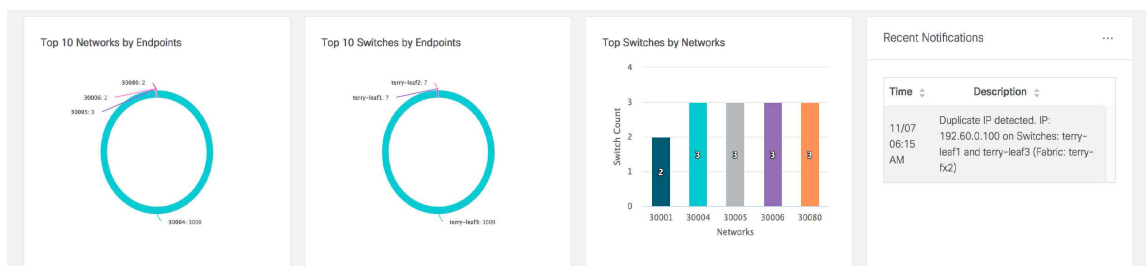
Historical analysis of data is performed and a statement mentioning if any deviation has occurred or not over the previous day is displayed at the bottom of each tile.

Click any tile in the top pane of the EPL dashboard to go to the [Endpoint History](#) window.

The 'middle pane' of the window displays the following information:

- **Top 10 Networks by Endpoints** - A pie chart is displayed depicting the top ten networks that have the most number of endpoints. Hover over the pie chart to display more information. Click on the required section to view the number of IPv4, IPv6, and MAC addresses.
- **Top 10 Switches by Endpoints** - A pie chart is displayed depicting the top ten switches that are connected to the most number of endpoints. Hover over the pie chart to display more information. Click on the required section to view the number of IPv4, IPv6, and MAC addresses.
- **Top Switches by Networks** - Bar graphs are displayed depicting the number of switches that are associated with a particular network. For example, if a vPC pair of switches is associated with a network, the number of switches associated with the network is 2.

- **Recent Notifications** - A list of the last 10 notifications is displayed. Notifications are generated for events such as Duplicate IP addresses, Duplicate MAC-Only addresses, VRF disappears from a fabric, all endpoints disappear from a switch, Endpoint moves, Endpoints on a fabric going to zero, when a switch appears for the first time, when the RR BGP connectivity status changes (RR connected status indicates that the RR is connected and the underlying Border Gateway Protocol, or BGP, is functioning normally. RR disconnected status indicates that the RR is disconnected and the underlying BGP is not functioning), and when a new VRF is detected. Click the **More** icon at the top right of this window to display the **Notifications** window. You can view the list of notifications and also delete notifications from this window by clicking **Delete**. Click the download icon to download list of the notifications as a CSV file.



The 'bottom pane' of the window displays the list of active endpoints.


Time	VRF	Endpoint Identifier	MAC	Switch Name	Port
11/07 06:25 AM	myvrf_50000	IPv4:192.60.3.100:30004	00:00:e5:21:68:84	terry-leaf3	Eth1/34
11/07 06:25 AM	myvrf_50000	IPv4:192.60.0.52:30004	00:00:e5:21:62:24	terry-leaf3	Eth1/34
11/07 06:25 AM	myvrf_50000	IPv4:192.60.2.101:30004	00:00:e5:21:66:86	terry-leaf3	Eth1/34
11/07 06:25 AM	myvrf_50000	IPv4:192.60.1.68:30004	00:00:e5:21:64:44	terry-leaf3	Eth1/34
11/07 06:25 AM	myvrf_50000	IPv4:192.60.2.249:30004	00:00:e5:21:67:ae	terry-leaf3	Eth1/34


Click the search icon in the **Endpoint Identifier** column to search for specific IP addresses.

Time	VRF	Endpoint Identifier	MAC	Switch Name	Port
11/07 06:25 AM	myvrf_50000	IPv4:192.60	00:00:e5:21:68:84	terry-leaf3	Eth1/34
11/07 06:25 AM	myvrf_50000	IPv4:192.60	00:00:e5:21:62:24	terry-leaf3	Eth1/34
11/07 06:25 AM	myvrf_50000	IPv4:192.60.2.101:30004	00:00:e5:21:66:86	terry-leaf3	Eth1/34
11/07 06:25 AM	myvrf_50000	IPv4:192.60.1.68:30004	00:00:e5:21:64:44	terry-leaf3	Eth1/34
11/07 06:25 AM	myvrf_50000	IPv4:192.60.2.249:30004	00:00:e5:21:67:ae	terry-leaf3	Eth1/34

In certain scenarios, the datapoint database may go out-of-sync and information, such as the number of endpoints, may not be displayed correctly due to network issues such as -

- Endpoint moves under the same switch between ports and the port information needs some time to be updated.
- An orphan endpoint is attached to the second VPC switch and is no longer an orphan endpoint.
- NX-API not enabled initially and then enabled at a later point in time.
- NX-API failing initially due to misconfiguration.
- Change in Route Reflector (RR).
- Management IPs of the switches are updated.

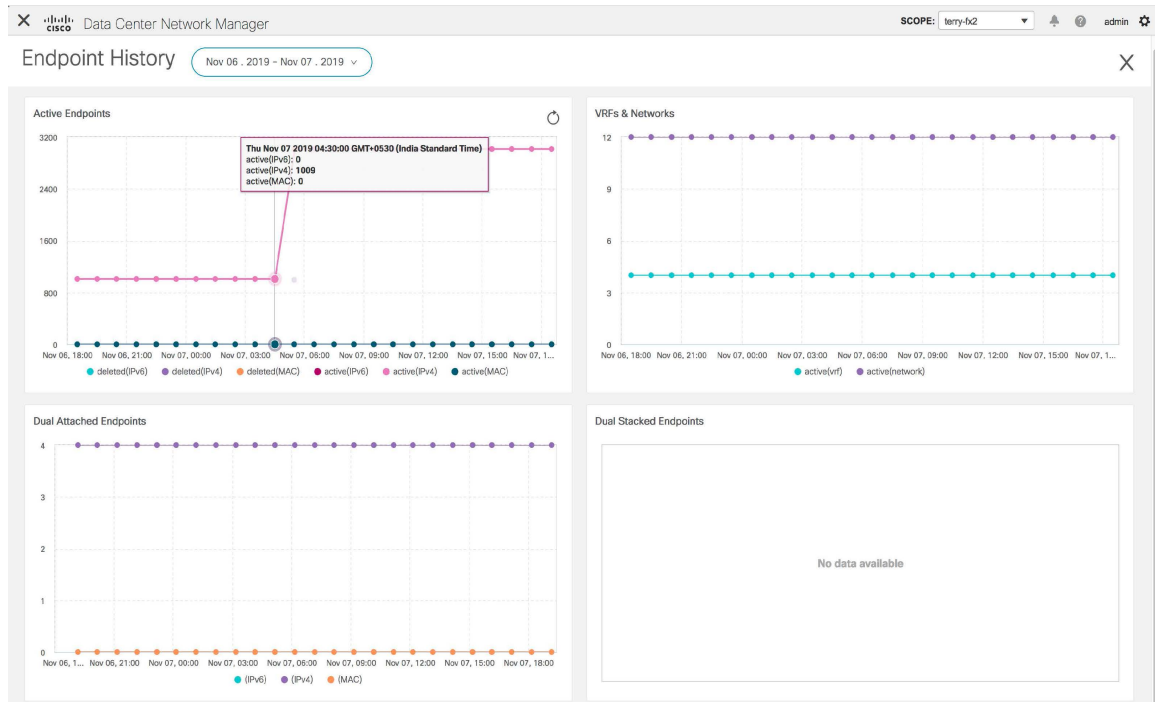
In such cases, clicking the **Resync**  icon leads to the dashboard syncing to the data currently in the RR. However, historical data is preserved. We recommend not clicking **Resync** multiple times as this is a compute-intensive activity.

Click the **Pause**  icon to temporarily stop the near real-time collection and display of data.

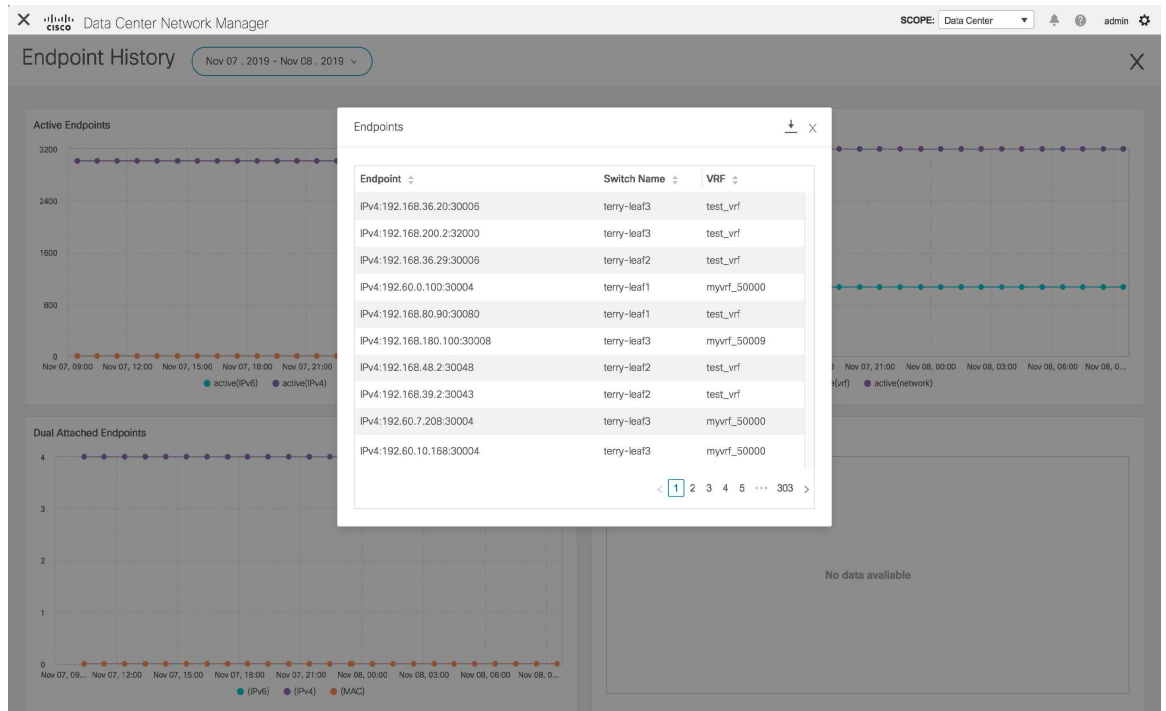
Consider a scenario in which EPL is first enabled and the **Process MAC-Only Advertisements** checkbox is selected. Then, EPL is disabled and enabled again without selecting the **Process MAC-Only Advertisements** checkbox. As the cache data in elasticsearch is not deleted on disabling of EPL, the MAC endpoint information is still displayed in the EPL dashboard. The same behavior is observed when a Route-Reflector is disconnected. Depending on the scale, the endpoints are deleted from the EPL dashboard after some time. In certain cases, it may take up to 30 minutes to remove the older MAC-only endpoints. However, to display the latest endpoint data, you can click the **Resync** icon at the top right of the EPL dashboard.

## Endpoint History

Click any tile in the top pane of the EPL dashboard to go to the **Endpoint History** window. A graph depicting the number of active endpoints, VRFs and networks, dual attached endpoints and dual stacked MAC endpoints at various points in time is displayed. The graphs that are displayed here depict all the endpoints and not only the endpoints that are present in the selected fabric. Endpoint history information is available for the last 180 days amounting to a maximum of 100 GB storage space.

