Configure VXLAN Cross Connect

VXLAN cross connect

VXLAN cross connect is a tunneling mechanism used on VXLAN fabric which

- provides point-to-point data tunneling,
- control packets such as CDP, LLDP, LACP, STP, BFD, and PAGP from one VTEP to another, and
- often referred to as **xConnect**.

BGP EVPN signaling identifies endpoints based on how the provider VNI is stretched in the fabric. Every attachment circuit will be part of a unique provider VNI. All inner customer tags will be preserved, as is, and packets are encapsulated in the provider VNI using **dot1q-tunnel**. On the decapsulation endpoint, the provider VNI will forward the packet to its attachment circuit while preserving all customer dot1q tags in the packets.

Supported xConnect tunnel combinations

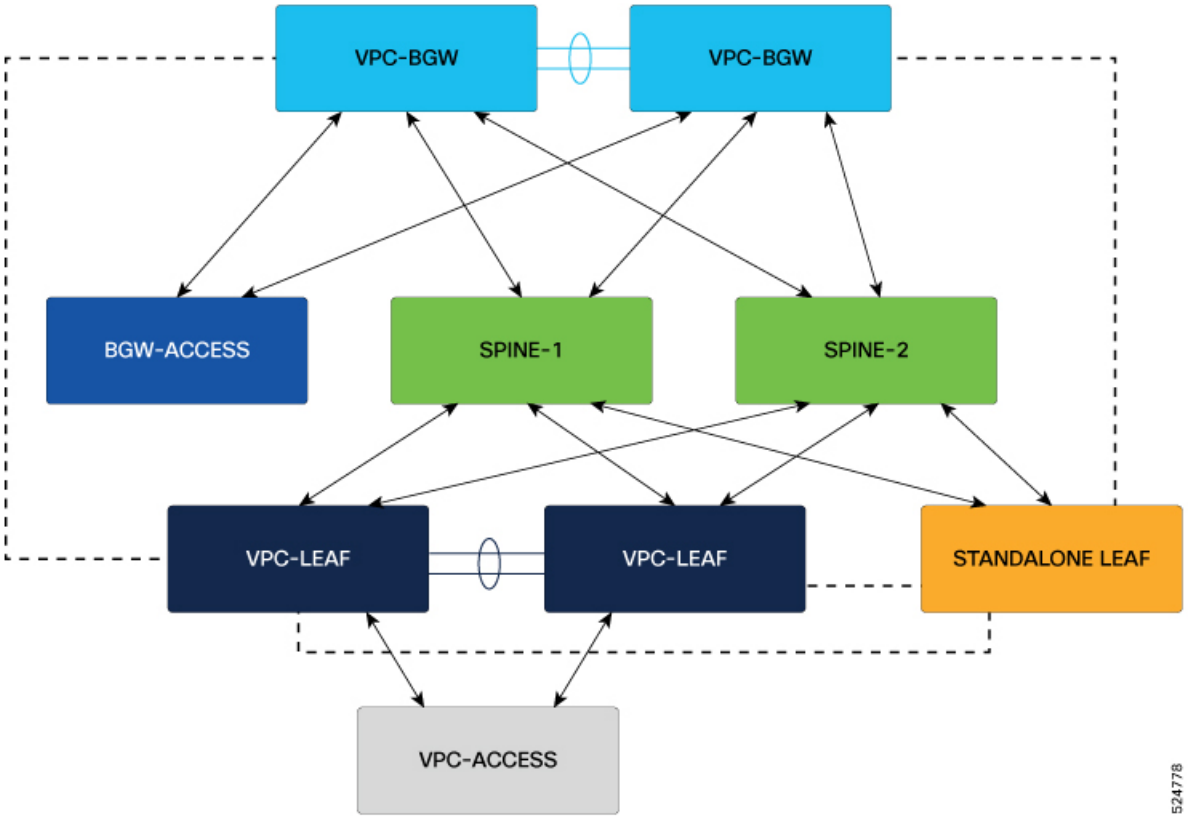
The supported tunnel combinations for VXLAN cross connect are

- physical interface to physical interface,
- port-channel to port-channel,
- mixed combinations of physical interface and port-channel, and
- additionally, this is supported over both underlay types (ingress replication and multicast) or a combination of both.

