Configuring FCoE

This chapter contains the following sections:
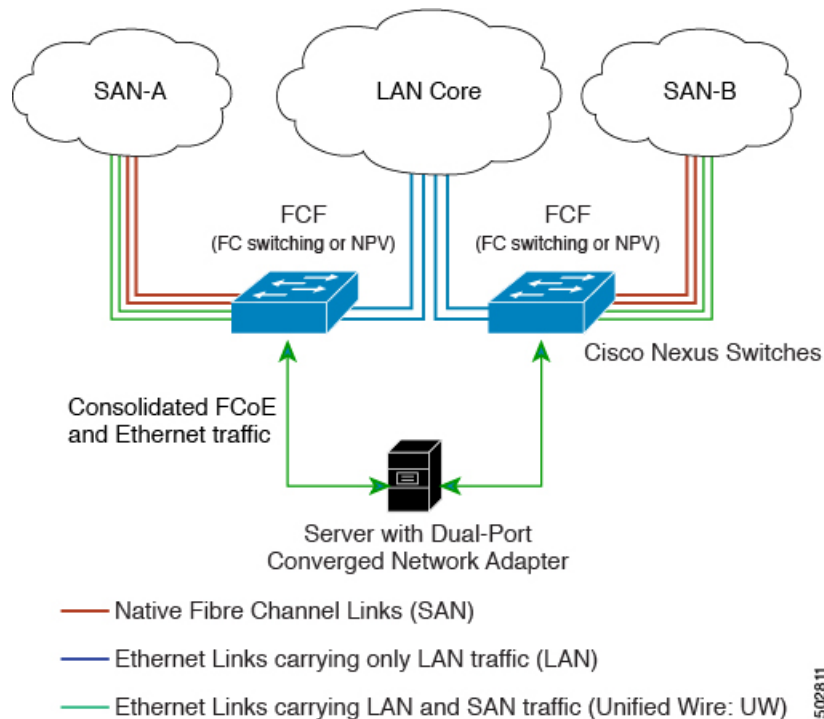
FCoE Topologies

Directly Connected CNA Topology

The Cisco Nexus device can be deployed as a Fibre Channel Forwarder (FCF) as shown in the following figure.

Figure 1: Directly Connected Fibre Channel Forwarder

The following rules are used to process FIP frames to avoid the FCF being used as a transit between an FCoE node (ENode) and another FCF. These rules also prevent login sessions between ENodes and FCFs in different fabrics.

- FIP solicitation and login frames received from the CNAs are processed by the FCF and are not forwarded.

- If an FCF receives solicitations and advertisements from other FCFs over an interface, the following occurs:

  - The frames are ignored and discarded if the FC-MAP value in the frame matches the value of the FCF (the FCF is in the same fabric).

  - The interface is placed in the "FCoE Isolated" state if the FC-MAP value in the FIP frame does not match that of the FCF (the FCF is in a different fabric).

CNAs cannot discover or log in to FCFs that are reachable only through a transit Cisco Nexus FCF. The Cisco Nexus device cannot perform the FCoE transit function between a CNA and another FCF due to hardware limitations.

Because the Cisco Nexus FCF cannot perform the transit FCoE function, you must design your network topology so that the active Spanning Tree Protocol (STP) path of FCoE VLANs is always over the directly connected links between the CNA and the FCF. Make sure that you configure the FCoE VLAN on the directly connected links only.

Remotely Connected CNA Topology

The Cisco Nexus device can be deployed as a Fibre Channel Forwarder (FCF) for remotely connected CNAs, but not as a FIP snooping bridge, as shown in the following figure.

Figure 2: Remotely Connected Fibre Channel Forwarder
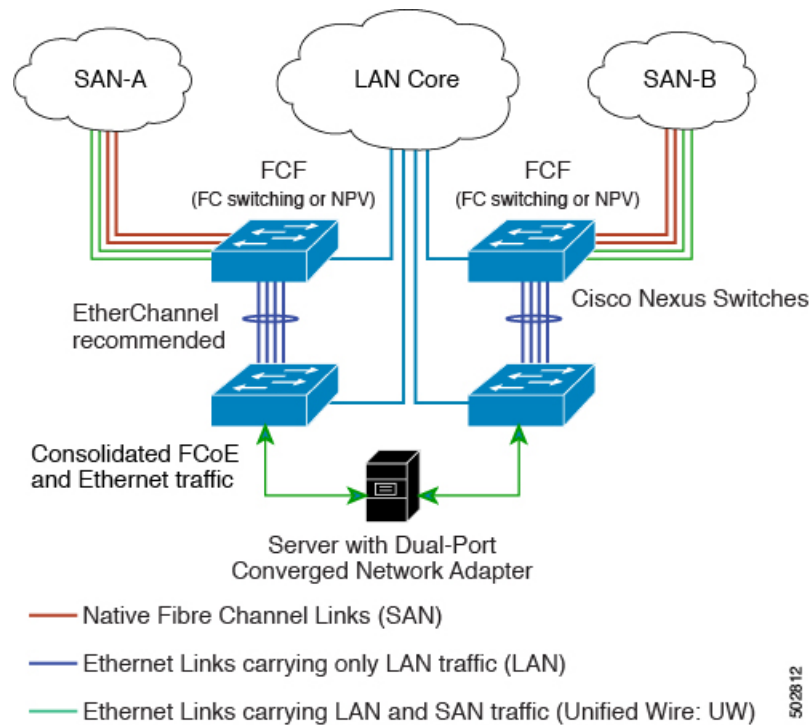
The following rules are used to process FIP frames to avoid the FCF being used as a transit between an ENode and another FCF. These rules also prevent login sessions between ENodes and FCFs in different fabrics.

- FIP solicitation and login frames received from the CNAs are processed by the FCF and are not forwarded.

- If an FCF receives solicitations and advertisements from other FCFs over an interface, the following occurs:

  - The frames are ignored and discarded if the FC-MAP value in the frame matches the value of the FCF (the FCF is in the same fabric).

  - The interface is placed in the "FCoE Isolated" state if the FC-MAP value in the FIP frame does not match that of the FCF (the FCF is in a different fabric).

Because the Cisco Nexus FCF cannot perform the transit FCoE function, you must design your network topology so that the active STP path of FCoE VLANs is always over the directly connected links between the CNA and the FCF. Make sure that you configure the FCoE VLAN on the directly connected links only.

FCoE Best Practices

Directly Connected CNA Best Practice

The following figure shows a best practices topology for an access network that is using directly connected CNAs with Cisco Nexus devices.

Figure 3: Directly Connected CNA
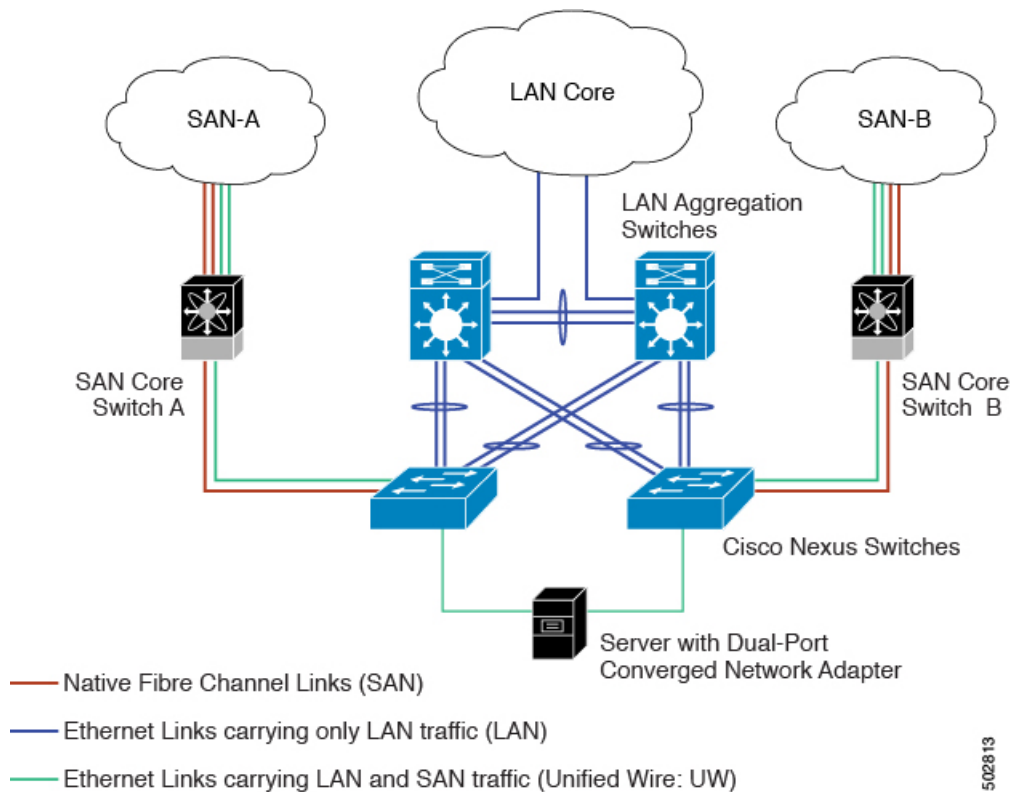
Follow these configuration best practices for the deployment topology in the preceding figure:

1. You must configure a unique dedicated VLAN at every converged access switch to carry traffic for each Virtual Fabric (VSAN) in the SAN (for example, VLAN 1002 for VSAN 1, VLAN 1003 for VSAN 2, and so on). If you enable Multiple Spanning Tree (MST), you must use a separate MST instance for FCoE VLANs.

2. You must configure the unified fabric (UF) links as trunk ports. Do not configure the FCoE VLAN as a native VLAN. You must configure all FCoE VLANs as members of the UF links to allow extensions for VF_Port trunking and VSAN management for the virtual Fibre Channel interfaces.

> **Note**   A unified wire carries both Ethernet and FCoE traffic.

3. You must configure the UF links as spanning-tree edge ports.

4. You must not configure the FCoE VLANs as members of Ethernet links that are not designated to carry FCoE traffic because you want to ensure that the scope of the STP for the FCoE VLANs is limited to UF links only.

5. If the converged access switches (in the same SAN fabric or in another) need to be connected to each other over Ethernet links for a LAN alternate path, you must explicitly configure such links to exclude all FCoE VLANs from membership. This action ensures that the scope of the STP for the FCoE VLANs is limited to UF links only.