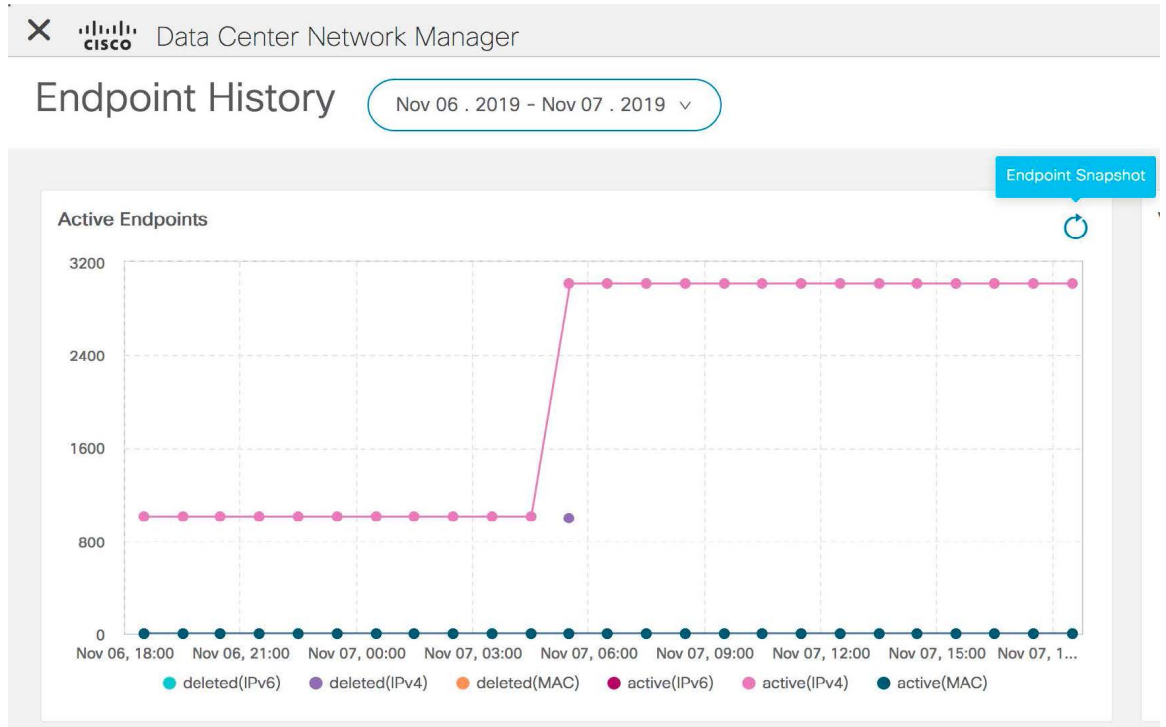


Hover over the graph at specific points to display more information. Click on each point in the graph to display detailed information at that point of time. You can display the graph for a specific requirement by clicking the color-coded points at the bottom of each graph. For example, click on all color-coded points other than **active (IPv4)** in the Active Endpoints window displayed above such that only **active (IPv4)** is highlighted and the other points are not highlighted. In such a scenario, only the active IPv4 endpoints are displayed on the graph. Click the Download icon at the top right of the **Endpoints** window to download the data as a CSV file.

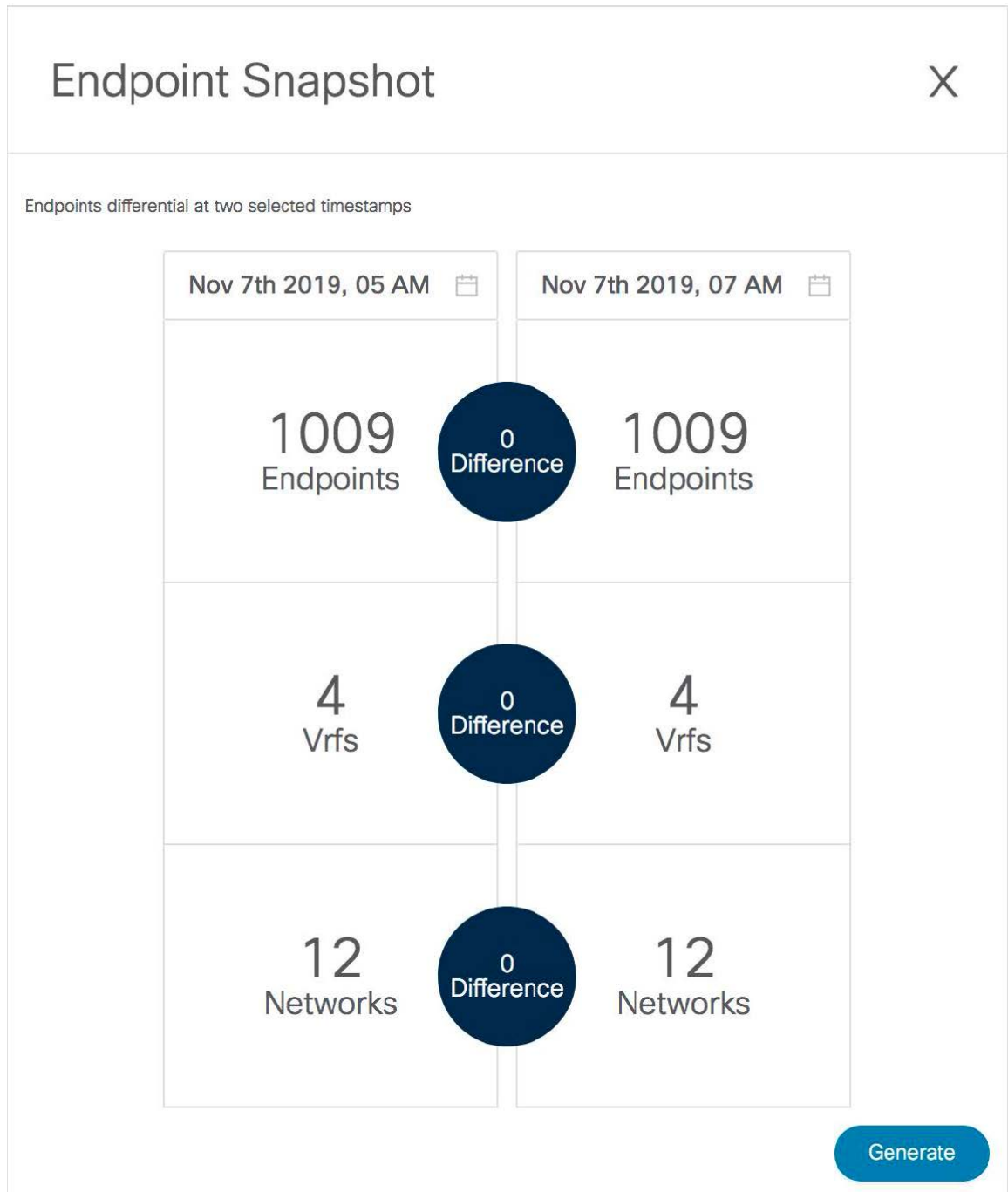


### Endpoint Snapshots

Starting from Cisco DCNM Release 11.3(1), you can compare endpoint data at two specific points in time. To display the **Endpoint Snapshot** window, click the **Endpoint Snapshot** icon at the top right of the **Active Endpoints** graph in the **Endpoint History** window.



By default, endpoint snapshot comparison data for the previous hour is displayed.



To compare endpoint snapshots at specific points in time, select two points in time, say T1 and T2, and click **Generate**.

# Endpoint Snapshot



Endpoints differential at two selected timestamps

Nov 7th 2019, 04 AM |
2019, 19 PM

<< < Nov 2019 > >>

Su	Mo	Tu	We	Th	Fr	Sa
27	28	29	30	31	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
1	2	3	4	5	6	7

Now
select time

Ok

12  
Networks

Difference

12  
Networks

Generate

A comparison of the endpoints, VRFs, and networks at the selected points in time are displayed. Click each tile to download more information about the endpoints, VRFs, or networks. Click the **Difference** icon to download details about the differences in data for the specified time interval. Snapshots are stored for a maximum of three months and then discarded.

## Endpoint Snapshot



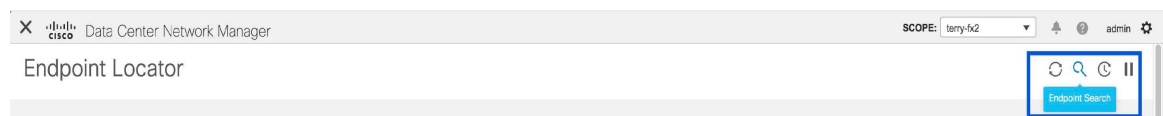
Endpoints differential at two selected timestamps



Generate

## Endpoint Search

Click the **Endpoint Search** icon at the top right of the Endpoint Locator landing page to view a real-time plot displaying endpoint events for the period specified in a date range.





You can search for a specific metric value in the search bar. Search is supported on any of the fields as specified under the **Available Fields** column from the menu on the left.

The screenshot shows the Cisco Data Center Network Manager Endpoint Search interface. The search bar contains the query "epl\_history.\*". The results are displayed in a table with columns for timestamp and source. The source column contains detailed endpoint information, including Fabric\_Id, IP, MAC, and OperationStatus. A bar chart above the table shows the count of hits over time, with a significant spike on November 7th, 2019.

Time	source
November 7th 2019, 06:25:54.251	Fabric_Id: 3:12vpn IP: 192.60.1.68 MAC: 00:00:e5:21:64:44 I2_VNI: 30,004 I3_VNI: 50,000 Switch_Name: terry-leaf3 Switch_Type: N9K Switch_IP: 192.168.126.154 Origin_IP: 10.2.0.5, 0.0.0.0, 0.0.0.0, 0.0.0.0 Switch_NextHop_IP: 10.3.0.5 Port: Eth1/34 VLAN: 60 I3_IW: 60 Operation: DELETE EndpointType: timestamp: November 7th 2019, 06:25:54.251 Seq_Num: 0 VRF: myvrf_50000 Br_Domain: 60 Cluster: Valid: 1 OperationStatus: RouteDistinguisher: 10.2.0.4:32827 EndpointIdentifier: IPv4:192.60.1.68:30004 IPVersion: IPv4 Id: AWSDFKahy yWxbCCzCoy _type: endpoint _index: epl_history_terry-fx2_2019_11_06 _score: -
November 7th 2019, 06:25:54.251	Fabric_Id: 3:12vpn IP: 192.60.0.52 MAC: 00:00:e5:21:62:24 I2_VNI: 30,004 I3_VNI: 50,000 Switch_Name: terry-leaf3 Switch_Type: N9K Switch_IP: 192.168.126.154 Origin_IP: 10.2.0.5, 0.0.0.0, 0.0.0.0, 0.0.0.0 Switch_NextHop_IP: 10.3.0.5 Port: Eth1/34 VLAN: 60 I3_IW: 60 Operation: DELETE EndpointType: timestamp: November 7th 2019, 06:25:54.251 Seq_Num: 0 VRF: myvrf_50000 Br_Domain: 60 Cluster: Valid: 1 OperationStatus: RouteDistinguisher: 10.2.0.4:32827 EndpointIdentifier: IPv4:192.60.0.52:30004 IPVersion: IPv4 Id: AWSDFKahy yWxbCCzCoy _type: endpoint _index: epl_history_terry-fx2_2019_11_06 _score: -
November 7th 2019, 06:25:54.251	Fabric_Id: 3:12vpn IP: 192.60.2.101 MAC: 00:00:e5:21:66:86 I2_VNI: 30,004 I3_VNI: 50,000 Switch_Name: terry-leaf3 Switch_Type: N9K Switch_IP: 192.168.126.154 Origin_IP: 10.2.0.5, 0.0.0.0, 0.0.0.0, 0.0.0.0 Switch_NextHop_IP: 10.3.0.5 Port: Eth1/34 VLAN: 60 I3_IW: 60 Operation: DELETE EndpointType: timestamp: November 7th 2019, 06:25:54.251 Seq_Num: 0 VRF: myvrf_50000 Br_Domain: 60 Cluster: Valid: 1 OperationStatus: RouteDistinguisher: 10.2.0.4:32827 EndpointIdentifier: IPv4:192.60.2.101:30004 IPVersion: IPv4 Id: AWSDFKahy yWxbCCzCoy _type: endpoint _index: epl_history_terry-fx2_2019_11_06 _score: -
November 7th 2019, 06:25:54.251	Fabric_Id: 3:12vpn IP: 192.60.3.100 MAC: 00:00:e5:21:68:84 I2_VNI: 30,004 I3_VNI: 50,000 Switch_Name: terry-leaf3 Switch_Type: N9K Switch_IP: 192.168.126.154 Origin_IP: 10.2.0.5, 0.0.0.0, 0.0.0.0, 0.0.0.0 Switch_NextHop_IP: 10.3.0.5 Port: Eth1/34 VLAN: 60 I3_IW: 60 Operation: DELETE EndpointType: timestamp: November 7th 2019, 06:25:54.251 Seq_Num: 0 VRF: myvrf_50000 Br_Domain: 60 Cluster: Valid: 1 OperationStatus: RouteDistinguisher: 10.2.0.4:32827 EndpointIdentifier: IPv4:192.60.3.100:30004 IPVersion: IPv4 Id: AWSDFKahy yWxbCCzCoy _type: endpoint _index: epl_history_terry-fx2_2019_11_06 _score: -