Topology of VXLAN Cross Connect

The following figure represents the **NGOAM xConnect** configuration on a VXLAN fabric. NGOAM xConnect is used to establish **pseudo-wire tunnels between VTEPs (VXLAN Tunnel Endpoints)**, which helps in

- proactive failure detection across the network fabric, and
- monitoring the enabled tunnels end-to-end connectivity across the fabric.



The dashed line in the diagram represents the NGOAM xConnect pseudo-wire, which is implemented between specific nodes to detect and react to failures.

Guidelines and Limitations of VXLAN Cross Connect

Generic guidelines and limitations of VXLAN Cross Connect

- MAC learning will be disabled on the xConnect VNIs and none of the host MAC will be learned on the tunnel access ports.
- Only one attachment circuit can be configured for a provider VNI on a given VTEP.
- A VNI can only be stretched in a point-to-point fashion. Point-to-multipoint is not supported.

- The scale of xConnect VLANs depends on the number of ports available on the switch. Every xConnect VLAN can tunnel all 4k customer VLANs.
- On vPC VTEP, cross connect needs backup SVI as native VLAN on the vPC peer-link.
- On vPC VTEP, spanning tree must be disabled on both vPC peers for xConnect VLANs.

ISSU and patch guidelines and limitations of VXLAN Cross Connect

- When an upgrade is performed non-disruptively from Cisco NX-OS Release 7.0(3)I7(4) to Cisco NX-OS Release 9.2(x) code, and if a VLAN is created and configured as xConnect, you must enter the **copy running-config startup-config** command and reload the switch. If the box was upgraded disruptively to Cisco NX-OS Release 9.2(x) code, a reload is not needed on configuring a VLAN as xConnect.
- Make sure that the **ngoam xconnect hb-interval** is set to 5000 milliseconds on all VTEPs before attempting ISSU/patch activation to avoid link flaps.
- Before activating the patch for the cfs process, you must move the NGOAM xConnect hb-interval to the maximum value of 5000 milliseconds. This prevents interface flaps during the patch activation.

Configuration and port guidelines and limitations of VXLAN Cross Connect

- If ARP Suppression is enabled on a VLAN, and you enable xConnect on the VLAN, the xConnect feature takes precedence.
- xConnect enabled VLANs are accessible only through switch port access mode.
- Configuring a static MAC on xConnect tunnel interfaces is not supported.
- The vPC orphan tunneled port per VNI should be either on the vPC primary switch or secondary switch, but not both.
- xConnect is not supported on FEX ports.
- xConnect access ports must be flapped after disabling NGOAM on all the VTEPs.
- After deleting and adding a VLAN, or removing xConnect from a VLAN, physical ports must be flapped with NGOAM.

Supported Features of VXLAN Cross Connect

- Supported only on BGP EVPN topology.
- Supported on vPC fabric peering.
- Does not support LACP bundling of attachment circuits.
- Does not support SVI on an xconnect VLAN.
- Does not support multi-site solution.

Supported platform and release of VXLAN cross connect

SupportedRelease	SupportedPlatform
9.3(3) and later	CiscoNexus 9300-FX/FX2/GX Series switches
9.3(5) and later	CiscoNexus 9300-FX3 Series switches
10.2(3)F and later	CiscoNexus 9300-GX2 Series switches
10.4(1)F and later	CiscoNexus 9332D-H2R switches
10.4(2)F and later	CiscoNexus 93400LD-H1 switches
10.4(3)F and later	CiscoNexus 9364C-H1 switches

Prerequisites

Result

- Before you begin, verify that the **feature ngoam** is enabled. For more information on how to enable, see .

```
switch# configure terminal
switch(config)# feature ngoam
```

- (Optional) Use the **ngoam xconnect hb-interval** *interval* command in global configuration mode, only if user wants to configure non-default heartbeat interval. Range:150 to 5000. Default: 190.

```
switch(config)# ngoam xconnect hb-interval 200
```

Configure the VXLAN Cross Connect

Use this procedure to configure the VXLAN Cross Connect feature on Cisco Nexus 9000 devices.