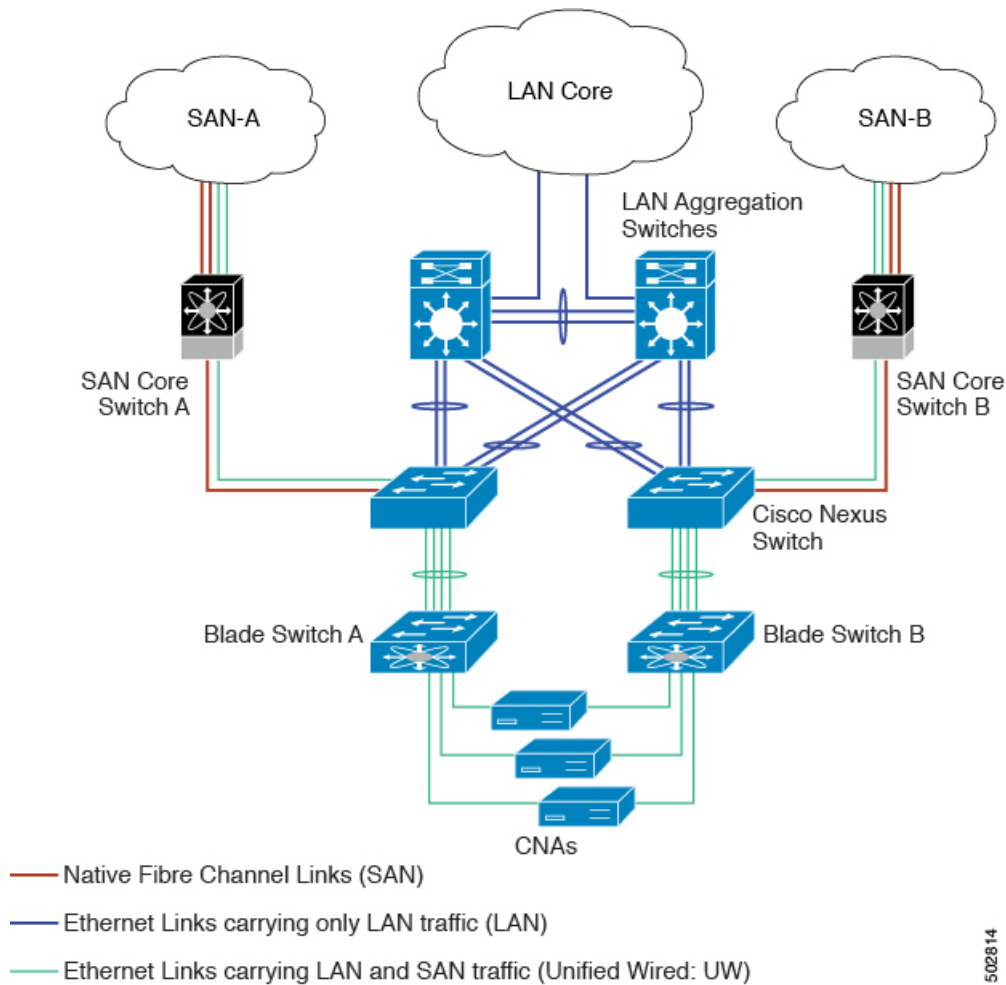6. You must use separate FCoE VLANs for FCoE in SAN-A and SAN-B.

Note All Gen-1 (pre-FIP) and Gen-2, Gen-3, and Gen-4 (FIP) CNAs are supported in a directly connected topology.

Remotely Connected CNA Best Practice

The following figure shows a best practices topology for an access network using remotely connected CNAs with Cisco Nexus devices.

Figure 4: Remotely Connected CNAs

— Native Fibre Channel Links (SAN)

— Ethernet Links carrying only LAN traffic (LAN)

— Ethernet Links carrying LAN and SAN traffic (Unified Wired: UW)

Follow these configuration best practices for the deployment topology in the preceding figure:

1. You must configure a unique dedicated VLAN at every converged access switch to carry traffic for each Virtual Fabric (VSAN) in the SAN (for example, VLAN 1002 for VSAN 1, VLAN 1003 for VSAN 2, and so on). If you enable MST, you must use a separate MST instance for FCoE VLANs.

2. You must configure the unified fabric (UF) links as trunk ports. Do not configure the FCoE VLAN as a native VLAN. You must configure all FCoE VLANs as members of the UF links to allow extensions for VF_Port trunking and VSAN management for the virtual Fibre Channel interfaces.

✎

Note A unified fabric link carries both Ethernet and FCoE traffic.

3. You must configure the CNAs and the blade switches as spanning-tree edge ports.

4. A blade switch must connect to exactly one Cisco Nexus device converged access switch, preferably over an EtherChannel, to avoid disruption due to STP reconvergence on events such as provisioning new links or blade switches.

5. You must configure the Cisco Nexus device converged access switch with a better STP priority than the blade switches that are connected to it. This requirement allows you to create an island of FCoE VLANs where the converged access switch is the spanning-tree root and all the blade switches connected to it become downstream nodes.

6. Do not configure the FCoE VLANs as members of Ethernet links that are not designated to carry FCoE traffic because you want to ensure that the scope of the STP for the FCoE VLANs is limited to UF links only.

7. If the converged access switches and/or the blade switches need to be connected to each over Ethernet links for the purposes of LAN alternate pathing, you must explicitly configure such links to exclude all FCoE VLANs from membership. This action ensures the scope of the STP for FCoE VLANs is limited to UF links only.

8. You must use separate FCoE VLANs for FCoE in SAN-A and SAN-B.

Note A remotely connected topology is supported only with Gen-2, Gen-3, and Gen-4 (FIP) CNAs.

Guidelines and Limitations

FC/FCoE has the following guidelines and limitations:

- FCoE on Cisco Nexus devices support the Gen-1 (pre-FIP) and Gen-2, Gen-3, and Gen-4 (FIP) CNAs. FCoE on the Cisco Nexus 2232PP fabric extender (FEX) supports Gen-2 CNAs only.

- Enabling FCoE on VLAN 1 is not supported.

- Enabling FCoE requires enabling the LLDP feature using **feature lldp**, as LLDP is not enabled by default.

- A combination of straight-through and active-active topologies is not supported on the same FEX.

- FCOE is not supported with Copper SFPs.

- FC/FCoE configuration does not support rollback. If FC/FCoE configurations are present, use the best-effort option. All other configurations will be successful, however, error message will be displayed for the FC/FCoE configuration.

- FCoE is supported on 10-Gigabit, 25-Gigabit, 40-Gigabit, and 100-Gigabit Ethernet interfaces. 100G breakout (4x25G) and 40G breakout (4x10G) is supported on FCoE interfaces.

- Direct connect FCoE (that is, a direct connect to CNAs through a bind interface) is not supported on a port channel of a Cisco Nexus device interface if it is configured to have more than one interface. Direct connect FCoE is supported on port channels with a single link to allow for FCoE from a CNA connected through a vPC with one 10/25/40/100 GB link to each upstream switch.

- Ethernet interfaces used for vFC must have the QOS policy configured manually regardless of default or custom policy defined globally.

> **Note** For a description of the default quality of service (QoS) policies for FC/FCoE, see the Quality of Service guide for your device. For the Nexus software release that you are using. The available versions of this document can be found at the following URL: https://www.cisco.com/c/en/us/support/switches/nexus-9000-series-switches/products-installation-and-configuration-guides-list.html

Configuring FC/FCoE

Perform TCAM Carving

This section explains how to perform TCAM carving.

SUMMARY STEPS

1. Install feature FCoE.
2. Configure the following command (if not configured already) for fcoe to be fully functional.
3. Perform TCAM carving.
4. Use the command **show hardware access-list tcam region** to view the configured TCAM region size.
5. Save the configuration and use the command **reload** to reload the switch.

DETAILED STEPS

Step 1 Install feature FCoE.

```
switch(config)# install feature-set fcoe
    switch(config)# switch(config)# feature-set fcoe
```

Step 2 Configure the following command (if not configured already) for fcoe to be fully functional.

```
hardware access-list tcam region ing-ifacl 256
hardware access-list tcam region ing-redirect 256
```

256 is the minimum tcam space required in ing-ifacl and ing-redirect regions for FC/FCoE.

> **Note** To verify the current tcam configuration use the `show hardware access-list tcam region` command.
>
> If the required tcam space is not available then ing-racl region can be reduced using the `hardware access-list tcam region ing-racl 1536` command.

Step 3 Perform TCAM carving.

Example:

```
Switch(config)# hardware access-list tcam region ing-racl 1536
Switch(config)# hardware access-list tcam region ing-ifacl 256
Switch(config)# hardware access-list tcam region ing-redirect 256
```

Step 4 Use the command **show hardware access-list tcam region** to view the configured TCAM region size.

Example:

```
Switch(config)# show hardware access-list tcam region
Switch(config)#
```

Step 5 Save the configuration and use the command **reload** to reload the switch.

Example:

```
Switch(config)# reload
Switch(config)#
```

What to do next

You must reload the switch after carving TCAM

Configuring LLDP

This section explains how to configure LLDP.

SUMMARY STEPS

1. **configure terminal**
2. **[no]feature lldp**

DETAILED STEPS

Step 1 **configure terminal**

Enters global configuration mode.

Step 2 **[no]feature lldp**

Enables or disables LLDP on the device. LLDP is disabled by default.

Configuring Default QoS

There are four types of FCoE default policies: network QoS, output queuing, input queuing, and QoS. You can enable the FCoE default policies by enabling the FCoE feature using the **feature-set fcoe command** command. The default QoS ingress policy, **default-fcoe-in-policy**, is implicitly attached to all FC and SAN-port-channel interfaces to enable FC to FCoE traffic; this can be verified by using **show interface** {*fc slot/port* | *san-port-channel <no>*} **all** command. The default QoS policy uses CoS3 and Q1 for all FC and FCoE traffic.

Configuring User Defined QoS

To use a different queue or CoS value for FCoE traffic, create user-defined policies. The user-defined QoS ingress policy has to be created and attached explicitly to both FC and FCoE interfaces to enable traffic to use a different queue or CoS. User-defined QoS policies must be created and activated for system-wide QoS.

> **Note** The Ethernet or port-channel interface must be configured with MTU 9216 (or the maximum available MTU size) for supporting FCoE.

The following example demonstrates how to configure and activate user-defined QoS policies that use CoS3 and Q2 for all FC and FCoE traffic.

- Creating a user-defined network QOS policy:

```
switch(config)# policy-map type network-qos fcoe_nq
switch(config-pmap-nqos)# class type network-qos c-nq1
switch(config-pmap-nqos-c)# mtu 1500
switch(config-pmap-nqos-c)# class type network-qos c-nq2
switch(config-pmap-nqos-c)# mtu 9216
switch(config-pmap-nqos-c)# pause pfc-cos 3
switch(config-pmap-nqos-c)# class type network-qos c-nq3
switch(config-pmap-nqos-c)# mtu 1500
switch(config-pmap-nqos-c)# class type network-qos c-nq-default
switch(config-pmap-nqos-c)# mtu 1500
switch(config-pmap-nqos-c)# exit
switch(config-pmap-nqos)# exit
switch(config)#
```

- Creating a user-defined input queuing policy:

```
switch(config)# policy-map type queuing fcoe-in-policy
switch(config-pmap-que)# class type queuing c-in-q2
switch(config-pmap-c-que)# bandwidth percent 50
switch(config-pmap-c-que)# class type queuing c-in-q-default
switch(config-pmap-c-que)# bandwidth percent 50
switch(config-pmap-c-que)# exit
switch(config-pmap-que)# exit
switch(config)
```

- Creating a user-defined output queuing policy:

```
switch(config)# policy-map type queuing fcoe-out-policy
switch(config-pmap-que)# class type queuing c-out-q3
switch(config-pmap-c-que)# priority level 1
switch(config-pmap-c-que)# class type queuing c-out-q-default
switch(config-pmap-c-que)# bandwidth remaining percent 50
switch(config-pmap-c-que)# class type queuing c-out-q1
switch(config-pmap-c-que)# bandwidth remaining percent 0
switch(config-pmap-c-que)# class type queuing c-out-q2
switch(config-pmap-c-que)# bandwidth remaining percent 50
switch(config-pmap-c-que)# exit
switch(config-pmap-que)# exit
switch(config)#
```

- Creating a user-defined QoS input policy:

```
switch(config)# class-map type qos match-any fcoe
switch(config-cmap-qos)# match protocol fcoe
switch(config-cmap-qos)# match cos 3
switch(config-cmap-qos)# exit
switch(config)#
switch(config)# policy-map type qos fcoe_qos_policy
switch(config-pmap-qos)# class fcoe
switch(config-pmap-c-qos)# set cos 3
switch(config-pmap-c-qos)# set qos-group 2
switch(config-pmap-c-qos)# exit
switch(config-pmap-qos)# exit
switch(config)#
```

- Activating a user-defined system QoS policy:

```
switch(config)# system qos
switch(config-sys-qos)# service-policy type queuing input fcoe-in-policy
switch(config-sys-qos)# service-policy type queuing output fcoe-out-policy
switch(config-sys-qos)# service-policy type network-qos fcoe_nq
switch(config-sys-qos)# exit
switch(config)#
```

- Applying a QoS input policy to an FC or FCoE interface:

```
switch# conf
switch(config)# interface {fc <slot>/<port> | ethernet <slot>/<port> | san-port-channel
 <no> | port-channel <no>}
switch(config-if)# service-policy type qos input fcoe_qos_policy
```

- Removing a QoS input policy from an FC or FCoE interface:

```
switch# conf
switch(config)# interface {fc <slot>/<port> | ethernet <slot>/<port> | san-port-channel
 <no> | port-channel <no>}
switch(config-if)# no service-policy type qos input fcoe_qos_policy
```

- Verifying a QoS input policy applied to an FC or FCoE interface:

```
switch# show running-config interface {fc <slot>/<port> | interface <slot>/<port> |
san-port-channel <no> | port-channel <no>} all
```

Note

- When a user-defined QoS policy is used, the same QoS input policy must be applied to all FC and FCoE interfaces in the switch.