## Endpoint Life

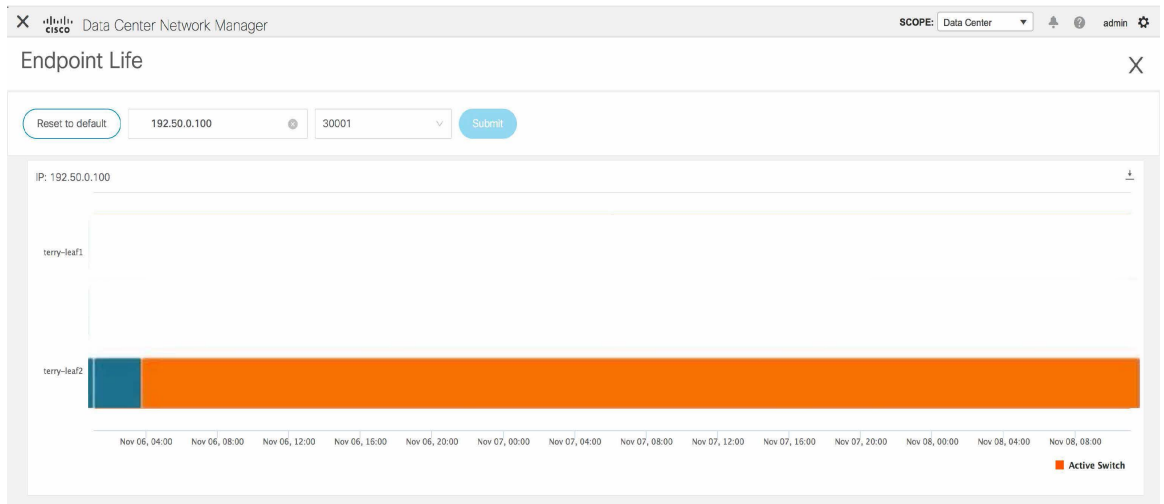
Click the **Endpoint Life** icon at the top right of the Endpoint Locator landing page to display a time line of a particular endpoint in its entire existence within the fabric.

The screenshot shows the Cisco Data Center Network Manager Endpoint Locator interface. The search bar contains the query "terry-fx2". An "Endpoint Life" icon is visible in the top right corner of the interface.

Specify the IP or MAC address of an endpoint and the VXLAN Network Identifier (VNI) to display the list of switches that an endpoint was present under, including the associated start and end dates. Click **Submit**.

The screenshot shows the Cisco Data Center Network Manager Endpoint Life interface. The search bar contains the query "terry-fx2". The interface shows a search bar with "Enter IP or MAC" and "Select VNI" fields, and a "Submit" button. Below the search bar, there is a message: "Please enter IP & VNI to see the graph".

The window that is displayed is essentially the endpoint life of a specific endpoint. The bar that is orange in color represents the active endpoint on that switch. If the endpoint is viewed as active by the network, it will have a band here. If an endpoint is dual-homed, then there will be two horizontal bands reporting the endpoint existence, one band for each switch (typically the vPC pair of switches). In case the endpoints are deleted or moved, you can also see the historical endpoint deletions and moves on this window.





## CHAPTER 2

# DCNM Integration with ServiceNow

---

- [DCNM Integration with ServiceNow, on page 35](#)

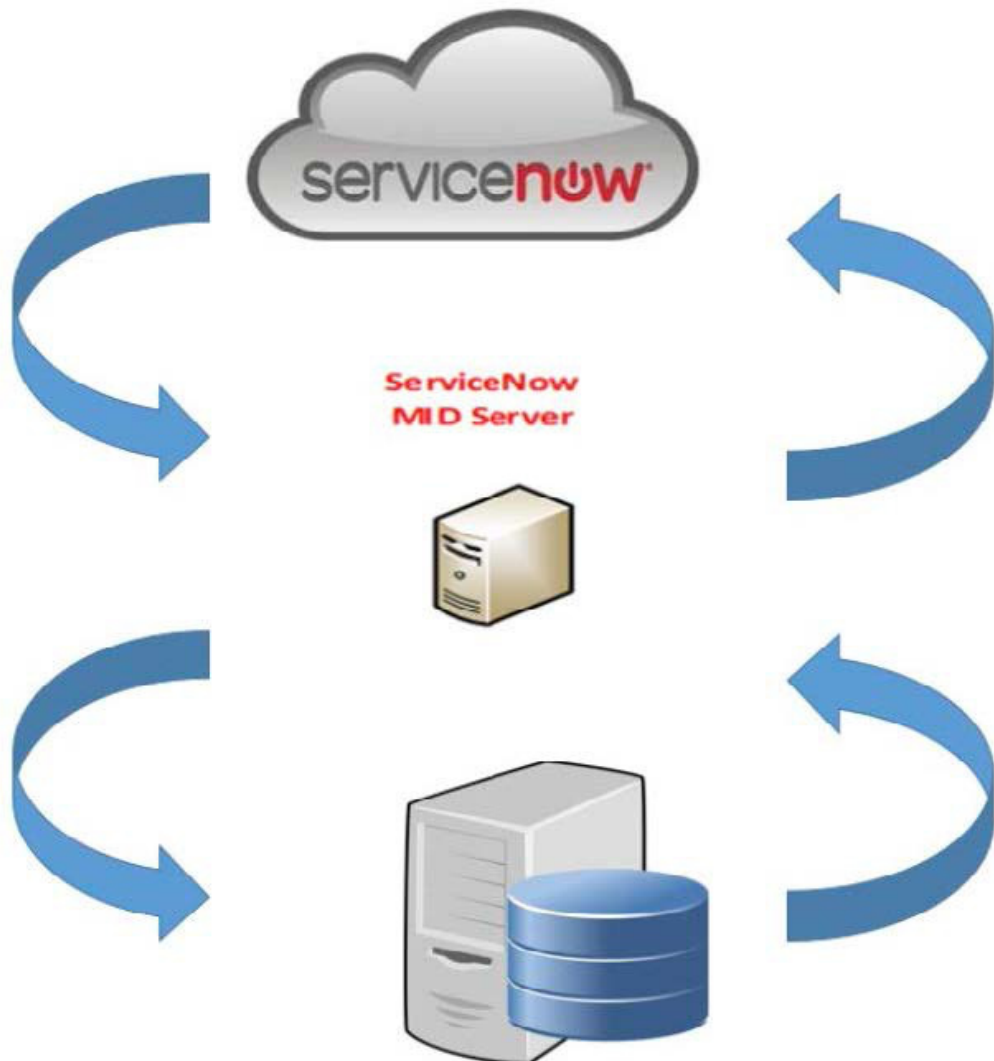
## DCNM Integration with ServiceNow

ServiceNow offers applications for IT Service Management (ITSM) and IT Operations Management (ITOM). There are four primary modules - inventory discovery, incident management, event management & change management workflows. Starting from Cisco DCNM Release 11.3(1), we provide Cisco DCNM integration with ServiceNow. This enables you to integrate end-user IT data with the ServiceNow platform. The integration provides a default set of ServiceNow custom tables which are populated with configuration data.

To utilize this functionality, install the DCNM application in the ServiceNow customer instance and provide the DCNM mid-server details. Information or data regarding switch details, port details, and alarms, is retrieved to the ServiceNow Configuration Management Database (CMDB) tables. By default, data is retrieved every 15 minutes and displayed.

Details about the switches and ports of each switch are collected from the DCNM inventory. The alarms are collected by polling DCNM. Alarms are then filtered and categorized based on their type, such as, CPU, MEMORY, POWER, LINKSTATE, EXTERNAL, ICMP, SNMP, and SSH. The alarms are then stored in an Events table. These events are then used to generate incidents for the CPU, MEMORY, SNMP, and SSH categories. The source, description, severity and category of each alarm is stored. When an alarm is cleared on DCNM, it is also cleared on ServiceNow in the next poll cycle. When polling of alarms is initiated for the first time, the alarms that were raised in the last seven days are pulled in from DCNM. In case there is a gap of more than seven days between collection of alarms, the old alarms are cleared and the polling process is reinitiated.

The DCNM application on ServiceNow runs scheduled scripts and connects with the mid-server which in turn connects with DCNM to retrieve data. DCNM sends the requested data to the mid-server which then passes on the data to the DCNM application on ServiceNow. The tables in the DCNM instance on ServiceNow are then populated with this retrieved data.



You can also watch the video that demonstrates Cisco DCNM integration with ServiceNow. See [Cisco DCNM Integration with ServiceNow](#).

## Guidelines and Limitations of DCNM Integration with ServiceNow

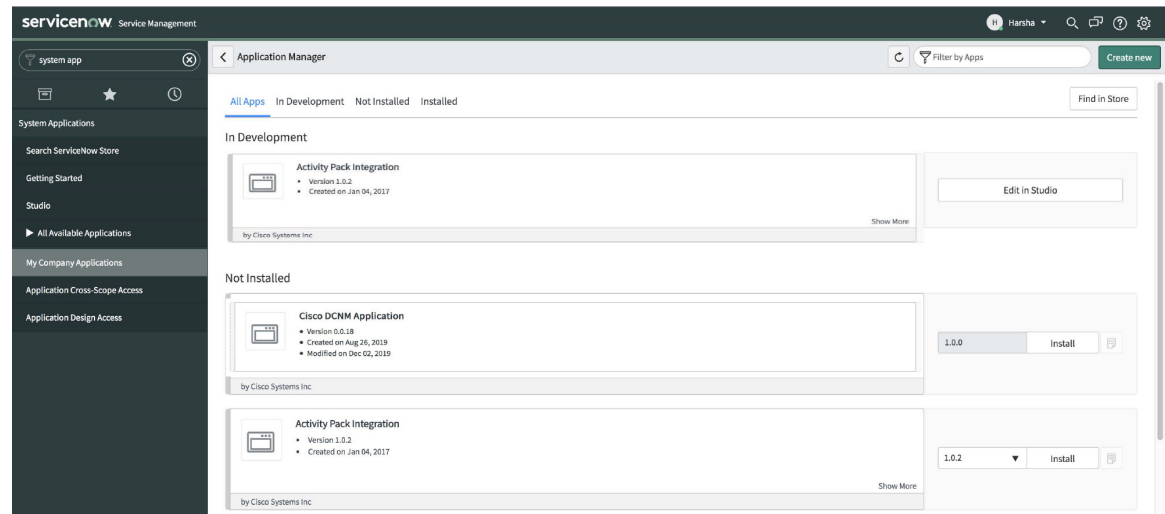
- Details about only one mid-server should be added in the **Cisco DCNM>Properties** table.
- Scheduled scripts to retrieve data are run only after insertion of a server record in the **Cisco DCNM>Properties** table.
- In case the mid-server IP Address and credentials in the **Cisco DCNM>Properties** table are changed, the data that was imported using the previous mid-server is deleted from the application scope tables. However, data that was imported to the ServiceNow CMDB (global scope) remains and is not deleted.
- To ensure optimal performance in the ServiceNow database, each entry is matched with the switch database ID and IP Address ensuring that there is no duplication of entries.