Procedure

Step 1 Follow these steps to create and configure xConnect VLANs:

a) Enter the **vlan** *session-num* command in global configuration mode, to create a VLAN.

Example:

```
switch(config)# vlan 550
switch(config-vlan)#
```

b) Enter the **vn-segment** *segment-num* command in VLAN configuration mode, to specify VXLAN VNID (Virtual Network Identifier).

Example:

```
switch(config-vlan)# vn-segment 5555
```

c) Enter the **xconnect** command in VLAN configuration mode, to define the provider VLAN with the attached VNI to be in cross connect mode.

Example:

```
switch(config-vlan)# xconnect
```

d) Enter the **exit** command in VLAN configuration mode, to exit the config mode and apply the configuration to the switch.

Example:

```
switch(config-vlan)# exit
switch(config)#
```

Step 2 Follow these steps to configure xConnect on a interface access ports:

a) Enter the **interface** [*physical-interface* | *port-channel*] command in global configuration mode, to create a VLAN.

Example:

For physical interface:

```
switch(config)# interface Ethernet1/30/3
```

For port-channel:

```
switch(config)# interface port-channel 550
```

b) Enter the **switchport** command in interface configuration mode, to enable the switchport mode.

Example:

```
switch(config-if)# switchport
```

c) Enter the **switchport mode dot1q-tunnel** command in interface configuration mode, to create a 802.1Q tunnel on the port.

Example:

```
switch(config-if)# switchport mode dot1q-tunnel
```

d) Enter the **switchport access vlan** *vlan-id* command in interface configuration mode, to specify the port assigned to a xConnect VLAN.

Example:

```
switch(config-if)# switchport access vlan 660
```

e) Enter the **exit** command to exit the config mode and apply the configuration to the switch.

Example:

```
switch(config-if)# exit
switch(config)#
```

Step 3 Run the **show running-config vlan** *session-num* command to verify the VLAN information.

Example:

```
switch# show running-config vlan 550

!Command: show running-config vlan 550
!Running configuration last done at: Sun Feb 23 16:34:03 2025
!Time: Mon Feb 24 13:53:01 2025

version 10.5(2) Bios:version 05.51
vlan 550
```

```
vlan 550
  vn-segment 5555
  xconnect
```

Step 4 Run the **show running-config interface port-channel** *pc-num* command to verify interface port channel information.

Example:

```
switch# show running-config interface port-channel 550

!Command: show running-config interface port-channel550
!Running configuration last done at: Sun Feb 23 16:34:03 2025
!Time: Mon Feb 24 13:53:30 2025

version 10.5(2) Bios:version 05.51

interface port-channel550
  switchport
  switchport mode dot1q-tunnel
  switchport access vlan 550
  spanning-tree bpdufilter enable
switch#
```

Step 5 Run the **show running-config interface Ethernet** *port-num* command to verify interface access ports information.

Example:

```
switch# show running-config interface Ethernet1/30/3

!Command: show running-config interface port-channel550
!Running configuration last done at: Sun Feb 23 16:34:03 2025
!Time: Mon Feb 24 13:53:30 2025

version 10.5(2) Bios:version 05.51

interface port-channel550
  switchport
  switchport mode dot1q-tunnel
  switchport access vlan 550
  spanning-tree bpdufilter enable
switch#
```

Verification of VXLAN Cross Connect Configuration

To display the status for the VXLAN Cross Connect configuration, enter one of the following commands:

Command	Purpose
show nve vni	Displays list of all VXLAN VNI and its status.
show nve vni *vlan-id*	Displays VNI status for specified VNI.
show ngoam xconnect session all	Displays all the xConnect session and its status.
show ngoam xconnect session *vlan-id*	Displays the specified xConnect session with details.