- Do not configure **match protocol fcoe** under more than one QoS class map, as FCoE traffic is supported only on a single CoS.

Configuring Traffic Shaping

Traffic shaping is used to control access to available bandwidth and to regulate the flow of traffic in order to avoid congestion that can occur when the sent traffic exceeds the access speed. Because traffic shaping limits the rate of transmission of data, you may use this command only when necessary.

The following example demonstrates how to configure traffic shaper:

- The following command displays the default system level settings for all FC interfaces:

```
switch(config)# show running-config all | i i rate
hardware qos fc rate-shaper
switch(config)#
```

- The following example shows how to configure rate shaper. This command is applied on all FC interfaces:

Note Rarely, you may see input discards on any of the 4G, 8G, 16G, or 32G interfaces. Use the command *hardware qos fc rate-shaper [low]*, to configure the rate shape. Because this is a system level configuration, it will apply to all the FC ports and will reduce the rates for all FC ports. The default option of the command *hardware qos fc rate-shaper* is applicable to all FC interfaces.

```
switch(config)# hardware qos fc rate-shaper low
switch(config)#
switch(config)#end
```

FCoE with vPC Configuration Examples

Beginning Cisco NX-OS Release 9.3(5), the Cisco Nexus N9K-93180YC-FX devices support vPCs and beginning Cisco NX-OS Release 10.1(1), the Cisco Nexus N9K-C93360YC-FX2 devices also support vPCs. The Cisco Nexus N9K-93180YC-FX, N9K-C9336C-FX2-E, and N9K-C93360YC-FX2 devices support vPCs. The vPCscan be configured to increase bandwidth and provide increased load-balancing to the Ethernet fabric. The following are example configurations to explain how to configure FCoE when using vPCs on the Cisco Nexus 9000 Series switches:

Figure 5: FCoE Traffic Flow with Host vPC
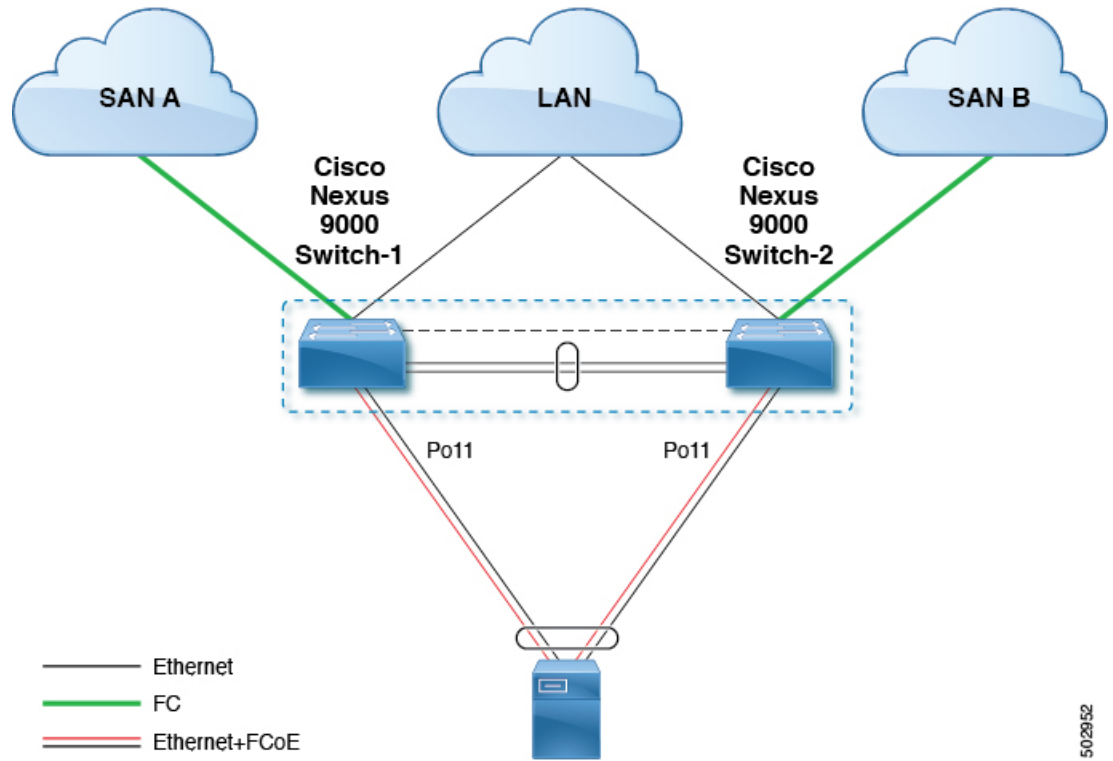
Figure 6: Nexus 9000 FCoE and vPC Lab Topology
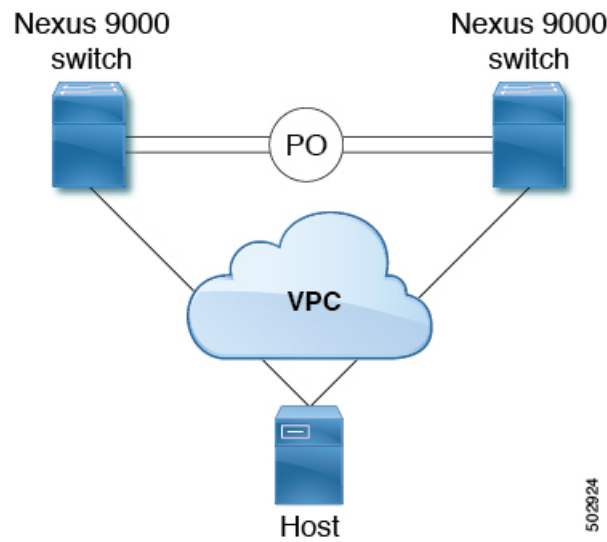
Note FCoE VLANs should not be trunked across vPC peer-links.

✎

Note Only FC uplinks are supported on Cisco Nexus N9K-93180YC-FX switches (switchmode) that connects to core switches.

The configuration example includes the following parameters:

```
switchname: tme-switch-1
switchname: tme-switch-2
mgmt ip: 172.25.182.66
mgmt ip: 172.25.182.67
```

The configuration example includes the following hardware:

- Dell Server PE2950

- Emulex CNA or CISCO CNA

- 2 Cisco Nexus 9000 switches running Cisco NX-OS Release 9.3(5)10.2(1)F or later releases.

The configuration example includes the following considerations and requirements:

- Generation 2 CNAs that support DCBX are required.

- Single host CNA port channel connection to a separate switch. FCoE interfaces will not be brought up if the port channel on a single switch contains more than one member port in a port channel or vPC.

- Cisco NX-OS Release 9.3(5)10.2(1)F or later releases.

- FC Features package is necessary for running FCoE. If this is not installed, there will be a temporary license that will last 90 days.

Cisco Nexus 9000 Series Switch vPC Configuration Example

This example presumes that the basic configuration has been completed on the switch (for example, IP Address (mgmt0), switchname, and password for the administrator).

✎

Note The configuration must be done on both peer switches in the vPC topology.

SUMMARY STEPS

1. **feature vpc**
2. **vPC domain**
3. **vpc peer-link**
4. **show vpc peer-keepalive**
5. **int po**
6. **vpc**
7. **show vpc statistics**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**feature vpc**
	
	Example:
	
	`tme-switch-1# `**`conf t`**
	`Enter configuration commands, one per line. End`
	`with CNTL/Z.`
	`tme-switch-1(config)# `**`feature vpc`**
	`tme-switch-1(config)#`
	
	`tme-switch-2# `**`conf t`**
	`Enter configuration commands, one per line. End`
	`with CNTL/Z.`
	`tme-switch-2(config)# `**`feature vpc`**
	`tme-switch-2(config)#`
		Enable the vPC feature on both peer switches.

Step 2	**vPC domain**
	
	Example:
	
	`tme-switch-1(config)# vpc domain 2`
	`tme-switch-1(config-vpc-domain)# peer-keepalive`
	`destination 192.165.200.230`
	
	`tme-switch-2(config)# vpc domain 2`
	`tme-switch-2(config-vpc-domain)# peer-keepalive`
	`destination 192.165.200.229`
		Configure the vPC domain and peer-keep alive destinations.
		
		Note In this set up, switch tme-switch-1 has the mgmt IP address of 192.165.200.229 and switch tme-switch-2 has the mgmt IP address of 192.165.200.230.

Step 3	**vpc peer-link**
	
	Example:
	
	`tme-switch-1(config)# int port-channel 1`
	`tme-switch-1(config-if)# vpc peer-link`
	
	Note The spanning tree port type is changed to network port type on vPC peer-link. This will enable STP Bridge Assurance on vPC peer-link provided that the STP Bridge Assurance (which is enabled by default) is not disabled.
	
	`tme-switch-2(config)# int port-channel 1`
	`tme-switch-2(config-if)# vpc peer-link`
		Configure the port channel interface that will be used as the vPC peer-link.

Step 4	**show vpc peer-keepalive**
	
	Example:
	
	`tme-switch-1(config)# show vpc peer-keepalive`
	`vPC keep-alive status : peer is alive`
	`--Destination : 172.25.182.167`
	`--Send status : Success`
	`--Receive status : Success`
	`--Last update from peer : (0) seconds, (975) msec`
		Verify that the peer-keepalive can be reached.