- Entries in the `cmdb_ci_ip_switch` table have to be manually deleted in case a new server is added in the **Cisco DCNM>Properties** table.

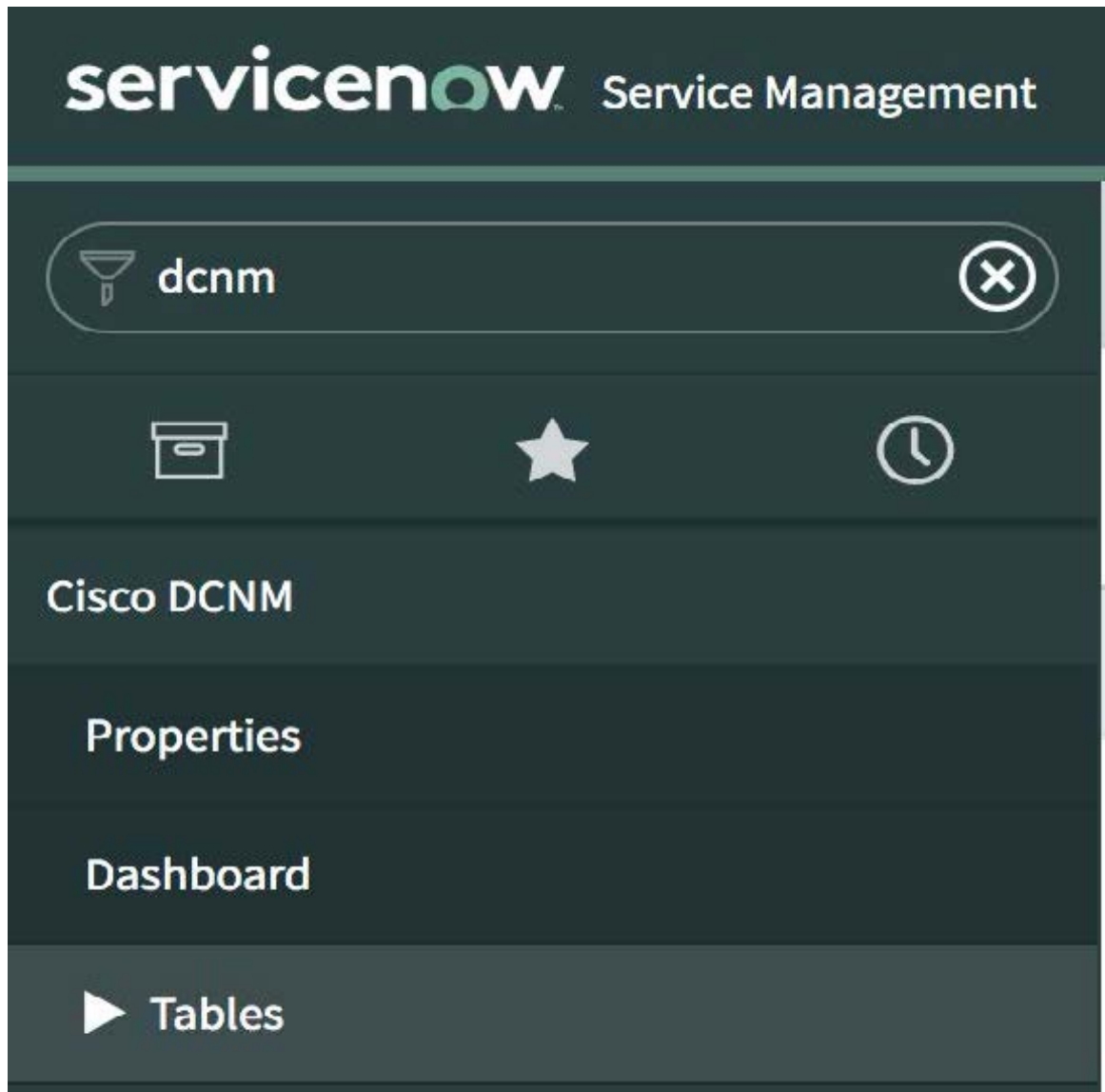
## Installing and Configuring the Cisco DCNM Application on ServiceNow

### Procedure

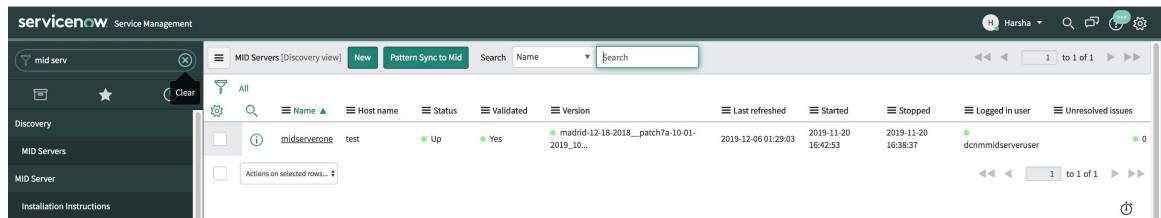
- Step 1** Log in to <https://dcnm1.service-now.com>. Select **System Applications > Applications**. Install the **Cisco DCNM Application** from the **All Apps** tab.



- Step 2** After installation is complete, verify that the Cisco DCNM Properties and Dashboard tabs are appearing in the application.



**Step 3** Choose **MID Servers** and click the MID Server that is used for DCNM integration.



**Step 4** Scroll down and click the **Properties** tab. Click **New** and add the property given below in the **MID Server Property New record** window. Click **Submit**.

Name	Type	Value
glide.http.outbound.max_timeout.enabled	True/false	False

The screenshot shows the 'MID Server Property' configuration form in ServiceNow. The form is titled 'MID Server Property' and is a 'New record'. A blue banner at the top states: 'MID Server Properties allow administrators to configure a MID Server with additional configuration parameters to alter any default behavior. [More Info](#)'. The form contains the following fields:

- Application: Global
- Name: glide.http.outbound.max\_timeout.enabled
- Value: false
- MID server: midserverone

A 'Submit' button is located at the bottom left of the form.

**Step 5** Now, select the **Configuration Parameters** tab.

The screenshot shows the 'MID Server' configuration page in ServiceNow, specifically the 'Configuration Parameters' tab. The page title is 'MID Server' and the record is 'midserverone'. The 'Configuration Parameters' tab is active, showing a table of parameters:

Parameter name	Value
mid_proxy_use_proxy	true
url	https://dcnm1.service-now.com/
mid_proxy_port	80
mid_instance_username	dcnm1midserveruser

A 'New' button is visible at the top of the table.

**Step 6** In the **Configuration Parameters** tab, click **New**. Enter the required details in the fields.

The screenshot shows the 'MID Server Configuration Parameter' form in ServiceNow. The form is titled 'MID Server Configuration Parameter' and is a 'New record'. The form contains the following fields:

- MID server: midserverone
- Parameter name: mid.disable\_amb (Disable the AMB Client on the MID Server. Default: false)
- Domain: global
- Value: true

A 'Submit' button is located at the bottom left of the form.

**Step 7** Click **Submit** to set up the MID Server.

**Step 8** Choose **Cisco DCNM > Properties**. Click **New Server**. Enter the required parameters.

The screenshot shows the 'DCNM Properties' configuration form in ServiceNow. The form is titled 'DCNM Properties' and is a 'New record'. A blue banner at the top states: 'Ensure DCNM is NTP time sync'. The form contains the following fields:

- DCNM IP Address: 172.28.11.96
- Username: admin
- Password: [masked]
- Mid Server: midserverone
- MidServer Status: Up
- DCNM Connection Status: Reachable

Below the main fields, there is a section for 'Incident Creation from the DCNM Alarms' with a 'Create Incident' checkbox checked and a 'User' dropdown set to 'Cisco DCNM'. An 'Update' button is located at the bottom left of the form.

DCNM IP Address - IP Address of the DCNM.

Username - Enter the username used to log in to DCNM.

Password - Enter the password used to log in to DCNM.

**Note** Access should be provided only for DCNM admins.

Mid server - Specify the name of the mid server to be used. The name is auto-populated as you type. You can also click the search icon next to this field to bring the MID Servers window. You can then select a MID Server from the list that is displayed.

User - Create a new user and add the user name in this field. The Caller field in the incidents that are created is populated with this user name. This field is auto-populated as you type. You can also click the search icon next to this field to bring the Users window. You can then select a user from the list that is displayed.

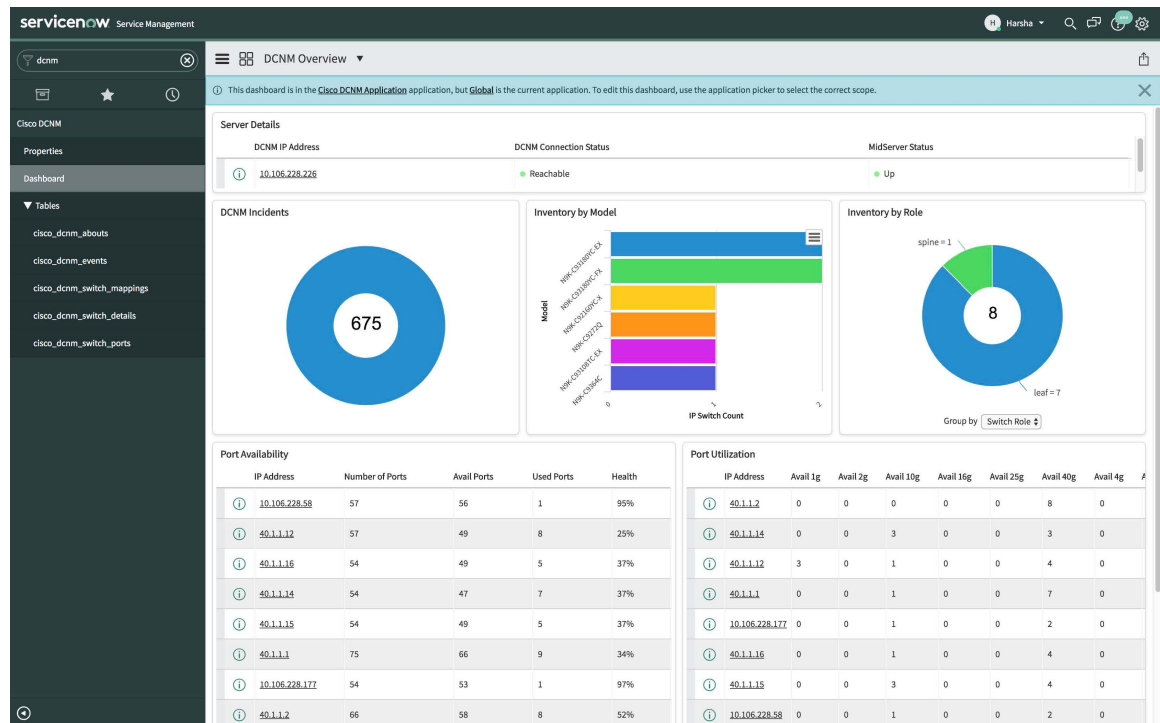
Create Incident - Select this checkbox in case you need incidents to be raised automatically for alarm events.

Now, click **Submit**.

After the server details are submitted, the **DCNM Connection Status** field will display **Reachable** on successful communication with DCNM, and **Unreachable**, in case the connection is unsuccessful.

## Viewing the Dashboard

Choose **Cisco DCNM>Dashboard** to display the dashboard. The **DCNM IP Address**, the **DCNM Connection Status** and the **MidServer Status** are displayed at the top of the dashboard.



Click **All** to retrieve and display data from all the DCNM Servers that are displayed in the dropdown list. When the **All** option is selected, the number of incidents that are displayed in the DCNM Incidents donut are color-coded and displayed based on the different DCNM server IP addresses. The Inventory by Model and



Inventory by Role donuts also display data from all the DCNM servers. The Port Availability and Port Utilization donuts display data along with the DCNM Server that each IP address belongs to.

