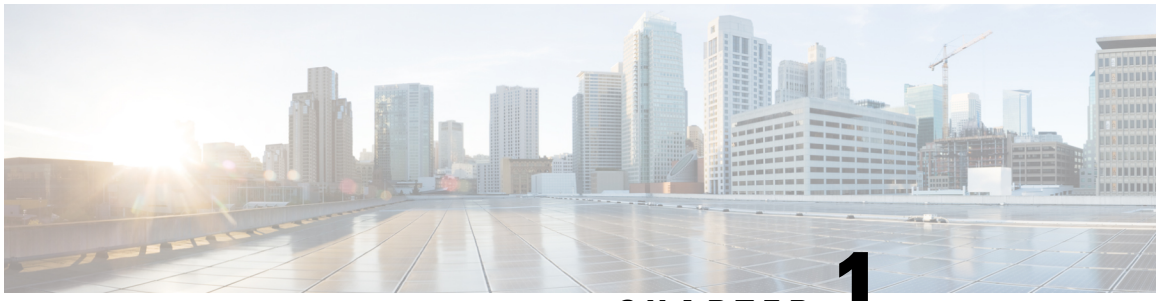
Cisco Smart Switches Configuration Guide for DPU Security Mode, Release 10.6(x)

First Published: 2026-08-23

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

CONTENTS



CHAPTER 1

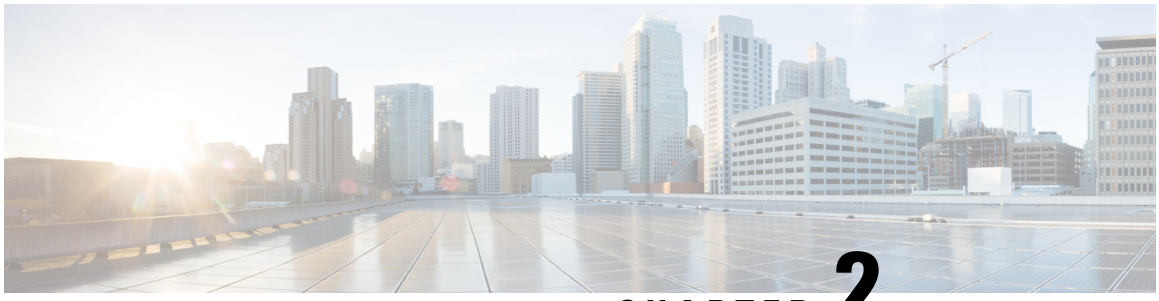
What's changed in this book

What's changed in this book

This table summarizes the new and changed features in this guide for Release 10.6(x) and provides links to the sections where they are documented.

Feature	Description	Changed in Release	Where Documented
DPU Security enablement – feature service acceleration	Allows you to power up the DPUs on the N9300 Smart Switches by enabling feature service acceleration	10.6(3s)F	Onboarding and Security Enablement
Network security operations	An exclusive admin role for the DPU security mode on N9300 Smart Switches	10.6(3s)F	User Accounts and RBAC
VRF redirection and VLAN redirection	Supports VLAN-based bridged traffic redirection and VRF-based routed traffic redirection to the DPU providing Layer 3 and Layer 4 inspection.	10.6(3s)F	Configure Traffic redirection to firewall service
Traffic inspection and integration with VXLAN	Integrates traffic inspection seamlessly with VXLAN overlays and traditional VLAN/VRF configurations, leveraging DPUs for hardware-accelerated security features within the switch.	10.6(3s)F	Traffic inspection with Smart Switches in VXLAN EVPN fabric

Feature	Description	Changed in Release	Where Documented
Inter-VRF Traffic Inspection	Inspects traffic in the DPU between sources and destinations in different VRFs; supported by symmetric hashing mode only.	10.6(3s)F	Inter-VRF Traffic Inspection
Layer 2 isolation for micro-segmentation	Allows routing of the same-subnet IPv4 traffic through security inspection using receive-only VLAN mapping, Layer 2 isolation on a service VLAN, and enhanced local Proxy ARP.	10.6(3s)F	Layer 2 Isolation for Micro-segmentation
High Availability	Ensures that the services remain available for both bridged and routed traffic during switch or DPU failure or outage.	10.6(3s)F	High Availability and Redundancy



CHAPTER 2

Smart Switches

Smart Switch

Smart Switch is a category of N9000 Series Switches that provides:

- embedded security with Hypershield
- advanced segmentation and connectivity, and
- hardware-accelerated architecture.

Embedded security with Hypershield—The N9300 Series Smart Switches provide service-accelerated performance and simplifies security architecture by integrating directly into the network, eliminating the need for separate Layer 4 stateful firewalls.