Command or Action	Purpose

`tme-switch-1(config)#`
 	
`tme-switch-2(config)# show vpc peer-keepalive`
`--PC keep-alive status : peer is alive`
`--Destination : 172.25.182.166`
`--Send status : Success`
`--Receive status : Success`
`--Last update from peer : (0) seconds, (10336) msec`
`tme-switch-2(config)#`	

Step 5   **int po**
 	
Example:
 	
`tme-switch-1(config-if-range)# int po 1`
`tme-switch-1(config-if)# switchport mode trunk`
`tme-switch-1(config-if)# no shut`
`tme-switch-1(config-if)# exit`
`tme-switch-1(config)# int eth 1/39-40`
`tme-switch-1(config-if-range)# switchport mode`
`trunk`
`tme-switch-1(config-if-range)# channel-group 1`
`tme-switch-1(config-if-range)# no shut`
`tme-switch-1(config-if-range)#`
 	
`tme-switch-2(config-if-range)# int po 1`
`tme-switch-2(config-if)# switchport mode trunk`
`tme-switch-2(config-if)# no shut`
`tme-switch-2(config-if)# exit`
`tme-switch-2(config)# int eth 1/39-40`
`tme-switch-2(config-if-range)# switchport mode`
`trunk`
`tme-switch-2(config-if-range)# channel-group 1`
`tme-switch-2(config-if-range)# no shut`
`tme-switch-2(config-if-range)#`
 	
`tme-switch-1(config-if-range)# show int po1`
`port-channel 1 is up`
`Hardware: Port-Channel, address: 000d.ecde.a92f`
`(bia 000d.ecde.a92f)`
`MTU 1500 bytes, BW 20000000 Kbit, DLY 10 usec,`
`reliability 255/255, txload 1/255, rxload 1/255`
`Encapsulation ARPA`
`Port mode is trunk`
`full-duplex, 10 Gb/s`
`Beacon is turned off`
`Input flow-control is off, output flow-control is`
` off`
`Switchport monitor is off`
`Members in this channel: Eth1/39, Eth1/40`
`Last clearing of "show interface" counters never`
`1 minute input rate 1848 bits/sec, 0 packets/sec`
`1 minute output rate 3488 bits/sec, 3 packets/sec`
`tme-switch-1(config-if-range)#`
 	
`tme-switch-2(config-if-range)# show int po1`
`port-channel1 is up`
`Hardware: Port-Channel, address: 000d.ecdf.5fae`
`(bia 000d.ecdf.5fae) MTU 1500 bytes, BW 20000000`
`Kbit, DLY 10 usec,`
`reliability 255/255, txload 1/255, rxload 1/255`
`Encapsulation ARPA`
	Add member ports to the vpc-peer link port channel and bring up the port channel interface.

	Command or Action	Purpose
| | ```
Port mode is trunk
full-duplex, 10 Gb/s
Beacon is turned off
Input flow-control is off, output flow-control is
 off
Switchport monitor is off
Members in this channel: Eth1/39, Eth1/40
Last clearing of "show interface" counters never
minute input rate 1848 bits/sec, 0 packets/sec
minute output rate 3488 bits/sec, 3 packets/sec
tme-switch-2(config-if-range)#
``` | |
| **Step 6** | **vpc**<br><br>**Example:**<br><br>```
tme-switch-1(config)# int po 11
tme-switch-1(config-if)# vpc 11
tme-switch-1(config-if)# switchport mode trunk
tme-switch-1(config-if)# no shut
tme-switch-1(config-if)# int eth 1/1
tme-switch-1(config-if)# switchport mode trunk
tme-switch-1(config-if)# channel-group 11
tme-switch-1(config-if)# spanning-tree port type
edge trunk
tme-switch-1(config-if)#
```<br><br>**Warning** Edge port type (portfast) should only be enabled on ports connected to a single host. Connecting some devices such as hubs, concentrators, switches, or bridges to this interface when edge port type (portfast) is enabled, can cause temporary bridging loops. Caution should be used in this type of configuration.<br><br>```
tme-switch-2(config)# int po 11
tme-switch-2(config-if)# vpc 11
tme-switch-2(config-if)# switchport mode trunk
tme-switch-2(config-if)# no shut
tme-switch-2(config-if)# int eth 1/1
tme-switch-2(config-if)# switchport mode trunk
tme-switch-2(config-if)# channel-group 11
tme-switch-2(config-if)# spanning-tree port type
edge trunk
```<br><br>**Warning** Edge port type (portfast) should only be enabled on ports connected to a single host. Connecting some devices such as hubs, concentrators, switches, or bridges to this interface when edge port type (portfast) is enabled, can cause temporary bridging loops. Caution should be used in this type of configuration. | Create the vPC and add member interfaces.<br><br>**Note** To run FCoE over a vPC topology, the port channel can only have a sinlge member interface.<br><br>**Note** The vPC number configured under the port channel interface must match on both Nexus 9000 switches. The port channel interface number does not have to match on both switches. |
| **Step 7** | **show vpc statistics**

Example: | Verify that the vPC interfaces are up and operational. |

Command or Action	Purpose
tme-switch-1(config-if)# show vpc statistics vpc 11 port-channel11 is up vPC Status: Up, vPC number: 11 Hardware: Port-Channel, address: 000d.ecde.a908 (bia 000d.ecde.a908) MTU 1500 bytes, BW 10000000 Kbit, DLY 10 usec, reliability 255/255, txload 1/255, rxload 1/255 Encapsulation ARPA Port mode is trunk full-duplex, 10 Gb/s Beacon is turned off Input flow-control is off, output flow-control is off Switchport monitor is off Members in this channel: Eth1/1 Last clearing of "show interface" counters never minute input rate 4968 bits/sec, 8 packets/sec minute output rate 792 bits/sec, 1 packets/sec tme-switch-1(config-if)# tme-switch-2(config-if)# show vpc statistics vpc 11 port-channel11 is up vPC Status: Up, vPC number: 11 Hardware: Port-Channel, address: 000d.ecdf.5fae (bia 000d.ecdf.5fae) MTU 1500 bytes, BW 10000000 Kbit, DLY 10 usec, reliability 255/255, txload 1/255, rxload 1/255 Encapsulation ARPA Port mode is trunk full-duplex, 10 Gb/s Beacon is turned off Input flow-control is off, output flow-control is off Switchport monitor is off Members in this channel: Eth1/1 Last clearing of "show interface" counters never minute input rate 4968 bits/sec, 8 packets/sec minute output rate 792 bits/sec, 1 packets/sec tme-switch-1(config-if)#	

Cisco Nexus 9000 Series Switch FCoE Configuration Example

After setting up vPC between the two Nexus 9000 switches, you can configure the FCoE topology. This procedure presumes that basic configuration has been executed on the Nexus 9000 switch that will provide IP Address (mgmt0), switch name, password for admin, etc. and that the vPC configuration has been completed as outlined in the previous section. The following steps will walk through the basic FCoE configuration necessary to set up an FCoE topology in conjunction with the vPC topology.

SUMMARY STEPS

1. **install feature-set fcoe**
2. **feature-set fcoe**
3. **vsan database**
4. **interface port-channel**
5. **int vfc**

6. **show int brief**
7. **show flogi database**
8. **show vpc statistics**

DETAILED STEPS

	Command or Action	**Purpose**
Step 1	**install feature-set fcoe**	Install FCoE feature.

Step 2	**feature-set fcoe**
	
	Example:
	
	`tme-switch-1(config)# feature-set fcoe`
	`Please configure the following for fcoe to be fully`
	`functional:`
	`- hardware access-list tcam region ing-racl TCAM`
	`size`
	`- hardware access-list tcam region ing-ifacl TCAM`
	`size`
	`- hardware access-list tcam region ing-redirect`
	`TCAM size`
	`tme-switch-1(config)#`
	
	`tme-switch-2(config)# feature-set fcoe`
	`Please configure the following for fcoe to be fully`
	`functional:`
	`- hardware access-list tcam region ing-racl TCAM`
	`size`
	`- hardware access-list tcam region ing-ifacl TCAM`
	`size`
	`- hardware access-list tcam region ing-redirect`
	`TCAM size`
	`tme-switch-2(config)#`
		Enable FCoE on the Cisco Nexus 9000 switch.
		
		Note This can take a few moments to complete. You must ensure to complete the TCAM carving before doing this step. After completing the TCAM carving, you must reload the switch.

Step 3	**vsan database**
	
	Example:
	`tme-switch-1(config)# vsan database`
	`tme-switch-1(config-vsan-db)# vsan 100`
	`tme-switch-1(config-vsan-db)# exit`
	`tme-switch-1(config)# vlan 100`
	`tme-switch-1(config-vlan)# fcoe vsan 100`
	`tme-switch-1(config-vlan)# show vlan fcoe`
	`VLAN VSAN Status`
	`-------- -------- --------`
	`100 100 Operational`
	`tme-switch-1(config-vlan)#`
	
	`tme-switch-2(config)# vsan database`
	`tme-switch-2(config-vsan-db)# vsan 101`
	`tme-switch-2(config-vsan-db)# exit`
	`tme-switch-2(config)# vlan 101`
	`tme-switch-2(config-vlan)# fcoe vsan 101`
	`tme-switch-2(config-vlan)# show vlan fcoe`
	`VLAN VSAN Status`
	`-------- -------- --------`
	`101 101 Operational`
	`tme-switch-2(config)#`
		Create a VSAN and map it to a VLAN that has been designated to carry FCoE traffic.
		
		Note VLAN and VSAN numbers are not required to be the same.

	Command or Action	**Purpose**
Step 4	**interface port-channel** **Example:** ``` tme-switch-1(config)# interface port-channel 11 tme-switch-1(config-if)# switchport trunk allowed vlan 1, 100 tme-switch-1(config-if)# mtu 9216 tme-switch-1(config-if)# service-policy type qos input default-fcoe-in-policy tme-switch-1(config-if)# show int trunk -- Port Native Status Port -- Eth1/1 1 trnk-bndl Po11 Eth1/39 1 trnk-bndl Po1 Eth1/40 1 trnk-bndl Po1 Po1 1 trunking -- Po11 1 trunking -- -- Port Vlans Allowed on Trunk -- Eth1/1 1,100 Eth1/39 1-3967,4048-4093 Eth1/40 1-3967,4048-4093 Po1 1-3967,4048-4093 Po11 1,100 -- Port Vlans Err-disabled on Trunk -- Eth1/1 none Eth1/39 100 Eth1/40 100 Po1 100 Po11 none -- Port STP Forwarding -- Eth1/1 none Eth1/39 none Eth1/40 none Po1 1 Po11 1,100 tme-switch-1(config-if)# tme-switch-2(config)# int po 11 tme-switch-2(config-if)# switchport trunk allowed vlan 1, 101 tme-switch-1(config-if)# mtu 9216 tme-switch-1(config-if)# service-policy type qos input default-fcoe-in-policy tme-switch-2(config-if)# show int trunk -- Port Native Status Port -- Eth1/1 1 trnk-bndl Po11 Eth1/39 1 trnk-bndl Po1 Eth1/40 1 trnk-bndl Po1 ```	Configure the VLANs that are allowed to transverse the vPC links.

	Command or Action	Purpose
| | ```
Po1 1 trunking --
Po11 1 trunking --


----------------------------------------------
Port Vlans Allowed on Trunk
----------------------------------------------
Eth1/1 1,101
Eth1/39 1-3967,4048-4093
Eth1/40 1-3967,4048-4093
Po1 1-3967,4048-4093
Po11 1,101


----------------------------------------------
Port Vlans Err-disabled on Trunk
----------------------------------------------
Eth1/1 none
Eth1/39 101
Eth1/40 101
Po1 101
Po11 none


----------------------------------------------
Port STP Forwarding
----------------------------------------------
Eth1/1 none
Eth1/39 none
Eth1/40 none
Po1 1
Po11 1,101
tme-switch-2(config-if)#
``` | |
| **Step 5** | **int vfc**<br><br>**Example:**<br><br>```
tme-switch-1(config)# int vfc 1
tme-switch-1(config-if)# bind interface po11
tme-switch-1(config-if)# no shut
tme-switch-1(config-if)#

tme-switch-2(config)# int vfc 1
tme-switch-2(config-if)# bind interface po11
tme-switch-2(config-if)# no shut
tme-switch-2(config-if)#

tme-switch-1(config)# vsan database
tme-switch-1(config-vsan-db)# vsan 100 interface
vfc 1
tme-switch-1(config)# show vsan membership
vsan 1 interfaces:
fc2/1 fc2/2 fc2/3 fc2/4
fc2/5 fc2/6 fc2/7 fc2/8

vsan 100 interfaces:
vfc1

vsan 4079(evfp_isolated_vsan) interfaces:

vsan 4094(isolated_vsan) interfaces:
tme-switch-1(config)#

tme-switch-2(config)# vsan database
``` | Create a virtual Fibre Channel interface (vfc) and add it to the VSAN that was created in the previous step. |

	Command or Action	Purpose
	``tme-switch-2(config-vsan-db)# vsan 101 interface vfc 1
tme-switch-2(config)# show vsan membership
vsan 1 interfaces:
fc2/1 fc2/2 fc2/3 fc2/4
fc2/5 fc2/6 fc2/7 fc2/8

vsan 101 interfaces:
vfc1

vsan 4079(evfp_isolated_vsan) interfaces:

vsan 4094(isolated_vsan) interfaces:
tme-switch-2(config)#``` | |
| **Step 6** | **show int brief**

Example:

```tme-switch-1(config-if)# show int brief

Ethernet VLAN Type Mode Status Reason Speed

Eth1/1 1 eth trunk up none 10G(D)
Eth1/2 1 eth access up none 10G(D)
Eth1/38 1 eth access down SFP not inserted 10G(D)
Eth1/39 1 eth trunk up none 10G(D)
Eth1/40 1 eth trunk up none 10G(D)

Port-channel VLAN Type Mode Status Reason Speed

Po1 1 eth trunk up none a-10G(D) none
Po11 1 eth trunk up none a-10G(D) none

Port VRF Status IP Address Speed MTU
----
mgmt0 -- up 172.25.182.166 1000 1500

Interface Vsan Admin Admin Status SFP Oper Oper
Port
----
vfc1 100 F on up -- F auto --
tme-switch-1(config-if)#

tme-switch-2(config-if)# show int brief

Ethernet VLAN Type Mode Status Reason Speed Port

Eth1/1 1 eth trunk up none 10G(D) 11
Eth1/2 1 eth access up none 10G(D) --
Eth1/38 1 eth access down SFP not inserted 10G(D)
 --
Eth1/39 1 eth trunk up none 10G(D) 1
Eth1/40 1 eth trunk up none 10G(D) 1

Port-channel VLAN Type Mode Status Reason Speed
Protocol
--``` | Verify that the vfc is up and operational: |

	Command or Action	Purpose
| | ```
Po1 1 eth trunk up none a-10G(D) none
Po11 1 eth trunk up none a-10G(D) none


------------------------------------------------
Port VRF Status IP Address Speed MTU
------------------------------------------------
mgmt0 -- up 172.25.182.167 1000 1500


------------------------------------------------
Interface Vsan Admin Admin Status SFP Oper Oper
------------------------------------------------
vfc1 101 F on up -- F auto --
tme-switch-2(config-if)#
``` | |
| Step 7 | **show flogi database**<br><br>**Example:**<br>```
tme-switch-1# show flogi database
------------------------------------------------
INTERFACE VSAN FCID PORT NAME NODE NAME
------------------------------------------------
vfc1 100 0x540000 21:00:00:c0:dd:11:2a:01
20:00:00:c0:dd:11:2a:01

Total number of flogi = 1.
tme-switch-2# show flogi database
------------------------------------------------
INTERFACE VSAN FCID PORT NAME NODE NAME
------------------------------------------------
vfc1 101 0x540000 21:00:00:c0:dd:11:2a:01
20:00:00:c0:dd:11:2a:01

Total number of flogi = 1.
``` | Verify that the virtual Fibre Channel interface has logged into the fabric. |
| Step 8 | **show vpc statistics**<br><br>**Example:**<br>```
tme-switch-1(config-if)# show vpc statistics vpc
11
port-channel11 is up
vPC Status: Up, vPC number: 11
Hardware: Port-Channel, address: 000d.ecde.a908
(bia 000d.ecde.a908)
MTU 1500 bytes, BW 10000000 Kbit, DLY 10 usec,
reliability 255/255, txload 1/255, rxload 1/255
Encapsulation ARPA
Port mode is trunk
full-duplex, 10 Gb/s
Beacon is turned off
Input flow-control is off, output flow-control is
 off
Switchport monitor is off
Members in this channel: Eth1/1
Last clearing of "show interface" counters never
1 minute input rate 4968 bits/sec, 8 packets/sec
1 minute output rate 792 bits/sec, 1 packets/sec

tme-switch-2(config-if)# show vpc statistics vpc
11
port-channel11 is up
``` | Verify that the vPC is up and operational. |

Command or Action	Purpose
vPC Status: Up, vPC number: 11 Hardware: Port-Channel, address: 000d.ecdf.5fae (bia 000d.ecdf.5fae) MTU 1500 bytes, BW 10000000 Kbit, DLY 10 usec, reliability 255/255, txload 1/255, rxload 1/255 Encapsulation ARPA Port mode is trunk full-duplex, 10 Gb/s Beacon is turned off Input flow-control is off, output flow-control is off Switchport monitor is off Members in this channel: Eth1/1 Last clearing of "show interface" counters never 1 minute input rate 4968 bits/sec, 8 packets/sec 1 minute output rate 792 bits/sec, 1 packets/sec	

Configuring QoS

You need to attach the system service policy to configure QoS. The **service-policy** command specifies the system class policy map as the service policy for the system.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **system qos**
3. switch(config-sys-qos)# **service-policy type** {**network-qos** | **qos** | **queuing**} [**input** | **output**] *fcoe default policy-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **system qos**	Enters system qos configuration mode.
Step 3	switch(config-sys-qos)# **service-policy type** {**network-qos** | **qos** | **queuing**} [**input** | **output**] *fcoe default policy-name*	Specifies the default FCoE policy map to use as the service policy for the system. There are four pre-defined policy-maps for FCoE: • service-policy type queuing input fcoe-default-in-policy • service-policy type queuing output fcoe-default-out-policy • service-policy type qos input fcoe-default-in-policy • service-policy type network-qos fcoe-default-nq-policy

Command or Action	Purpose	
	Note	Before enabling FCoE on a Cisco Nexus device, you must attach the pre-defined FCoE policy maps to the type qos, type network-qos, and type queuing policy maps.

Information About TCAM Carving

The Ternary Content-Addressable Memory (TCAM) carving feature uses a template-based approach that enables you to modify the default region sizes of the TCAM. When the switch boots up, you see this default template, unless you have configured any other template. This table lists the types and sizes of various regions in a template.

Table 1: Predefined Built-In Default Template

Region	Size (Entries)	Size (Blocks)	Features
Vacl	1024	16	Ingress VLAN access control list (VACL), egress VACL
Ifacl	1152	18	Ingress interface ACL, ingress Layer 3 physical port/subinterface RACL, egress RACL for all ports, default Control Plane Policing (CoPP)
Qos	448	7	Ingress vlan-qos, ingress system-qos, ingress interface-qos
Rbacl	1152	18	Ingress Layer 3 switch virtual interface, ingress Layer 3 port channel/port channel subinterface router access control list (RACL), egress Cisco Trusted Security (CTS)
Span	64	1	Span
Sup	256	4	Sup-rdt
Total	**4096**	**64**	

Information About User-Defined Templates

In addition to the default template, you can create a maximum of 16 templates (which means that you can have 17 templates at one time). You can specify whatever sizes of ternary content addressable memory (TCAM) regions you want.

You can apply the following operations on each template:

- Create
- Modify
- Delete
- Commit

Each template can be in one of the following states:

- Saved
- Committed

Create

When you create a template, the size of the TCAM regions are initialized to the default values. When a template is created, the template is in the saved state by default. Once you create a template, you can modify it to change the size of any TCAM region. You should configure the size of the region in multiples of 64 because the size of each TCAM block is 64 entries. If you enter a value that is not a multiple of 64, an error message asks you to enter the value again.

Modify

You can modify any saved template to change the size of any TCAM region but you cannot modify the size of any region in the TCAM to 0. During the modification, the software checks that the size that you entered is on a 64 boundary. When you modify a template, the combined size of all the TCAM regions might have fewer than 4096 entries. During a modification, the software does not check that you have fewer than 4096 entries.

You can modify a template only when it is in the saved state. After a template is committed, you cannot modify it.

A user-defined committed template can be changed to the created state by servicing another user-defined template or default template.

To service another user-defined template, enter the following command:

hardware profile tcam resource service-template *user-defined-template*

To service a default template, enter the following command:

no hardware profile tcam resource service-template *currently-committed- template*

Delete

You can delete any saved template. After you delete a template, all information about the template is lost. A committed template cannot be deleted.

A user-defined committed template can be changed to the created state by servicing another user-defined template or default template.

To service another user-defined template, enter the following command:

hardware profile tcam resource service-template *user-defined-template*

To service a default template, enter the following command:

no hardware profile tcam resource service-template *currently-committed- template*

Commit

You can commit any of your user-defined templates or the default template that is provided by the software. To commit a template, enter the **commit** command and perform a reboot of the switch. When you enter the **commit** command, the software validates the template. If the validation is successful, the software prompts you to reboot the switch. The template (user defined or default) is applied after the reboot. If you did not choose to reboot, no changes are made to the TCAM regions and no template is committed.

From Cisco NX-OS Release 9.3(3) onwards, after you commit a template, the system prompts you whether to proceed with copying the running configuration to the startup configuration and rebooting the switch. After you agree to continue, the following occurs:

- The committed template is saved in the startup configuration.

- The switch is rebooted.

- The committed template is used by the software.

- The template goes to the running state.

Note Prior to Cisco NX-OS Release 9.3(3), after you commit a template, the system does not automatically reboot but a message is displayed in the **commit** command output asking you to reboot the switch for the committed template to take effect.

If you perform a write erase, reload, and copy running configuration from a back-up configuration containing uncommitted TCAM profile, the following occurs:

1. After the TCAM profile is committed, switch automatically reloads without any prompt.

2. Any configuration after TCAM carving CLI is not applied.

3. To restore configuration with the committed TCAM profile, you need to copy backup configuration to running configuration again. However, there is no switch reload as the TCAM carving profile is already committed.

When the switch is reloaded due to the new committed TCAM profile, the **show system reset-reason** command displays the reason for the reload as shown below:

```
switch# show system reset-reason
----- reset reason for Supervisor-module 1 (from Supervisor in slot 1) ---
1) At 302777 usecs after Sun Jan 20 22:02:37 2019
   Reason: Reload due to change in TCAM service-template
   Service:
   Version: 9.3(3)

2) At 314447 usecs after Sun Jan 20 21:52:58 2019
```

```
    Reason: Reset Requested by CLI command reload
    Service:
    Version: 9.3(3)