**DCNM Incidents** - This displays the number of incidents that have been raised based on the alarms retrieved from DCNM. Click the donut for more details about the

Number	Opened	Short description	Caller	Priority	State	Category	Assignment group	Assigned to	Updated	Updated by
INC0010677	2019-12-04 19:45:09	DCNM Server Alert	Cisco DCNM	2 - High	New	Inquiry / Help	(empty)	(empty)	2019-12-04 19:45:09	system
INC0010676	2019-12-02 00:10:10	DCNM Server Alert	Cisco DCNM	2 - High	New	Inquiry / Help	(empty)	(empty)	2019-12-02 00:10:10	system
INC0010675	2019-12-02 00:10:10	DCNM Server Alert	Cisco DCNM	2 - High	New	Inquiry / Help	(empty)	(empty)	2019-12-02 00:10:10	system
INC0010674	2019-12-02 00:10:10	DCNM Server Alert	Cisco DCNM	2 - High	New	Inquiry / Help	(empty)	(empty)	2019-12-02 00:10:10	system

**Inventory by Model** - This displays the number and type of switches present in DCNM. Each band represents a device model. Click a band for more

Name	Manufacturer	Model ID	IP Address	Serial number	Can partition VLANs	Can route IP	Can switch IP
93180YC-EX-leaf5	(empty)	Unknown	40.1.1.15	FDO210705Q6	false	false	false
Leaf1-93180YC-EX_Sender	(empty)	Unknown	10.106.228.58	FDO22400W0D	false	false	false

**Inventory by Role** - This displays the number and types of switch roles present in DCNM. Click the required section to display the number of roles that are operational and click on that pictorial representation to display more details about the roles.

IP Address	Switch Role	Fabric	License Detail	Operational Status
10.106.228.58	leaf	Default_LAN	Honor	Operational
40.1.1.12	leaf	Default_LAN	Permanent	Operational
40.1.1.16	leaf	Default_LAN	Permanent	Operational
40.1.1.14	leaf	Default_LAN	Honor	Operational

**Port Availability** - This displays information about port availability. The IP address along with the total number of ports, available ports, used ports and health of the switch is displayed. Click an IP address to display more

Number of Ports	57	Peer	
Switch DB ID	14060	Peer Switch DB ID	0
Avail Ports	56	Switch Role	leaf
Health	95%	Used Ports	1
License Detail	Honor	VPC Domain	0
IP Address	10.106.228.58		

**Port Utilization** - This displays information about port utilization based on each IP address. The number of ports having 1G, 2G, 4G, 8G, 10G, 16G, 25G, 32G, 40G, and 100G availability, are displayed. Click an IP

address to display more

cisco_dcnm_switch_port		2120	
Switch DB ID	2120		
Avall 10g	0	Avall 16g	0
Avall 1g	0	Avall 25g	0
Avall 2g	0	Avall 32g	0
Avall 4g	0	Avall 40g	8
Avall 8g	0	Avall na	0
Avall 100g	0	Health	53%

## Troubleshooting DCNM Integration with ServiceNow

In case data is not being retrieved in the ServiceNow table:

- Check if the MID server is up or down.
- Check for information entries in system logs with the source “x\_caci\_cisco\_dcnm”.
- Check the login credentials added in Cisco DCNM Properties.

For more information on DCNM application integration with ServiceNow, [click here](#).



## CHAPTER 3

# Template Usage in Cisco DCNM LAN Fabric Deployment

templateType	Specifies the type of Template used.	<ul style="list-style-type: none"><li>• CLI</li><li>• POLICY</li><li>• SHOW</li><li>• PROFILE</li><li>• ABSTRACT</li></ul>
--------------	--------------------------------------	--

- [Policy Template, on page 43](#)
- [Fabric Template, on page 47](#)
- [Profile Template, on page 47](#)
- [Viewing, Editing, and Adding Policies, on page 48](#)
- [Deploying New Configurations, on page 52](#)
- [switch\\_freeform Template Usage, on page 53](#)
- [Changing the Contents of a Template in Use, on page 56](#)

## Policy Template

For the policy template, there are two template content types: CLI and PYTHON. With CLI content type, the policy templates are parameterized CLI templates. They can have a lot of variables and CLIs. Typically, CLI policy templates are small and do not have any if-else-for etc. like constructs. An example CLI policy template for AAA server configuration is shown below:

The screenshot shows the Cisco Data Center Network Manager interface. The top navigation bar includes the Cisco logo, the text 'Data Center Network Manager', and a user profile 'admin'. Below this, the breadcrumb 'Control / Template Library' is visible. The main content area is titled 'Template Content: [edit icon]' and shows a code editor for a template named 'aaa\_radius'. The code is as follows:

```


1  ##template variables
2
3  # Copyright (c) 2018 by Cisco Systems, Inc.
4  # All rights reserved.
5
6  @(DisplayName="AAA Server Name/IP", Description="Name or IPv4/IPv6 Address of an AAA Server")
7  ipAddressWithoutPrefix AAA_SERVER;
8
9  @(DisplayName="AAA group", Description="Name of AAA Group")
10 string AAA_GROUP {
11     minLength = 1;
12     maxLength = 127;
13 };
14
15 ##
16 ##template content
17
18 aaa group server radius $$AAA_GROUP$$
19 server $$AAA_SERVER$$
20
21 ##






```

At the top right of the code editor, there is a status bar that reads '0 Errors, 0 Warnings' along with icons for refresh, save, share, settings, and back.

But you can also have policy templates of template content type PYTHON. Essentially, this allows multiple CLI policy templates to be combined together with a common “source” so that they get all applied/un-applied at one go. For example, when you want to create a vPC host port, it has to be created symmetrically on both peers that are part of the vPC pair. In addition, you have to create port-channel, member interfaces, channel-group, etc. This is why a python vPC host policy template has been added. An example interface PYTHON template for setting up a routed interface is shown below:

Control / Template Library

Template Content: 

int\_routed\_host\_11\_1 0 Errors, 0 Warnings     

```

1  ##template variables
2
3  # Copyright (c) 2018 by Cisco Systems, Inc.
4  # All rights reserved.
5
6  @(IsInternal=true)
7  string SERIAL_NUMBER;
8
9  @(PrimaryAssociation=true, IsInternal=true)
10 interface INTF_NAME;
11
12 @(IsMandatory=false, DisplayName="Interface VRF", Description="Interface VRF name, default VRF if not specified")
13 string INTF_VRF {
14     minLength = 1;
15     maxLength = 32;
16 };
17
18 @(IsMandatory=false, DisplayName="Interface IP", Description="IP address of the interface")
19 ipv4Address IP;
20
21 @(IsMandatory="IP!=null", DisplayName="IP Netmask Length", Description="IP netmask length used with the IP address (Min:1, Max:31)")
22 integer PREFIX {
23     min = 1;
24     max = 31;
25 };
26
27 @(IsMandatory=false, DisplayName="Routing TAG", Description="Routing tag associated with interface IP")
28 string ROUTING_TAG;
29
30 @(DisplayName="MTU", IsMTU=true, Description="MTU for the interface (Min:576, Max:9216)")
31 integer MTU {
32     min = 576;
33     max = 9216;
34     defaultValue=9216;
35 };
36
37 @(DisplayName="SPEED", Description="Interface Speed")
38 enum SPEED {
39     validValues=Auto,100Mb,1Gb,10Gb,25Gb,40Gb,100Gb;
40     defaultValue=Auto;
41 };
42
43 @(IsMandatory=false, DisplayName="Interface Description", Description="Add description to the interface (Max Size 254)")
44 string DESC {
45     minLength = 1;
46     maxLength = 254;
47 };
48
49 @(IsMandatory=false, IsMultiLineString=true, DisplayName="Freeform Config", Description="Additional CLI for the interface")
50 string CONF;
51
52 @(DisplayName="Enable Interface", Description="Uncheck to disable the interface")
53 boolean ADMIN_STATE {
54     defaultValue=true;
55 };
56
57 ##
58 ##template content
59
60 from com.cisco.dcbu.vincil.rest.services.jython import PTIWrapper
61 from com.cisco.dcbu.vincil.rest.services.jython import Wrapper
62 from com.cisco.dcbu.vincil.rest.services.jython import WrappersResp
63 from utility import *
64
65 def add():
66     try:
67         if CONF != "":
68             respObj, conf = Util.adjustIntfFreeformConfig(SERIAL_NUMBER, INTF_NAME, CONF)
69             if respObj.isRetCodeFailure():
70                 return respObj
71
72     # modify to be done, calling delete now to clean up PTIs before add
73     delete()
74
75     intfVrf = "default"
76     try:
77         if INTF_VRF != "":
78             intfVrf = INTF_VRF
79     except:
80         Wrapper.print("Switch/Intf = [%s/%s] - Template[int_routed_host_11_1]: INTF_VRF not defined" %
81                     (SERIAL_NUMBER, INTF_NAME))
82         pass
83
84     routingTag = ""
85     try:
86         if ROUTING_TAG != "":
87             routingTag = ROUTING_TAG
88     except:
89         Wrapper.print("Switch/Intf = [%s/%s] - Template[int_routed_host_11_1]: ROUTING_TAG not defined" %
90                     (SERIAL_NUMBER, INTF_NAME))
91         pass
92
93     # routed_interface has only one CLI command: no switchport
94     # It must be configured before interface_vrf
95     # p2p_routed_interface that configures the IP address must come after interface_vrf
96     Util.exe(PTIWrapper.createOrUpdate(SERIAL_NUMBER, "INTERFACE",
97                                     INTF_NAME, INTF_NAME,
98                                     ConfigPriority.CONFIG_PRIO_INTF,
99                                     "routed_interface",
100                                     {"INTF_NAME": INTF_NAME}))
101
102     if intfVrf != "default":
103         # Create/Update PTI for interface VRF
104         Util.exe(PTIWrapper.createOrUpdate(SERIAL_NUMBER, "INTERFACE",
105                                         INTF_NAME, INTF_NAME,
106                                         ConfigPriority.CONFIG_PRIO_INTF_SUB_LVL1,
107                                         "interface_vrf",
108                                         {"INTF_NAME": INTF_NAME, "INTF_VRF": intfVrf}))
109
110     if IP != "":
111         if routingTag == "":
112             Util.exe(PTIWrapper.createOrUpdate(SERIAL_NUMBER, "INTERFACE",
113                                               INTF_NAME, INTF_NAME,
114                                               ConfigPriority.CONFIG_PRIO_INTF_SUB_LVL2,
115                                               "p2p_routed_interface",
116                                               {"INTF_NAME": INTF_NAME, "IP": IP, "PREFIX": PREFIX}))

```

Each policy template has a template subtype like DEVICE, INTERFACE, etc. This allows the right policy template to appear at the right selection point. For example, in the Interface window, you will only see the interface policy templates.

Name	Supported Platforms	Tags	Template ...	Template ...	Published	Modified T...	D...
csr1kv_loopback	CSR1KV	[interface_...]	POLICY	INTERFAC...	false	2019-06-03...	
epf_routed_intf	N9K	[interface_...]	POLICY	INTERFAC...	false	2019-06-03...	
GigabitEthernet	CSR1KV	[interface_...]	POLICY	INTERFAC...	false	2019-06-03...	
GigabitEthernet_freeform	CSR1KV	[interface_...]	POLICY	INTERFAC...	false	2019-06-03...	
int_access_host_11_1	All	[interface_...]	POLICY	INTERFAC...	false	2019-06-03...	
int_loopback_11_1	All	[interface_...]	POLICY	INTERFAC...	false	2019-06-03...	
int_mgmt_11_1	N9K	[interface_...]	POLICY	INTERFAC...	false	2019-06-03...	
int_monitor_ethernet_11_1	N9K	[interface_...]	POLICY	INTERFAC...	false	2019-06-03...	
int_monitor_port_channel_11_1	N9K	[interface_...]	POLICY	INTERFAC...	false	2019-06-03...	
int_port_channel_access_host_11_1	All	[interface_...]	POLICY	INTERFAC...	false	2019-06-03...	
int_port_channel_trunk_host_11_1	All	[interface_...]	POLICY	INTERFAC...	false	2019-06-03...	
int_routed_host_11_1	All	[interface_...]	POLICY	INTERFAC...	false	2019-06-03...	
int_subif_11_1	All	[interface_...]	POLICY	INTERFAC...	false	2019-06-03...	
int_trunk_host_11_1	All	[interface_...]	POLICY	INTERFAC...	false	2019-06-03...	
int_vpc_access_host_11_1	All	[interface_...]	POLICY	INTERFAC...	false	2019-06-03...	
int_vpc_trunk_host_11_1	All	[interface_...]	POLICY	INTERFAC...	false	2019-06-03...	