- Example of **show nve vni** command: Once the xConnect enabled VLANs are tagged on the NVE interface, you can see the **Conn** flag for the xConnect enabled VLANs.

```
switch# show nve vni
Codes: CP - Control Plane        DP - Data Plane
       UC - Unconfigured         SA - Suppress ARP
       S-ND - Suppress ND
       SU - Suppress Unknown Unicast
       Xconn - Crossconnect
       MS-IR - Multisite Ingress Replication
       HYB - Hybrid IRB mode

Interface VNI      Multicast-group   State Mode Type [BD/VRF]      Flags
--------- -------- ----------------- ----- ---- ------------------ -----
nve1      5555     225.5.5.5         Up    CP   L2 [550]           Xconn
nve1      6666     225.6.6.6         Up    CP   L2 [660]           Xconn
nve1      7777     225.7.7.7         Up    CP   L2 [770]           Xconn
nve1      10101    225.1.1.1         Up    CP   L2 [101]           SA
nve1      10102    225.1.1.2         Up    CP   L2 [102]           SA
nve1      10103    225.1.1.3         Up    CP   L2 [103]           SA
nve1      10104    225.1.1.4         Up    CP   L2 [104]           SA
nve1      10105    225.1.1.5         Up    CP   L2 [105]           SA
nve1      10106    225.1.1.6         Up    CP   L2 [106]           SA
nve1      10107    225.1.1.7         Up    CP   L2 [107]           SA
nve1      10108    225.1.1.8         Up    CP   L2 [108]           SA
nve1      10109    225.1.1.9         Up    CP   L2 [109]           SA
nve1      10110    225.1.1.10        Up    CP   L2 [110]           SA
nve1      20201    225.1.1.1         Up    CP   L2 [201]           SA
nve1      20202    225.1.1.2         Up    CP   L2 [202]           SA
nve1      20203    225.1.1.3         Up    CP   L2 [203]           SA
nve1      20204    225.1.1.4         Up    CP   L2 [204]           SA
nve1      20205    225.1.1.5         Up    CP   L2 [205]           SA
nve1      20206    225.1.1.6         Up    CP   L2 [206]           SA
nve1      20207    225.1.1.7         Up    CP   L2 [207]           SA
nve1      20208    225.1.1.8         Up    CP   L2 [208]           SA
nve1      20209    225.1.1.9         Up    CP   L2 [209]           SA
nve1      20210    225.1.1.10        Up    CP   L2 [210]           SA
nve1      202020   UnicastBGP        Up    CP   L2 [20]            Xconn
nve1      1001001  225.1.1.1         Up    CP   L2 [1001]          SA
nve1      1001002  225.1.1.2         Up    CP   L2 [1002]          SA
nve1      1001003  225.1.1.3         Up    CP   L2 [1003]          SA
nve1      1001004  225.1.1.4         Up    CP   L2 [1004]          SA
nve1      1001005  225.1.1.5         Up    CP   L2 [1005]          SA
nve1      1001006  225.1.1.6         Up    CP   L2 [1006]          SA
nve1      1001007  225.1.1.7         Up    CP   L2 [1007]          SA
nve1      1001008  225.1.1.8         Up    CP   L2 [1008]          SA
nve1      1001009  225.1.1.9         Up    CP   L2 [1009]          SA
nve1      1001010  225.1.1.10        Up    CP   L2 [1010]          SA
nve1      2002001  225.1.1.1         Up    CP   L2 [2001]          SA
nve1      2002002  225.1.1.2         Up    CP   L2 [2002]          SA
nve1      2002003  225.1.1.3         Up    CP   L2 [2003]          SA
nve1      2002004  225.1.1.4         Up    CP   L2 [2004]          SA
nve1      2002005  225.1.1.5         Up    CP   L2 [2005]          SA
nve1      2002006  225.1.1.6         Up    CP   L2 [2006]          SA
nve1      2002007  225.1.1.7         Up    CP   L2 [2007]          SA
nve1      2002008  225.1.1.8         Up    CP   L2 [2008]          SA
nve1      2002009  225.1.1.9         Up    CP   L2 [2009]          SA
nve1      2002010  225.1.1.10        Up    CP   L2 [2010]          SA
nve1      5005001  n/a               Up    CP   L3 [vxlan-5001]
nve1      5005002  n/a               Up    CP   L3 [vxlan-5002]
nve1      5005003  n/a               Up    CP   L3 [vxlan-5003]
nve1      5005004  n/a               Up    CP   L3 [vxlan-5004]
nve1      5005005  n/a               Up    CP   L3 [vxlan-5005]
nve1      5005006  n/a               Up    CP   L3 [vxlan-5006]
nve1      5005007  n/a               Up    CP   L3 [vxlan-5007]
nve1      5005008  n/a               Up    CP   L3 [vxlan-5008]
nve1      5005009  n/a               Up    CP   L3 [vxlan-5009]
```

```
nve1      5005010  n/a               Up    CP   L3 [vxlan-5010]
switch#
```