3) At 20142 usecs after Sun Jan 20 21:27:33 2019
    Reason: Reset Requested by CLI command reload
    Service:
    Version: 9.3(3)
```

After the switch reboots, the committed template is applied to all ASICs on the Cisco Nexus device. You cannot commit different templates to different ASICs on the Cisco Nexus device. All saved templates and committed templates along with the size of each region of each template are displayed in the running configuration.

When a template is committed, the software checks the following:

1. The combined size of all regions in the TCAM is 4096 entries.

2. The size of each region fits within the TCAM. At any point of time, there is always a running size for the TCAM region. This running size (the current size in the hardware TCAM) is defined by either the default or a user-defined template that was committed and is currently being used as the running template. If you increase the size of a region in a template that is currently being committed, from the current running size, the software checks if there are enough free entries outside the current region (entries that are not allocated to any other region) that can be used to increase the size of the region. If you decrease the size of a region in a template that is currently being committed from the current running size, the software checks to determine if there are enough free entries within the region that can be freed up to reduce the size of the TCAM region. All changes that reduce the sizes of the regions within the template are done before the changes to increase the sizes of regions within the template.

3. You cannot change the supervisor region size to be smaller than 256 entries because the software must have 256 entries to support all features in the sup-region.

4. The supervisor region default size 128 entries even though 256 entries are available. By using TCAM carving, the additional 128 entries can be used. The **sup** keyword is available in the CLI to change the values of the sup-region to 128, 192, or 256.

5. The hardware does not support more than 256 entries in the supervisor region and span regions. This check is done during validation.

If all these checks pass, you can commit he template and you are prompted to apply the template by rebooting.

If these checks fail, the commit fails and the template goes back to the saved state. If the commit fails, the **commit** command output displays the reasons that it failed.

You cannot modify or delete the default template. You can only move this template from saved to committed or committed to saved. If the default template is committed, it is not displayed in the running configuration. To apply the default template, enter the **no commit** command using the currently running template. Entering this command executes the same validation checks that were performed when you committed the template. If all validations succeed, the software prompts you to reboot the switch. If you agree to reboot, the template is saved in the startup configuration and the system is rebooted. After the reboot, the default template is applied. The startup configuration has the committed template that you committed before rebooting. After rebooting, the template in the startup configuration is used. If there is no committed template in the startup configuration, the default template is used.

You create and manage the TCAM carving templates by entering the template manager commands. The template-based TCAM carving CLI is supported in config-sync. Only template creation is supported inside config-sync. Template commit should be performed separately on each switch outside the config-sync context.

Creating a User-Defined Template

SUMMARY STEPS

    1. switch# **configure terminal**
    2. switch(config)# **hardware profile tcam resource template** *template-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **hardware profile tcam resource template** *template-name*	Creates a new template with the default region sizes. A maximum of 16 templates (plus the default) can be created. The *template-name* argument can be a maximum of 64 characters.

Example

This example shows how to create a user-defined template named qos-template:

```
switch# configure terminal
switch(config)# hardware profile tcam resource template qos-template
```

Information About User-Defined Templates

In addition to the default template, you can create a maximum of 16 templates (which means that you can have 17 templates at one time). You can specify whatever sizes of ternary content addressable memory (TCAM) regions you want.

You can apply the following operations on each template:

- Create
- Modify
- Delete
- Commit

Each template can be in one of the following states:

- Saved
- Committed

Create

When you create a template, the size of the TCAM regions are initialized to the default values. When a template is created, the template is in the saved state by default. Once you create a template, you can modify it to change the size of any TCAM region. You should configure the size of the region in multiples of 64 because the size

of each TCAM block is 64 entries. If you enter a value that is not a multiple of 64, an error message asks you to enter the value again.

Modify

You can modify any saved template to change the size of any TCAM region but you cannot modify the size of any region in the TCAM to 0. During the modification, the software checks that the size that you entered is on a 64 boundary. When you modify a template, the combined size of all the TCAM regions might have fewer than 4096 entries. During a modification, the software does not check that you have fewer than 4096 entries.

You can modify a template only when it is in the saved state. After a template is committed, you cannot modify it.

A user-defined committed template can be changed to the created state by servicing another user-defined template or default template.

To service another user-defined template, enter the following command:

hardware profile tcam resource service-template *user-defined-template*

To service a default template, enter the following command:

no hardware profile tcam resource service-template *currently-committed- template*

Delete

You can delete any saved template. After you delete a template, all information about the template is lost. A committed template cannot be deleted.

A user-defined committed template can be changed to the created state by servicing another user-defined template or default template.

To service another user-defined template, enter the following command:

hardware profile tcam resource service-template *user-defined-template*

To service a default template, enter the following command:

no hardware profile tcam resource service-template *currently-committed- template*

Commit

You can commit any of your user-defined templates or the default template that is provided by the software. To commit a template, enter the **commit** command and perform a reboot of the switch. When you enter the **commit** command, the software validates the template. If the validation is successful, the software prompts you to reboot the switch. The template (user defined or default) is applied after the reboot. If you did not choose to reboot, no changes are made to the TCAM regions and no template is committed.

From Cisco NX-OS Release 9.3(3) onwards, after you commit a template, the system prompts you whether to proceed with copying the running configuration to the startup configuration and rebooting the switch. After you agree to continue, the following occurs:

 • The committed template is saved in the startup configuration.

 • The switch is rebooted.

 • The committed template is used by the software.

 • The template goes to the running state.

✎

Note   Prior to Cisco NX-OS Release 9.3(3), after you commit a template, the system does not automatically reboot but a message is displayed in the **commit** command output asking you to reboot the switch for the committed template to take effect.

If you perform a write erase, reload, and copy running configuration from a back-up configuration containing uncommitted TCAM profile, the following occurs:

1. After the TCAM profile is committed, switch automatically reloads without any prompt.

2. Any configuration after TCAM carving CLI is not applied.

3. To restore configuration with the committed TCAM profile, you need to copy backup configuration to running configuration again. However, there is no switch reload as the TCAM carving profile is already committed.

When the switch is reloaded due to the new committed TCAM profile, the **show system reset-reason** command displays the reason for the reload as shown below:

```
switch# show system reset-reason
----- reset reason for Supervisor-module 1 (from Supervisor in slot 1) ---
1) At 302777 usecs after Sun Jan 20 22:02:37 2019
   Reason: Reload due to change in TCAM service-template
   Service:
   Version: 9.3(3)

2) At 314447 usecs after Sun Jan 20 21:52:58 2019
   Reason: Reset Requested by CLI command reload
   Service:
   Version: 9.3(3)

3) At 20142 usecs after Sun Jan 20 21:27:33 2019
   Reason: Reset Requested by CLI command reload
   Service:
   Version: 9.3(3)
```

After the switch reboots, the committed template is applied to all ASICs on the Cisco Nexus device. You cannot commit different templates to different ASICs on the Cisco Nexus device. All saved templates and committed templates along with the size of each region of each template are displayed in the running configuration.

When a template is committed, the software checks the following:

1. The combined size of all regions in the TCAM is 4096 entries.

2. The size of each region fits within the TCAM. At any point of time, there is always a running size for the TCAM region. This running size (the current size in the hardware TCAM) is defined by either the default or a user-defined template that was committed and is currently being used as the running template. If you increase the size of a region in a template that is currently being committed, from the current running size, the software checks if there are enough free entries outside the current region (entries that are not allocated to any other region) that can be used to increase the size of the region. If you decrease the size of a region in a template that is currently being committed from the current running size, the software checks to determine if there are enough free entries within the region that can be freed up to reduce the size of the TCAM region. All changes that reduce the sizes of the regions within the template are done before the changes to increase the sizes of regions within the template.

3. You cannot change the supervisor region size to be smaller than 256 entries because the software must have 256 entries to support all features in the sup-region.

4. The supervisor region default size 128 entries even though 256 entries are available. By using TCAM carving, the additional 128 entries can be used. The **sup** keyword is available in the CLI to change the values of the sup-region to 128, 192, or 256.

5. The hardware does not support more than 256 entries in the supervisor region and span regions. This check is done during validation.

If all these checks pass, you can commit he template and you are prompted to apply the template by rebooting.

If these checks fail, the commit fails and the template goes back to the saved state. If the commit fails, the **commit** command output displays the reasons that it failed.

You cannot modify or delete the default template. You can only move this template from saved to committed or committed to saved. If the default template is committed, it is not displayed in the running configuration. To apply the default template, enter the **no commit** command using the currently running template. Entering this command executes the same validation checks that were performed when you committed the template. If all validations succeed, the software prompts you to reboot the switch. If you agree to reboot, the template is saved in the startup configuration and the system is rebooted. After the reboot, the default template is applied. The startup configuration has the committed template that you committed before rebooting. After rebooting, the template in the startup configuration is used. If there is no committed template in the startup configuration, the default template is used.

You create and manage the TCAM carving templates by entering the template manager commands. The template-based TCAM carving CLI is supported in config-sync. Only template creation is supported inside config-sync. Template commit should be performed separately on each switch outside the config-sync context.

Modifying a User Defined Template

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **hardware profile tcam resource template** *template-name*
3. switch(config-tmpl)# {**vacl** *vacl-region* | **ifacl** *ifacl-region* | **qos** *qos-region* | **rbacl** *rbacl-region* | **span** *span-region*}
4. switch(config-tmpl)# {**vacl** *vacl-region* | **ifacl** *ifacl-region* | **qos** *qos-region* | **rbacl** *rbacl-region* | **span** *span-region* **iracl** *iracl-region* **eracl** *eracl-region* **sup** *sup-region* **iracl** *iracl-region* **eracl** *eracl-region* **sup** *sup-region* }

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **hardware profile tcam resource template** *template-name*	Creates a new template with the default region sizes. A maximum of 16 templates (plus the default) can be created. Use this command to enter template mode.
Step 3	switch(config-tmpl)# {**vacl** *vacl-region*	**ifacl** *ifacl-region*	**qos** *qos-region*	**rbacl** *rbacl-region*	**span** *span-region*}	Sets the region block size.

	Command or Action	Purpose
		• *vacl-region*—The block size of the region can be 64 to 3584.
		• *ifacl-region*—The block size of the region can be 320 to 3584.
		• *qos-region*—The block size of the region can be 64 to 3584.
		• *rbacl-region*—The block size of the region can be 64 to 3584.
		• *span-region*—The block size of the region can be 64 to 256.
		Note You cannot set the size of a region to zero. The block size must be a multiple of 64.
Step 4	switch(config-tmpl)# {**vacl** *vacl-region* | **ifacl** *ifacl-region* | **qos** *qos-region* | **rbacl** *rbacl-region* | **span** *span-region* **iracl** *iracl-region* **eracl** *eracl-region* **sup** *sup-region* **iracl** *iracl-region* **eracl** *eracl-region* **sup** *sup-region* }	Sets the region block size. • *vacl-region*—The block size of the region can be 64 to 3584. • *ifacl-region*—The block size of the region can be 320 to 3584. • *qos-region*—The block size of the region can be 64 to 3584. • *rbacl-region*—The block size of the region can be 64 to 3584. • *span-region*—The block size of the region can be 64 to 256. • *iracl-region*—The block size of the region can be 64 to 3648. • *eracl-region*—The block size of the region can be 64 to 3648. • *sup-region*—The block size of the region can be 64 to 256. **Note** You cannot set the size of a region to zero. The block size must be a multiple of 64. The block size for iracl and eracl should add upto 3712.

Example

This example shows how to modify a user-defined qos template.