In the View/Edit Policies window on the Fabric Builder, you will only see device policy templates.

Name	Supported Platforms	Tags	Template ...	Template ...	Published	Modified T...	D...
aaa_radius	N9K		POLICY	DEVICE	false	2019-06-03...	
aaa_radius_deadtime	N9K		POLICY	DEVICE	false	2019-06-03...	
aaa_radius_key	N9K		POLICY	DEVICE	false	2019-06-03...	
aaa_radius_src_interface	N9K		POLICY	DEVICE	false	2019-06-03...	
aaa_radius_use_vrf	N9K		POLICY	DEVICE	false	2019-06-03...	
aaa_tacacs	N9K		POLICY	DEVICE	false	2019-06-03...	
aaa_tacacs_key	N9K		POLICY	DEVICE	false	2019-06-03...	
aaa_tacacs_src_interface	N9K		POLICY	DEVICE	false	2019-06-03...	
aaa_tacacs_use_vrf	N9K		POLICY	DEVICE	false	2019-06-03...	
anycast_gateway	N9K		POLICY	DEVICE	false	2019-06-03...	
anycast_rp	N9K		POLICY	DEVICE	false	2019-06-03...	
azure_network_selector	CSR1KV		POLICY	DEVICE	false	2019-06-03...	
banner	N9K		POLICY	DEVICE	false	2019-06-03...	
base_aaa	N9K		POLICY	DEVICE	false	2019-06-03...	
base_bgp	N9K		POLICY	DEVICE	false	2019-06-03...	
base_bgp_external	N9K,N7K		POLICY	DEVICE	false	2019-06-03...	
base_dhcp	N9K		POLICY	DEVICE	false	2019-06-03...	

You can make a copy of any of these templates and customize them as per their needs. That is the typical use-case for customization. **Do not** modify existing policies but make a copy, and then customize as per the requirements. Otherwise, after a DCNM upgrade, the changes may be lost.

In general, a template already in use, meaning one that is already applied to some switch within any fabric, cannot be edited.



**Note** No Type-CLI templates are used in the LAN fabric installation mode. They are all replaced with more powerful Policy templates which are a super set.

## Fabric Template

A fabric template is basically a python template, specifically jython, which is java + python. A fabric template is quite comprehensive, and in that it embeds the rules that are required for deploying a fabric, including all the logic required to generate intended configuration of all switches within the entire fabric. Configuration is generated based on published Cisco best practice guidelines. In addition to the embedded rules, the fabric template also integrates with other entities such as resource manager, topology database, device roles, configuration compliance, etc. and generates the configuration accordingly for all the devices in the fabric. This is the inherent part of the DCNM fabric builder.

The expectation is that users will not create their own fabric templates. DCNM provides a few fabric templates out of the box such as Easy Fabric, External Fabric, MSD Fabric, eBGP Fabric (introduced in DCNM 11.2).

Name	Supported Platforms	Tags	Template ...	Template ...	Published	Modified T...	D...
Easy_Fabric_11_1	All		FABRIC	NA	false	2019-06-03...	F...
Easy_Fabric_eBGP	All		FABRIC	NA	false	2019-06-03...	F...
External_Fabric_11_1	All		FABRIC	NA	false	2019-06-03...	F...
MSD_Fabric_11_1	All		FABRIC	NA	false	2019-06-03...	F...

## Profile Template

A profile template is used for provisioning of overlays (networks or VRFs). The idea is that when you apply some overlay configuration, there are multiple pieces of configurations that should go together. For example, valid layer-3 network configuration in a VXLAN EVPN fabric requires VLAN, SVI, int nve config, EVPN route-target, etc. All of these pieces are put together into what is called a configuration profile (NX-OS construct) and then effectively applied at one go. Either the whole configuration profile gets applied or nothing gets applied, on the switch. In this way, you are not left with any dangling or stray configurations on the switches. For any kind of overlay configurations, whether it is on the leaf or on the borders, DCNM employs profile templates.

There are four kinds of profile templates that are distinguished with tags as depicted below:

- Network Profile (applied to all devices with role leaf)
- Network Extension Profile (applied to all devices with role 'border\*')



- VRF Profile (applied to all devices with role leaf)
- VRF Extension Profile (applied to all devices with role ‘border\*’)

The screenshot shows the Cisco Data Center Network Manager (DCNM) interface. The breadcrumb navigation is 'Control / Template Library'. The main content area is titled 'Templates' and shows a table of available templates. The table has the following columns: Name, Supported Platforms, Tags, Template Type, Template Name, Published, Modified Time, and a 'D...' column. The table lists several templates, including 'base\_external\_router', 'Default\_Network\_Extension\_Universal', 'Default\_Network\_Universal', 'Default\_VRF\_Extension\_Universal', 'Default\_VRF\_Universal', 'ext\_base\_setup', 'ext\_fabric\_intf', 'ext\_fabric\_multisite\_intf\_11\_1', 'ext\_multisite\_overlay\_setup\_11\_1', 'ext\_multisite\_rs\_base\_feature', and 'ext\_multisite\_rs\_base\_setup'.

Name	Supported Platforms	Tags	Template ...	Template ...	Published	Modified T...	D...
base_external_router	N9K		PROFILE	NA	false	2019-06-03...	s...
Default_Network_Extension_Universal	All	[networkEx...	PROFILE	VXLAN	false	2019-06-03...	D...
Default_Network_Universal	All	[network]	PROFILE	VXLAN	false	2019-06-03...	D...
Default_VRF_Extension_Universal	All	[vrfExtension]	PROFILE	VXLAN	false	2019-06-03...	D...
Default_VRF_Universal	All	[vrf]	PROFILE	VXLAN	false	2019-06-03...	D...
ext_base_setup	All	[borderBase]	PROFILE	VXLAN	false	2019-06-03...	
ext_fabric_intf	All		PROFILE	VXLAN	false	2019-06-03...	
ext_fabric_multisite_intf_11_1	All		PROFILE	VXLAN	false	2019-06-03...	
ext_multisite_overlay_setup_11_1	All	[multiSiteO...	PROFILE	VXLAN	false	2019-06-03...	
ext_multisite_rs_base_feature	N9K,N7K	[multiSiteO...	PROFILE	VXLAN	false	2019-06-03...	s...
ext_multisite_rs_base_setup	N9K	[multiSiteO...	PROFILE	VXLAN	false	2019-06-03...	s...

For more information about how to apply overlay configuration via the Networks & VRFs workflow in DCNM, see *Creating and Deploying Networks and VRFs* section.

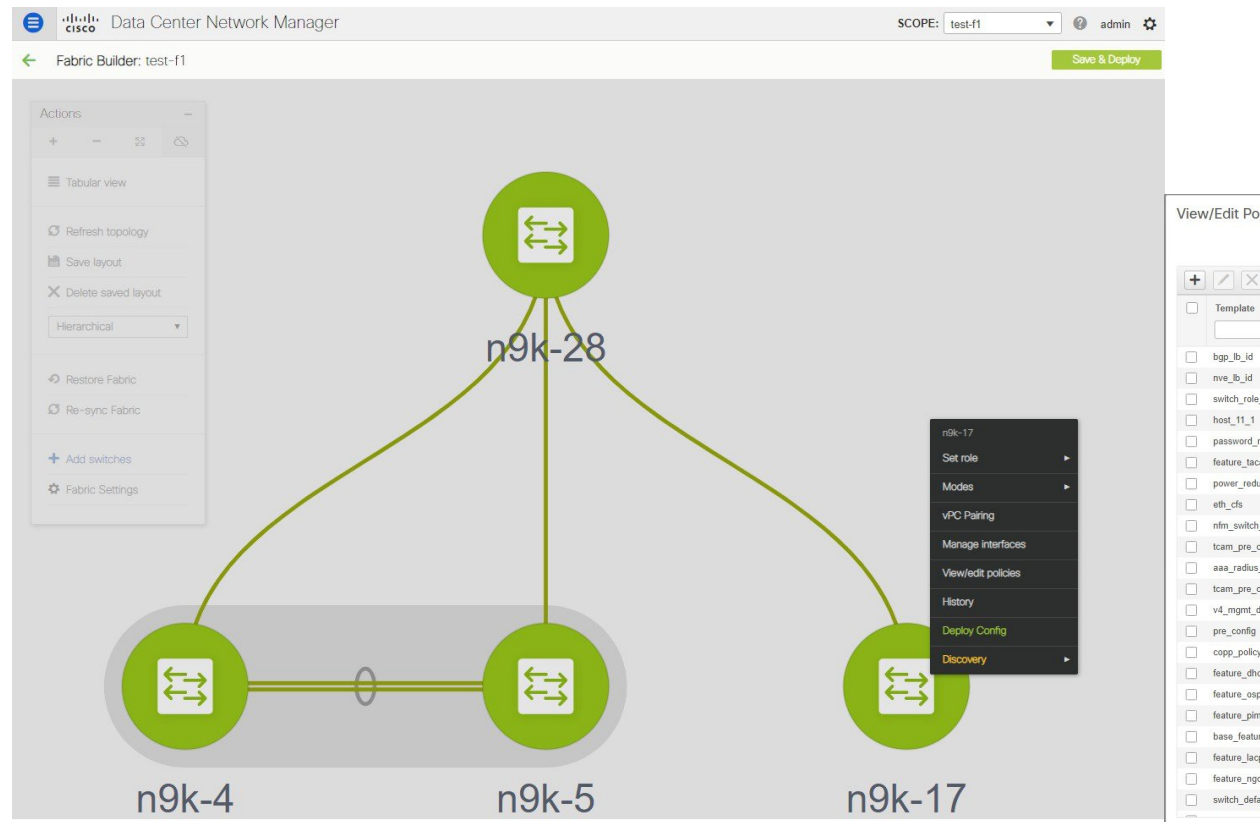
### Additional Notes

When a policy or profile template is applied, an instance is created for each application of the template. The common terminology used for this is Policy Template Instance or PTI. A PTI is effectively a policy or profile template + the Name-value pairs that give it a specific instance, post substitution. PTIs created for a device can be viewed under the View/Edit policies option for that device in Fabric Builder. In the tabular view, the View/Edit policies button allows selection and bulk creation/deletion of policies across a subset of devices in the entire fabric. For more information, see *Viewing and Editing Policies* section.

## Viewing, Editing, and Adding Policies

To navigate to the View/Edit Policies window, right-click a device in the Fabric Builder window and select View/edit policies.





The View/Edit Policies window can be used to view, edit, or create a policy for a device. Note that Interface policies can only be viewed but cannot be edited/created from the View/Edit Policies window. Interfaces can only be edited, created, or deleted from the Interfaces window.

## Viewing Policies

To view certain policies for a device, you can use filters by specifying the search criteria in the empty boxes under each field. After the policies are found, you can view the content by selecting multiple policies and clicking on the “View” button. Below are examples that show how to use filters and how to view the configuration associated with a policy instance.

### Example: Viewing Policies for a Device

Enter `tcam` in the search field to filter the templates, select the template that you want to view, and click the View button to view TCAM policies created for the device.