- Example of **show ngoam xconnect session all** command: use the command to check all the NGOAM xConnect session on the switch.

```
switch# show ngoam xconnect session all

States: LD = Local interface down, RD = Remote interface Down
          HB = Heartbeat lost, DB = Database/Routes not present
          * - Showing Vpc-peer interface info
Vlan           Peer-ip/vni        XC-State      Local-if/State        Rmt-if/State
================================================================================
20      100.100.100.8 / 202020      Active      Eth1/7/1 / UP       Eth1/50 / UP
550   200.200.200.200 / 5555       Active        Po550 / UP     Eth1/63/4 / UP
660     100.100.100.8 / 6666       Active     Eth1/30/3 / UP       Eth1/48 / UP
770   200.200.200.200 / 7777       Active        Po770 / UP         Po770 / UP
switch#
```

- Use the following commands to check the details of the specific NGOAM xConnect session on the switch.
  - The below example shows the output of vPC pairs as local interface and remote interface.

```
switch# sh ngoam xconnect session 550
Vlan ID: 550
Peer IP: 200.200.200.200  VNI : 5555
State: Active
Last state update: 02/23/2025 22:21:01.958
Local interface: Po550  State:  UP
Local vpc interface: Po550  State:  UP
Remote interface: Eth1/63/4  State:  UP
Remote vpc interface: Eth1/63/2  State:  UP
```

  - The below example shows the local interface as vPC and the remote interface as non-vPC pairs.

```
switch# sh ngoam xconnect session 660
Vlan ID: 660
Peer IP: 100.100.100.8  VNI : 6666
State: Active
Last state update: 02/23/2025 15:58:14.735
Local interface: Eth1/30/3  State:  UP
Local vpc interface: Eth1/30/3  State:  UP
Remote interface: Eth1/48  State:  UP
Remote vpc interface: Unknown  State:  DOWN
switch#
```

Remove a Cross Connect VNI

Use this procedure to remove the cross connect tag on the VLAN of Cisco Nexus switches.

Procedure

Step 1 Enter the **vlan** *session-num* command in global configurtion mode, to remove the VNI under NVE.

Example:

```
switch# configure terminal
switch(config)# vlan 550
switch(config-vlan)#
```

Step 2 Enter the **no xconnect** command in VLAN configurtion mode, to remove the xconnect tag from the VLAN configurations.

Example:

```
switch(config-vlan)# no xconnect
```

Note
Removing the xconnect tag is mandatory before removing the vn-segment for the xconnect enabled VLANs. Otherwise, you can see the below syslog that indicates the procedure.

```
xconnect is enabled on vlan 550, please disable xconnect before removing vn-segment
Cannot run commands in the mode at this moment. Please try again.
```

Step 3 Enter the **no vn-segment** *segment-num* command in VLAN configurtion mode, to remove the VNI.

Example:

```
switch(config-vlan)# no vn-segment
```

Step 4 Enter the **no vlan** *session-num* command in VLAN configurtion mode, to remove the VLAN.

Example:

```
switch(config-vlan)# no vlan 550
```