```
switch# configure terminal
switch(config)# hardware profile tcam resource template qos-template
switch(config-tmpl) qos 64
```

Information About User-Defined Templates

In addition to the default template, you can create a maximum of 16 templates (which means that you can have 17 templates at one time). You can specify whatever sizes of ternary content addressable memory (TCAM) regions you want.

You can apply the following operations on each template:

- Create

- Modify

- Delete

- Commit

Each template can be in one of the following states:

- Saved

- Committed

Create

When you create a template, the size of the TCAM regions are initialized to the default values. When a template is created, the template is in the saved state by default. Once you create a template, you can modify it to change the size of any TCAM region. You should configure the size of the region in multiples of 64 because the size of each TCAM block is 64 entries. If you enter a value that is not a multiple of 64, an error message asks you to enter the value again.

Modify

You can modify any saved template to change the size of any TCAM region but you cannot modify the size of any region in the TCAM to 0. During the modification, the software checks that the size that you entered is on a 64 boundary. When you modify a template, the combined size of all the TCAM regions might have fewer than 4096 entries. During a modification, the software does not check that you have fewer than 4096 entries.

You can modify a template only when it is in the saved state. After a template is committed, you cannot modify it.

A user-defined committed template can be changed to the created state by servicing another user-defined template or default template.

To service another user-defined template, enter the following command:

hardware profile tcam resource service-template *user-defined-template*

To service a default template, enter the following command:

no hardware profile tcam resource service-template *currently-committed- template*

Delete

You can delete any saved template. After you delete a template, all information about the template is lost. A committed template cannot be deleted.

A user-defined committed template can be changed to the created state by servicing another user-defined template or default template.

To service another user-defined template, enter the following command:

hardware profile tcam resource service-template *user-defined-template*

To service a default template, enter the following command:

no hardware profile tcam resource service-template *currently-committed- template*

Commit

You can commit any of your user-defined templates or the default template that is provided by the software. To commit a template, enter the **commit** command and perform a reboot of the switch. When you enter the **commit** command, the software validates the template. If the validation is successful, the software prompts you to reboot the switch. The template (user defined or default) is applied after the reboot. If you did not choose to reboot, no changes are made to the TCAM regions and no template is committed.

From Cisco NX-OS Release 9.3(3) onwards, after you commit a template, the system prompts you whether to proceed with copying the running configuration to the startup configuration and rebooting the switch. After you agree to continue, the following occurs:

- The committed template is saved in the startup configuration.

- The switch is rebooted.

- The committed template is used by the software.

- The template goes to the running state.

> **Note** Prior to Cisco NX-OS Release 9.3(3), after you commit a template, the system does not automatically reboot but a message is displayed in the **commit** command output asking you to reboot the switch for the committed template to take effect.

If you perform a write erase, reload, and copy running configuration from a back-up configuration containing uncommitted TCAM profile, the following occurs:

1. After the TCAM profile is committed, switch automatically reloads without any prompt.

2. Any configuration after TCAM carving CLI is not applied.

3. To restore configuration with the committed TCAM profile, you need to copy backup configuration to running configuration again. However, there is no switch reload as the TCAM carving profile is already committed.

When the switch is reloaded due to the new committed TCAM profile, the **show system reset-reason** command displays the reason for the reload as shown below:

```
switch# show system reset-reason
----- reset reason for Supervisor-module 1 (from Supervisor in slot 1) ---
1) At 302777 usecs after Sun Jan 20 22:02:37 2019
```

```
    Reason: Reload due to change in TCAM service-template
    Service:
    Version: 9.3(3)

2) At 314447 usecs after Sun Jan 20 21:52:58 2019
   Reason: Reset Requested by CLI command reload
   Service:
   Version: 9.3(3)

3) At 20142 usecs after Sun Jan 20 21:27:33 2019
   Reason: Reset Requested by CLI command reload
   Service:
   Version: 9.3(3)
```

After the switch reboots, the committed template is applied to all ASICs on the Cisco Nexus device. You cannot commit different templates to different ASICs on the Cisco Nexus device. All saved templates and committed templates along with the size of each region of each template are displayed in the running configuration.

When a template is committed, the software checks the following:

1. The combined size of all regions in the TCAM is 4096 entries.

2. The size of each region fits within the TCAM. At any point of time, there is always a running size for the TCAM region. This running size (the current size in the hardware TCAM) is defined by either the default or a user-defined template that was committed and is currently being used as the running template. If you increase the size of a region in a template that is currently being committed, from the current running size, the software checks if there are enough free entries outside the current region (entries that are not allocated to any other region) that can be used to increase the size of the region. If you decrease the size of a region in a template that is currently being committed from the current running size, the software checks to determine if there are enough free entries within the region that can be freed up to reduce the size of the TCAM region. All changes that reduce the sizes of the regions within the template are done before the changes to increase the sizes of regions within the template.

3. You cannot change the supervisor region size to be smaller than 256 entries because the software must have 256 entries to support all features in the sup-region.

4. The supervisor region default size 128 entries even though 256 entries are available. By using TCAM carving, the additional 128 entries can be used. The **sup** keyword is available in the CLI to change the values of the sup-region to 128, 192, or 256.

5. The hardware does not support more than 256 entries in the supervisor region and span regions. This check is done during validation.

If all these checks pass, you can commit he template and you are prompted to apply the template by rebooting.

If these checks fail, the commit fails and the template goes back to the saved state. If the commit fails, the **commit** command output displays the reasons that it failed.

You cannot modify or delete the default template. You can only move this template from saved to committed or committed to saved. If the default template is committed, it is not displayed in the running configuration. To apply the default template, enter the **no commit** command using the currently running template. Entering this command executes the same validation checks that were performed when you committed the template. If all validations succeed, the software prompts you to reboot the switch. If you agree to reboot, the template is saved in the startup configuration and the system is rebooted. After the reboot, the default template is applied. The startup configuration has the committed template that you committed before rebooting. After rebooting, the template in the startup configuration is used. If there is no committed template in the startup configuration, the default template is used.

You create and manage the TCAM carving templates by entering the template manager commands. The template-based TCAM carving CLI is supported in config-sync. Only template creation is supported inside config-sync. Template commit should be performed separately on each switch outside the config-sync context.

Committing a User-Defined Template

You can commit a user-defined template.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **hardware profile tcam resource service-template** *template-name*
3. (Optional) switch# **show hardware profile tcam resource template**

DETAILED STEPS

	Command or Action	Purpose
Step 1	Required: switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **hardware profile tcam resource service-template** *template-name*	Commits a previously defined template in the running image. After you commit a template, the system prompts you whether to proceed with copying the running configuration to the startup configuration and rebooting the switch. If you agree to continue, the specified template is applied after the reboot. Otherwise, no changes are made to the TCAM regions and no template is committed.
Step 3	(Optional) switch# **show hardware profile tcam resource template**	Displays all templates.
		Note After the switch reloads, use this command to display the committed template.

Example

This example shows how to commit a user-defined template:

```
switch# configure terminal
switch(config)# hardware profile tcam resource service-template temp1

Details of the temp1 template you are trying to commit are as follows:

-------------------------------------------------------------------------------
Template name: temp1
Current state: Created

Region  Features  Size-allocated   Current-size  Current-usage  Available/free
-------------------------------------------------------------------------------
Vacl    Vacl               1024           1024             15            1009
Ifacl   Ifacl              1152           1152            209             943
Rbacl   Rbacl              1152           1152              3            1149
Qos     Qos                 448            448             30             418
Span    Span                 64             64              2              62
Sup     Sup                 256            256             58             198
```

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


To finish committing the template, the system will do the following:
    1> Save running config :  "copy running-config startup-config"
    2> Reboot the switch   :  "reload"


--------------------------------------------------------------------------------
Do you really want to continue with RELOAD ? (y/n) [no] yes
System is still initializing
Configuration mode is blocked until system is ready
switch(config)# [16152.925385] Shutdown Ports..
[16152.959744]  writing reset reason 9
[snip]


/AFTER SWITCH RELOADS/

switch# show hardware profile tcam resource template
    Template  Type      State    Vacl  Ifacl  Rbacl  Qos  Span  Sup    TOTAL
--------------------------------------------------------------------------------
    default system   Created   1024  1152   1152   448   64   256    4096
      temp1   user Committed   1024  1152   1152   448   64   256    4096
      temp2   user   Created   1024  1152   1152   448   64   256    4096
--------------------------------------------------------------------------------
```

This example shows how to commit and apply a user-defined template for a Layer 3 Card-facing UPC:

```
switch# configure terminal
switch(config)# hardware profile tcam resource service-template temp1

Details of the temp1 template you are trying to commit are as follows:

--------------------------------------------------------------------------------
Template name: temp1
Current state: Created

Region  Features  Size-allocated   Current-size  Current-usage  Available/free
--------------------------------------------------------------------------------
Vacl    Vacl            1984           2048             11             2037
Ifacl   Ifacl           1216           1152             26             1126
Rbacl   Rbacl            128            128              3              125
Qos     Qos              448            448              9              439
Span    Span              64             64              3               61
Sup     Sup              256            128             81               47
ERacl   ERacl           1920              0              0                0
IRacl   IRacl           1792              0              0                0
--------------------------------------------------------------------------------


To finish committing the template, the system will do the following:
    1> Save running config :  "copy running-config startup-config"
    2> Reboot the switch   :  "reload"


--------------------------------------------------------------------------------
Do you really want to continue with RELOAD ? (y/n) [no] yes
System is still initializing
Configuration mode is blocked until system is ready
5548(config)# [166850.680711] Shutdown Ports..
[166850.716114]  writing reset reason 9,
```

```
[snip]

/AFTER SWITCH RELOADS/

switch# show hardware profile tcam resource template
    Template   Type     State    ERacl  Ifacl  IRacl   Qos   Span   Sup    TOTAL
--------------------------------------------------------------------------------
    default  system   Created   2048     64   1664    64    64    64     4096
      temp1    user Committed   1920     64   1792    64    64    64     4096
      temp2    user   Created   2048     64   1664    64    64    64     4096
--------------------------------------------------------------------------------
```

Information About User-Defined Templates

In addition to the default template, you can create a maximum of 16 templates (which means that you can have 17 templates at one time). You can specify whatever sizes of ternary content addressable memory (TCAM) regions you want.

You can apply the following operations on each template:

- Create

- Modify

- Delete

- Commit

Each template can be in one of the following states:

- Saved

- Committed

Create

When you create a template, the size of the TCAM regions are initialized to the default values. When a template is created, the template is in the saved state by default. Once you create a template, you can modify it to change the size of any TCAM region. You should configure the size of the region in multiples of 64 because the size of each TCAM block is 64 entries. If you enter a value that is not a multiple of 64, an error message asks you to enter the value again.

Modify

You can modify any saved template to change the size of any TCAM region but you cannot modify the size of any region in the TCAM to 0. During the modification, the software checks that the size that you entered is on a 64 boundary. When you modify a template, the combined size of all the TCAM regions might have fewer than 4096 entries. During a modification, the software does not check that you have fewer than 4096 entries.

You can modify a template only when it is in the saved state. After a template is committed, you cannot modify it.

A user-defined committed template can be changed to the created state by servicing another user-defined template or default template.

To service another user-defined template, enter the following command:

hardware profile tcam resource service-template *user-defined-template*

To service a default template, enter the following command:

no hardware profile tcam resource service-template *currently-committed- template*

Delete

You can delete any saved template. After you delete a template, all information about the template is lost. A committed template cannot be deleted.

A user-defined committed template can be changed to the created state by servicing another user-defined template or default template.

To service another user-defined template, enter the following command:

hardware profile tcam resource service-template *user-defined-template*

To service a default template, enter the following command:

no hardware profile tcam resource service-template *currently-committed- template*

Commit

You can commit any of your user-defined templates or the default template that is provided by the software. To commit a template, enter the **commit** command and perform a reboot of the switch. When you enter the **commit** command, the software validates the template. If the validation is successful, the software prompts you to reboot the switch. The template (user defined or default) is applied after the reboot. If you did not choose to reboot, no changes are made to the TCAM regions and no template is committed.

From Cisco NX-OS Release 9.3(3) onwards, after you commit a template, the system prompts you whether to proceed with copying the running configuration to the startup configuration and rebooting the switch. After you agree to continue, the following occurs:

- The committed template is saved in the startup configuration.

- The switch is rebooted.

- The committed template is used by the software.

- The template goes to the running state.

> **Note** Prior to Cisco NX-OS Release 9.3(3), after you commit a template, the system does not automatically reboot but a message is displayed in the **commit** command output asking you to reboot the switch for the committed template to take effect.

If you perform a write erase, reload, and copy running configuration from a back-up configuration containing uncommitted TCAM profile, the following occurs:

1. After the TCAM profile is committed, switch automatically reloads without any prompt.

2. Any configuration after TCAM carving CLI is not applied.

3. To restore configuration with the committed TCAM profile, you need to copy backup configuration to running configuration again. However, there is no switch reload as the TCAM carving profile is already committed.