Template	Policy ID	Fabric Name	Serial Number	Editable	Entity Type	Entity Name
tcam						
<input type="checkbox"/> tcam_pre_config_9300	POLICY-9300	test-f1	SAL18432P6M	true	SWITCH	SWITCH
<input type="checkbox"/> tcam_pre_config_vxlan	POLICY-9330	test-f1	SAL18432P6M	true	SWITCH	SWITCH

**Example: Viewing Policies for an Interface**

Enter the interface name in the search field under Entity Name to filter interfaces. Select an interface, and click the View button to view policies created for the interface.

Template	Policy ID	Fabric Name	Serial Number	Editable	Entity Type	Entity Name	Source	Priority	Content Type	Mark Deleted
						Ethernet1/29				
<input type="checkbox"/> trunk_interface	POLICY-9420	test-f1	SAL18432P6M	false	INTERFACE	Ethernet1/29	Ethernet1/29	350	TEMPLATE_CLI	false
<input type="checkbox"/> int_trunk_host_11_1	POLICY-9390	test-f1	SAL18432P6M	false	INTERFACE	Ethernet1/29	Ethernet1/29	350	PYTHON	false
<input type="checkbox"/> interface_mtu	POLICY-9450	test-f1	SAL18432P6M	false	INTERFACE	Ethernet1/29	Ethernet1/29	352	TEMPLATE_CLI	false
<input type="checkbox"/> porttype_fast_trunk	POLICY-9520	test-f1	SAL18432P6M	false	INTERFACE	Ethernet1/29	Ethernet1/29	352	TEMPLATE_CLI	false
<input type="checkbox"/> no_shut_interface	POLICY-9530	test-f1	SAL18432P6M	false	INTERFACE	Ethernet1/29	Ethernet1/29	352	TEMPLATE_CLI	false



**Note**

- Each interface should be associated with one interface jython policy template.
- An interface jython policy template does not have CLI in its content but rather creates PTIs of CLI policy templates. All these PTIs are combined to generate a complete configuration associated with an interface.

## Editing Policies

Not all device policies can be edited from the View/Edit policies window. Only the policies that are created with an empty Source and have the flag Editable = true, can be edited.

### Procedure

- Step 1** To edit a device policy, select an existing policy and click on the edit or ‘Pencil’ button. The ‘Edit Policy’ window opens.
- Step 2** After changing 1 or more Name-value pairs, press the ‘Save’ button to save the changes on the Edit Policy window.
- Step 3** To deploy the changed config, go back to the Fabric Builder window, right-click on the device and select ‘Deploy Config’.
- This will invoke Configuration Compliance to generate the pending config for the device. Pending config is the diff between the current config on the switch and the new intent config.
- Step 4** If the pending config is correct, click ‘Deploy Config’ to push the pending config onto the switch.

### Example: Editing a Policy

This example shows how to change the IPv4 management default gateway.

The screenshot displays the 'Edit Policy' window for policy POLICY-9140. The policy is of type SWITCH and is based on the template v4\_mgmt\_default\_gateway. The priority is set to 910. In the 'General' tab, the 'Default Gateway' is configured as 22.0.0.88. The background shows a list of policies, with 'v4\_mgmt\_default\_gat...' selected. To the right, a 'Config Deployment' panel shows a table with columns 'Switch Name' and 'IP Address', with one entry: n9k-17, 22.0.0.17.

## Adding Policies

### Procedure

**Step 1** To add a policy to a device, click the '+' button on the View/Edit Policies page.  
The 'Add Policy' windows opens.

**Step 2** From the Policy drop-down list, select a policy to be added to the device.

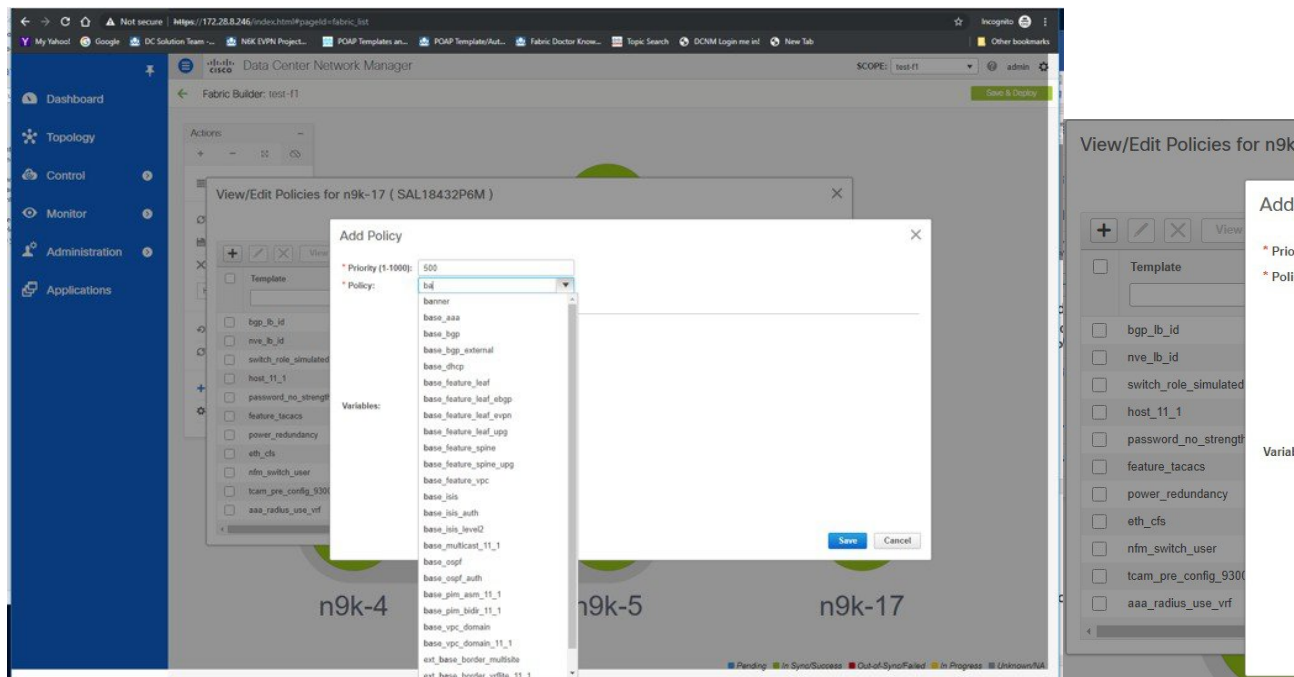
**Step 3** Set the policy priority and input the mandatory fields.

**Step 4** Click the 'Save' button to save and complete adding the policy.

**Note** Policy Priority is used to determine the order in which the configuration will be applied to the switch. Lower priority PTIs are placed before the higher priority PTIs in the expected configuration or intent and this in turn is the order to which the configuration will be pushed via the deployer module. Default priority is 500.

### Adding a Banner Policy

This example shows how to add a banner policy to a device.



## Deploying New Configurations

There are two ways to deploy the new configurations:

1. Navigate to the Fabric Builder window, right-click on the device and select 'Deploy Config' (this is the recommended way).
2. From the View/Edit Policies window, select the newly added policy, click 'View' to verify the config. If the new config looks good, click the 'Push Config' button to push the new config to the device. Note that 'Push Config' will bypass Configuration Compliance. This option should only be used for exception scenarios such as the case where a new user or SNMP user needs to be added to the switch.

## switch\_freeform Template Usage

The **switch\_freeform** is a special policy template that allows users to specify any freeform config for a device. Usage of the template is as follows:

- Specify switch-level config in the **Switch Freeform Config** parameter.
- The specified config must match the **show run** output with respect to case and newlines. Any mismatch will yield unexpected diffs during deploy.
- An internal **switch\_freeform\_config** CLI policy is created for the specified config.
- Should not use this template for interface configuration except for the SVI interface, as SVI interfaces cannot be configured on the Interfaces page currently.
- Users can create many **switch\_freeform** policies for different configs.
- **switch\_freeform** PTIs are sorted together with the other PTIs based on their policy priorities from low to high.
- A **switch\_freeform** policy can be edited before or after the config is deployed.
- If there is any change in the config content, the previously created internal **switch\_freeform\_config** policy will have its priority changed from a positive to a negative number, and a new internal policy is created for the new config.
- A **negative** priority PTI means that CLIs in the PTI need to be deleted; **Configuration Compliance** will generate the **no** commands accordingly.
- Deleting a **switch\_freeform** policy will change the PTI priority of its internal policy to a negative number.

The following section shows how to create a **switch\_freeform** policy, deploy the policy, and subsequently edit and redeploy the updated policy.

### Example: Create a switch\_freeform policy

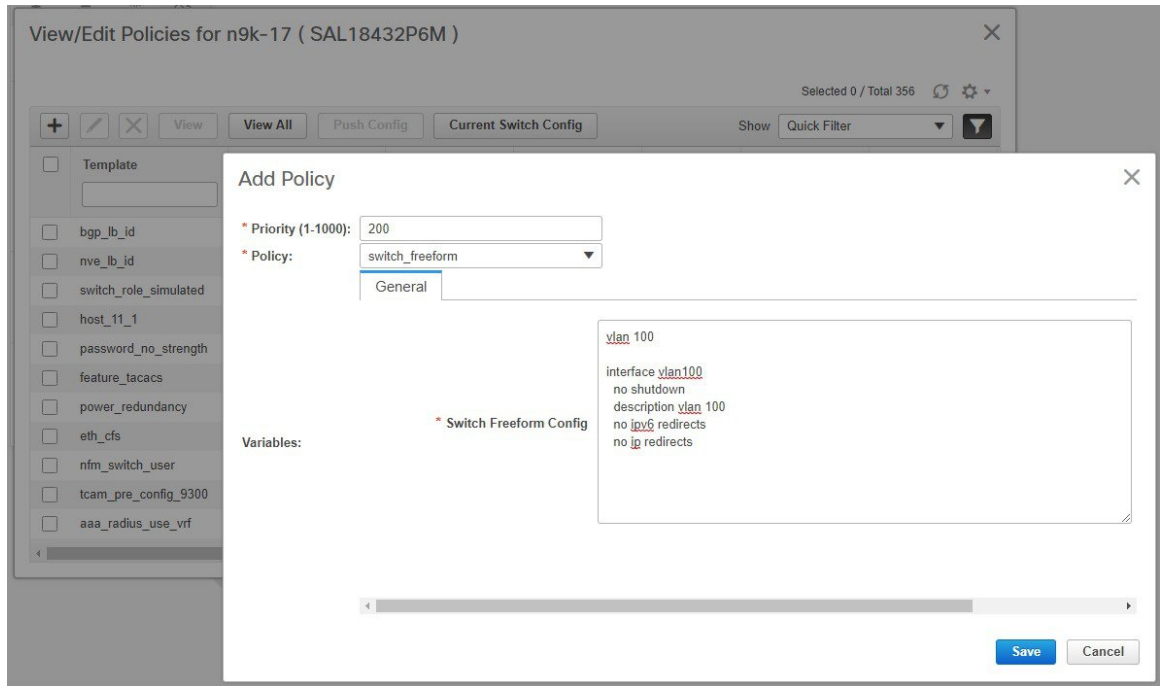
To create a **switch\_freeform** policy, perform the following steps:

#### Procedure

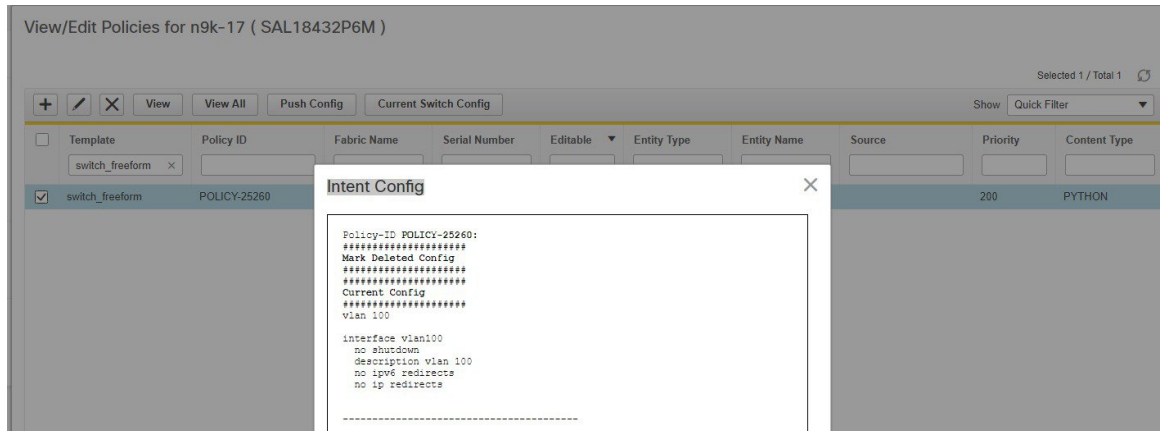
---

- Step 1** Select the **switch\_freeform** template from the policy list in the **Add Policy** screen.  
Set the priority and switch freeform config. Save the policy.