When the switch is reloaded due to the new committed TCAM profile, the **show system reset-reason** command displays the reason for the reload as shown below:

```
switch# show system reset-reason
----- reset reason for Supervisor-module 1 (from Supervisor in slot 1) ---
1) At 302777 usecs after Sun Jan 20 22:02:37 2019
   Reason: Reload due to change in TCAM service-template
   Service:
   Version: 9.3(3)

2) At 314447 usecs after Sun Jan 20 21:52:58 2019
   Reason: Reset Requested by CLI command reload
   Service:
   Version: 9.3(3)

3) At 20142 usecs after Sun Jan 20 21:27:33 2019
   Reason: Reset Requested by CLI command reload
   Service:
   Version: 9.3(3)
```

After the switch reboots, the committed template is applied to all ASICs on the Cisco Nexus device. You cannot commit different templates to different ASICs on the Cisco Nexus device. All saved templates and committed templates along with the size of each region of each template are displayed in the running configuration.

When a template is committed, the software checks the following:

1. The combined size of all regions in the TCAM is 4096 entries.

2. The size of each region fits within the TCAM. At any point of time, there is always a running size for the TCAM region. This running size (the current size in the hardware TCAM) is defined by either the default or a user-defined template that was committed and is currently being used as the running template. If you increase the size of a region in a template that is currently being committed, from the current running size, the software checks if there are enough free entries outside the current region (entries that are not allocated to any other region) that can be used to increase the size of the region. If you decrease the size of a region in a template that is currently being committed from the current running size, the software checks to determine if there are enough free entries within the region that can be freed up to reduce the size of the TCAM region. All changes that reduce the sizes of the regions within the template are done before the changes to increase the sizes of regions within the template.

3. You cannot change the supervisor region size to be smaller than 256 entries because the software must have 256 entries to support all features in the sup-region.

4. The supervisor region default size 128 entries even though 256 entries are available. By using TCAM carving, the additional 128 entries can be used. The **sup** keyword is available in the CLI to change the values of the sup-region to 128, 192, or 256.

5. The hardware does not support more than 256 entries in the supervisor region and span regions. This check is done during validation.

If all these checks pass, you can commit he template and you are prompted to apply the template by rebooting.

If these checks fail, the commit fails and the template goes back to the saved state. If the commit fails, the **commit** command output displays the reasons that it failed.

You cannot modify or delete the default template. You can only move this template from saved to committed or committed to saved. If the default template is committed, it is not displayed in the running configuration. To apply the default template, enter the **no commit** command using the currently running template. Entering this command executes the same validation checks that were performed when you committed the template.

If all validations succeed, the software prompts you to reboot the switch. If you agree to reboot, the template is saved in the startup configuration and the system is rebooted. After the reboot, the default template is applied. The startup configuration has the committed template that you committed before rebooting. After rebooting, the template in the startup configuration is used. If there is no committed template in the startup configuration, the default template is used.

You create and manage the TCAM carving templates by entering the template manager commands. The template-based TCAM carving CLI is supported in config-sync. Only template creation is supported inside config-sync. Template commit should be performed separately on each switch outside the config-sync context.

Deleting a Template

After creating a template, the template can be deleted. Deleting removes all the information about the template from the software.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **no hardware profile tcam resource template** *template-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **no hardware profile tcam resource template** *template-name*	Deletes a user-defined template. Only saved templates can be deleted. Templates that are committed/running cannot be deleted. A template that is in the running configuration (same as the startup configuration) cannot be deleted. Any other user-defined template that is in a saved state can be deleted. The default template cannot be deleted.

Example

This example shows how to delete a template:

```
switch# configure terminal
switch(config)# no hardware profile tcam resource template qos-template
```

Information About User-Defined Templates

In addition to the default template, you can create a maximum of 16 templates (which means that you can have 17 templates at one time). You can specify whatever sizes of ternary content addressable memory (TCAM) regions you want.

You can apply the following operations on each template:

- Create

- Modify

- Delete

- Commit

Each template can be in one of the following states:

- Saved

- Committed

Create

When you create a template, the size of the TCAM regions are initialized to the default values. When a template is created, the template is in the saved state by default. Once you create a template, you can modify it to change the size of any TCAM region. You should configure the size of the region in multiples of 64 because the size of each TCAM block is 64 entries. If you enter a value that is not a multiple of 64, an error message asks you to enter the value again.

Modify

You can modify any saved template to change the size of any TCAM region but you cannot modify the size of any region in the TCAM to 0. During the modification, the software checks that the size that you entered is on a 64 boundary. When you modify a template, the combined size of all the TCAM regions might have fewer than 4096 entries. During a modification, the software does not check that you have fewer than 4096 entries.

You can modify a template only when it is in the saved state. After a template is committed, you cannot modify it.

A user-defined committed template can be changed to the created state by servicing another user-defined template or default template.

To service another user-defined template, enter the following command:

hardware profile tcam resource service-template *user-defined-template*

To service a default template, enter the following command:

no hardware profile tcam resource service-template *currently-committed- template*

Delete

You can delete any saved template. After you delete a template, all information about the template is lost. A committed template cannot be deleted.

A user-defined committed template can be changed to the created state by servicing another user-defined template or default template.

To service another user-defined template, enter the following command:

hardware profile tcam resource service-template *user-defined-template*

To service a default template, enter the following command:

no hardware profile tcam resource service-template *currently-committed- template*

Commit

You can commit any of your user-defined templates or the default template that is provided by the software. To commit a template, enter the **commit** command and perform a reboot of the switch. When you enter the **commit** command, the software validates the template. If the validation is successful, the software prompts you to reboot the switch. The template (user defined or default) is applied after the reboot. If you did not choose to reboot, no changes are made to the TCAM regions and no template is committed.

From Cisco NX-OS Release 9.3(3) onwards, after you commit a template, the system prompts you whether to proceed with copying the running configuration to the startup configuration and rebooting the switch. After you agree to continue, the following occurs:

- The committed template is saved in the startup configuration.

- The switch is rebooted.

- The committed template is used by the software.

- The template goes to the running state.

> **Note** Prior to Cisco NX-OS Release 9.3(3), after you commit a template, the system does not automatically reboot but a message is displayed in the **commit** command output asking you to reboot the switch for the committed template to take effect.

If you perform a write erase, reload, and copy running configuration from a back-up configuration containing uncommitted TCAM profile, the following occurs:

1. After the TCAM profile is committed, switch automatically reloads without any prompt.

2. Any configuration after TCAM carving CLI is not applied.

3. To restore configuration with the committed TCAM profile, you need to copy backup configuration to running configuration again. However, there is no switch reload as the TCAM carving profile is already committed.

When the switch is reloaded due to the new committed TCAM profile, the **show system reset-reason** command displays the reason for the reload as shown below:

```
switch# show system reset-reason
----- reset reason for Supervisor-module 1 (from Supervisor in slot 1) ---
1) At 302777 usecs after Sun Jan 20 22:02:37 2019
    Reason: Reload due to change in TCAM service-template
    Service:
    Version: 9.3(3)

2) At 314447 usecs after Sun Jan 20 21:52:58 2019
    Reason: Reset Requested by CLI command reload
    Service:
    Version: 9.3(3)

3) At 20142 usecs after Sun Jan 20 21:27:33 2019
    Reason: Reset Requested by CLI command reload
    Service:
    Version: 9.3(3)
```

After the switch reboots, the committed template is applied to all ASICs on the Cisco Nexus device. You cannot commit different templates to different ASICs on the Cisco Nexus device. All saved templates and

committed templates along with the size of each region of each template are displayed in the running configuration.

When a template is committed, the software checks the following:

1. The combined size of all regions in the TCAM is 4096 entries.

2. The size of each region fits within the TCAM. At any point of time, there is always a running size for the TCAM region. This running size (the current size in the hardware TCAM) is defined by either the default or a user-defined template that was committed and is currently being used as the running template. If you increase the size of a region in a template that is currently being committed, from the current running size, the software checks if there are enough free entries outside the current region (entries that are not allocated to any other region) that can be used to increase the size of the region. If you decrease the size of a region in a template that is currently being committed from the current running size, the software checks to determine if there are enough free entries within the region that can be freed up to reduce the size of the TCAM region. All changes that reduce the sizes of the regions within the template are done before the changes to increase the sizes of regions within the template.

3. You cannot change the supervisor region size to be smaller than 256 entries because the software must have 256 entries to support all features in the sup-region.

4. The supervisor region default size 128 entries even though 256 entries are available. By using TCAM carving, the additional 128 entries can be used. The **sup** keyword is available in the CLI to change the values of the sup-region to 128, 192, or 256.

5. The hardware does not support more than 256 entries in the supervisor region and span regions. This check is done during validation.

If all these checks pass, you can commit he template and you are prompted to apply the template by rebooting.

If these checks fail, the commit fails and the template goes back to the saved state. If the commit fails, the **commit** command output displays the reasons that it failed.

You cannot modify or delete the default template. You can only move this template from saved to committed or committed to saved. If the default template is committed, it is not displayed in the running configuration. To apply the default template, enter the **no commit** command using the currently running template. Entering this command executes the same validation checks that were performed when you committed the template. If all validations succeed, the software prompts you to reboot the switch. If you agree to reboot, the template is saved in the startup configuration and the system is rebooted. After the reboot, the default template is applied. The startup configuration has the committed template that you committed before rebooting. After rebooting, the template in the startup configuration is used. If there is no committed template in the startup configuration, the default template is used.

You create and manage the TCAM carving templates by entering the template manager commands. The template-based TCAM carving CLI is supported in config-sync. Only template creation is supported inside config-sync. Template commit should be performed separately on each switch outside the config-sync context.

Verifying the TCAM Carving Configuration

To display TCAM carving configuration information, enter one of the following commands:

Command	Purpose
show hardware profile tcam resource template	Displays all templates.

Command	Purpose
show hardware profile tcam resource template name *template-name*	Displays a user-defined template.
show hardware profile tcam resource template default	Displays a default template.

Verifying the FCoE Configuration

To verify FCoE configuration information, perform one of these tasks:

Command	Purpose
switch# **show fcoe**	Displays whether FCoE is enabled on the switch.
switch# **show fcoe database**	Displays the contents of the FCoE database.
switch# **show interface** [*interface number*] **fcoe**	Displays the FCoE settings for an interface or all interfaces.
switch# **show queuing interface**[*interface slot/port*]	Displays the queue configuration and statistics.
switch# **show policy-map interface**[*interface number*]	Displays the policy map settings for an interface or all interfaces.

This example shows how to verify that the FCoE capability is enabled:

```
switch# show fcoe

Global FCF details

        FCF-MAC is 00:0d:ec:6d:95:00

        FC-MAP is 0e:fc:00

        FCF Priority is 128

        FKA Advertisement period for FCF is 8 seconds
```

This example shows how to display the FCoE database:

```
switch# show fcoe database

-----------------------------------------------------------------------------

INTERFACE       FCID            PORT NAME               MAC ADDRESS

-----------------------------------------------------------------------------

vfc3            0x490100        21:00:00:1b:32:0a:e7:b8 00:c0:dd:0e:5f:76
```

This example shows how to display the FCoE settings for an interface.

```
switch# show interface ethernet 1/37 fcoe

Ethernet1/37 is FCoE UP

    vfc3 is Up

        FCID is 0x490100
```

```
PWWN is 21:00:00:1b:32:0a:e7:b8

MAC addr is 00:c0:dd:0e:5f:76
```