Example: Create a switch\_freeform policy

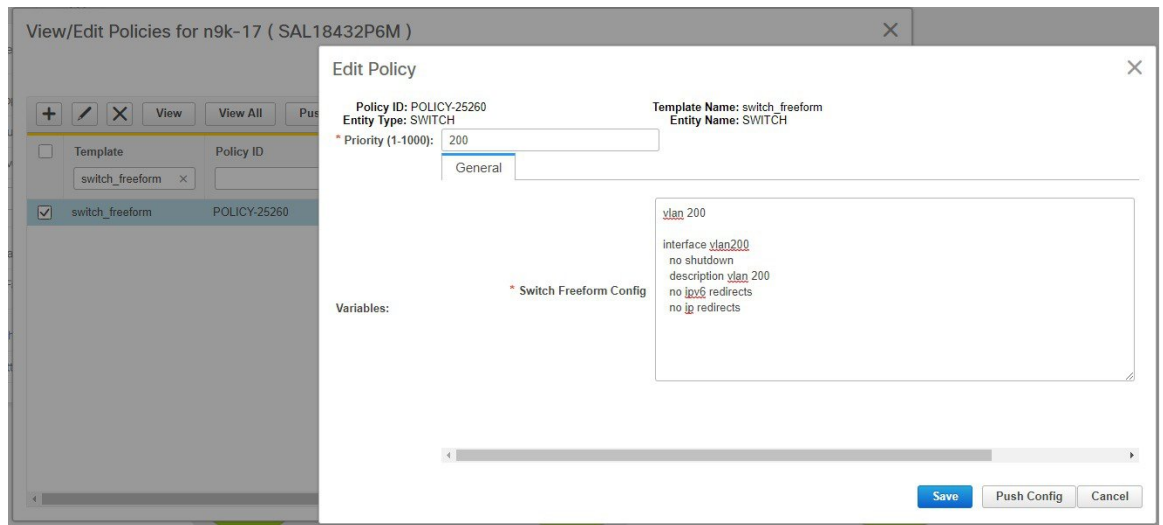


**Step 2** View the intent config of the **switch\_freeform** policy.



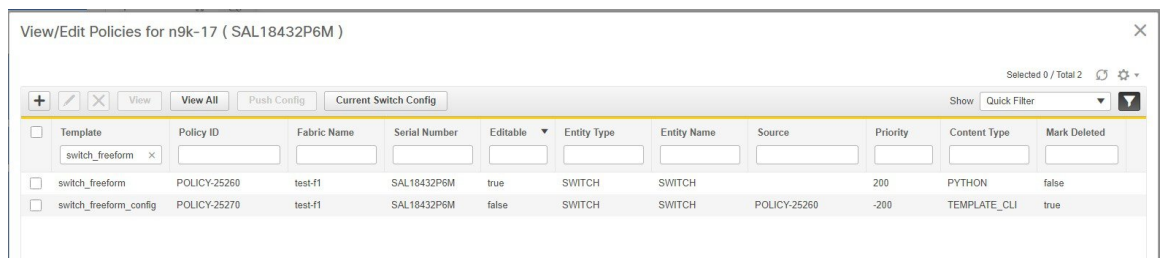
**Step 3** Deploy the switch\_freeform policy from Fabric Builder.

**Step 4** Edit the switch\_freeform policy from the View/Edit Policies window.  
Change the config.

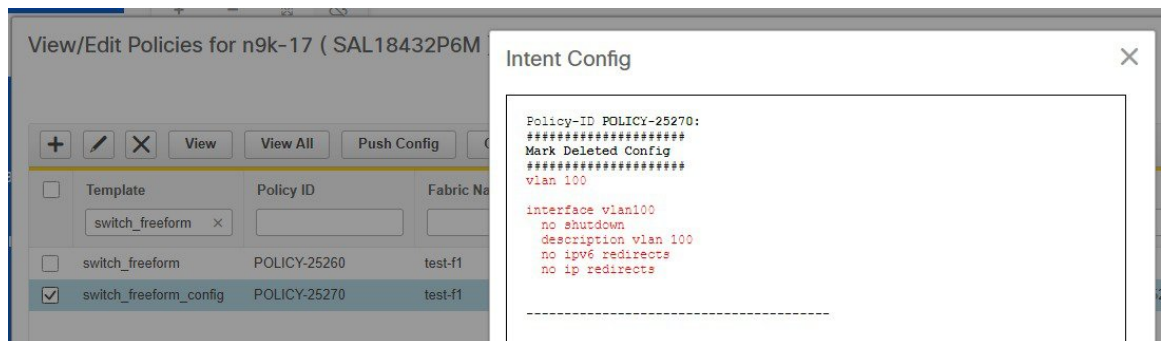


**Step 5** Save the change.

As shown below, the previously created internal **switch\_freeform\_config** policy has its priority changed to a negative number (-200), and the **Mark Deleted** flag is set to true. However, by design, the newly created internal **switch\_freeform\_config** policy is NOT shown.

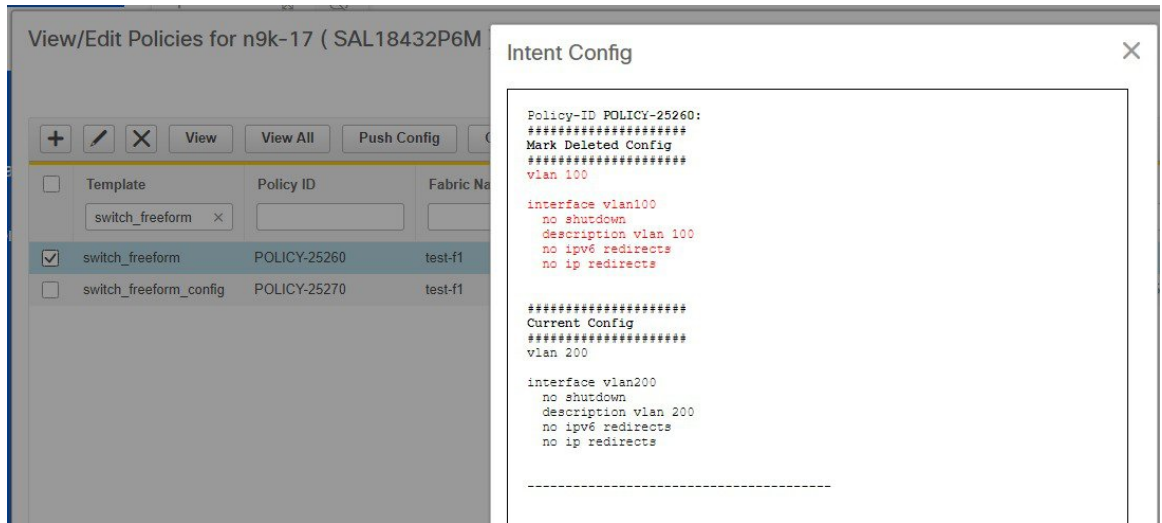


**Step 6** View the intent config of the **mark deleted** internal policy.

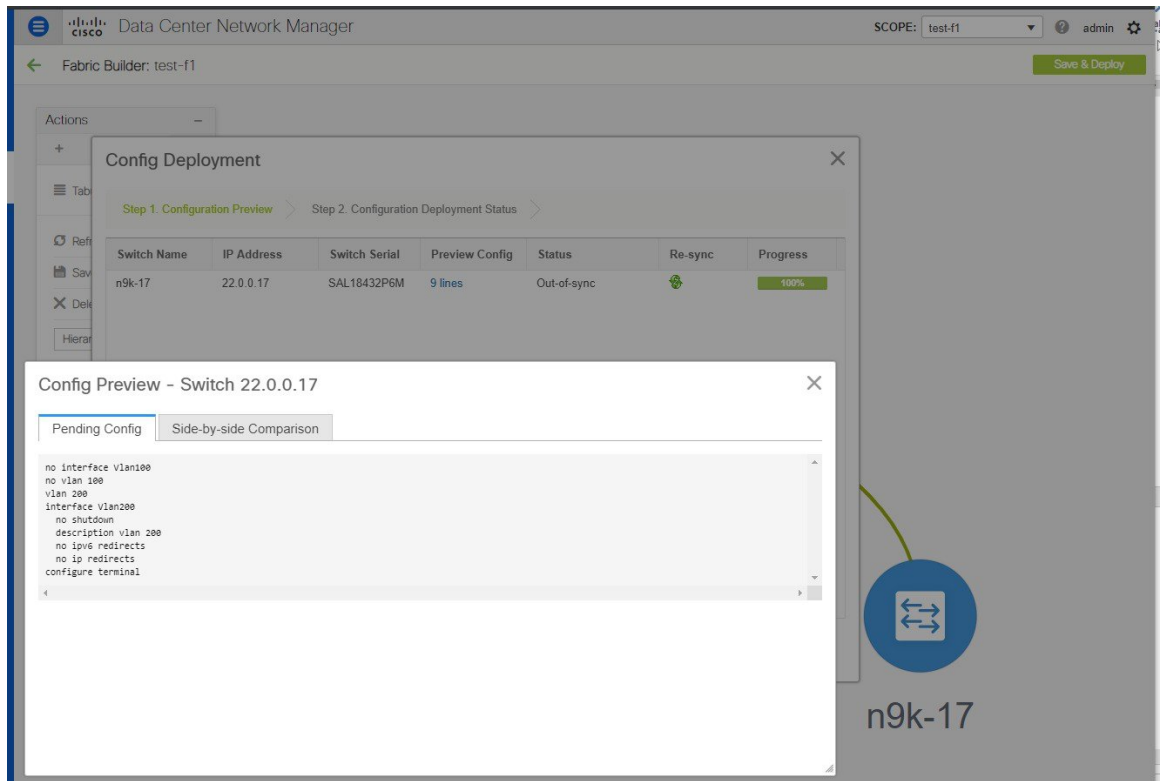


**Step 7** View the intent config of the **changed** switch\_freeform policy before deployment. Note that both the **mark-deleted** and **current configs** are shown.

Changing the Contents of a Template in Use



**Step 8** Deploy the changed config from Fabric Builder.



## Changing the Contents of a Template in Use

A template in general, whether it is a policy, fabric or profile template, cannot be modified once it has been instantiated. However, there could be cases where you want to edit the content of a template, like fixing a bug



in the template or changing an already deployed config. This can be achieved by toggling the `template.in_use.check` option in the **Administration > Server Properties** tab.

**Procedure**

- Step 1** Change the `template.in_use.check` from **true (default)** to **false**.
- Step 2** Click ‘Apply Changes’ at the upper righthand corner.  
A warning will be popped up indicating that a restart of DCNM is needed.  
Ignore this warning as no restart is needed for the `in_use` flag to take effect.
- Step 3** Edit the desired template(s).
- Step 4** Go to the Fabric Builder page and click ‘Save & Deploy’ for the entire fabric.  
This will regenerate PTIs and the updated content will be picked up and used for the expected configuration (or intent).
- Step 5** Once the contents are re-generated and deployed, change the `template.in_use.check` back to **true** to avoid performance issues.