Advanced segmentation and connectivity—These switches facilitate secure segmentation and communication between security zones across the data center, interconnects, and cloud environments.

Hardware-accelerated architecture—Smart Switch integrates Data Processing Units (DPUs) with networking ASICs to enhance both data center networking throughput and security processing capabilities.

The Smart Switches that are currently available are the N9300 Series Smart Switches:

- N9324C-SE1U from 10.6(2)F
- N9348Y2C6D-SE1U from 10.6(2)F

The N9300 Series Smart Switches deliver an integrated solution for scalable, secure, and efficient data center operations, by combining advanced networking and security features with hardware acceleration and software flexibility.

The N9300 Series Smart Switches offer converged switching and routing services. These switches operate in network mode and DPU security mode. In the DPU security mode, features related to the network mode are also available.

The N9300 Smart Switches NX-OS Release 10.6(3s)F maintain functional parity with the N9000 Series NX-OS Release 10.6(3)F and additionally provides the capability to enable the DPU feature that inspects IP traffic. For more information on the software features supported in network mode, refer to Cisco Nexus 9000 Series NX-OS Release Notes of versions 10.6(2)F and 10.6(3)F and Configuration Guides for Cisco NX-OS

10.6(x). For more information about DPU security mode, refer to Cisco Nexus 9000 Series NX-OS Release Notes, Release 10.6(3s)F and this document.

Concepts

A few key concepts related to the architecture of the N9300 Smart Switch include:

- CPU—Central Processing Unit (CPU)—Controls and manages the switch; handles general control-plane tasks such as operating system processes, managing the device, routing protocols, logging, and system coordination.
- NPU—Network Processing Unit (NPU)—High-speed packet-forwarding engine; optimized for network traffic processing such as switching, routing, filtering, and traffic handling.
- DPU—Data Processing Unit (DPU)—Specialized programmable data processing engine; hosts services such as security, telemetry more efficiently.
- Modes—Two modes that the Smart Switch operates in, network and DPU security mode. The default mode is network mode. To use the DPU security mode, you need to enable feature service acceleration.
- Hypershield—A controller application designed to protect modern datacenters. Hypershield is a distributed security architecture that provides hardware-accelerated security policy enforcement using DPU technology. Hypershield centrally manages and distributes policies across all enforcement points, ensuring consistent security across the entire network.

  The N9300 Smart switch serves as a Network-Based Enforcer for implementing and managing security policies across the network. You can view the N9300 Smart switch in Hypershield under Network-based enforcers in Hypershield. For more information, refer to the Cisco Hypershield User Documentation.
- Hypershield Agent—A containerized agent that receives Security configurations and policies from Hypershield and pushes them to the DPU complex that in-turn configures security rules on DPUs. The Hypershield Agent constructs flow logs and exports them.
- Network Security Operations administrator role—A specialized role (netsecops-admin) available only on the N9300 Smart Switches that has the privileges to execute the commands within the DPU and the Hypershield agent container. For more information, refer to User Accounts and RBAC for Smart Switches.
- DPU security mode—The N9300 Smart Switch DPU security mode differs from the Security Firewall product as well as the network security features in N9000 Series Switches. The Smart Switch, when used in DPU security mode, provides a distributed, stateful Layer-4 segmentation and DPU security enforcement directly within the switch using embedded Data Processing Units (DPUs).

  To use the DPU security mode, you need to enable feature service acceleration. When the DPU is in a powered-on state, the switch works in both network and DPU security modes. When the DPU is in a powered-off state, the switch works in a network-only mode.

NX-OS and Hypershield

You can use the NX-OS command line interface or Nexus Dashboard and Hypershield to manage the operations on the N9300 Series Smart Switch.

From the NX-OS command line interface you can:

- manage the traffic redirection to the DPU

- configure network policies, and
- observe network analytics, and topology.

From the Hypershield, you can:

- manage and monitor security policies
- orchestrate the usage of security policies, and
- observe security policies and ensure security compliance.

References

The documents related to Smart Switches that provide additional references include:

- Cisco N9324C-SE1U NX-OS-Mode Switch Hardware Installation Guide
- Cisco N9348Y2C6D-SE1U NX-OS Mode Switch Hardware Installation Guide
- For Optics support on the N9300 Series Smart Switches, refer to Cisco Optics-to-Device Compatibility Matrix
- N9300 Series Smart Switch Data Sheet
- For network mode support features documentation, refer to Cisco Nexus 9000 Series Switches
- Cisco Hypershield User Documentation

Smart Switch architecture

Summary

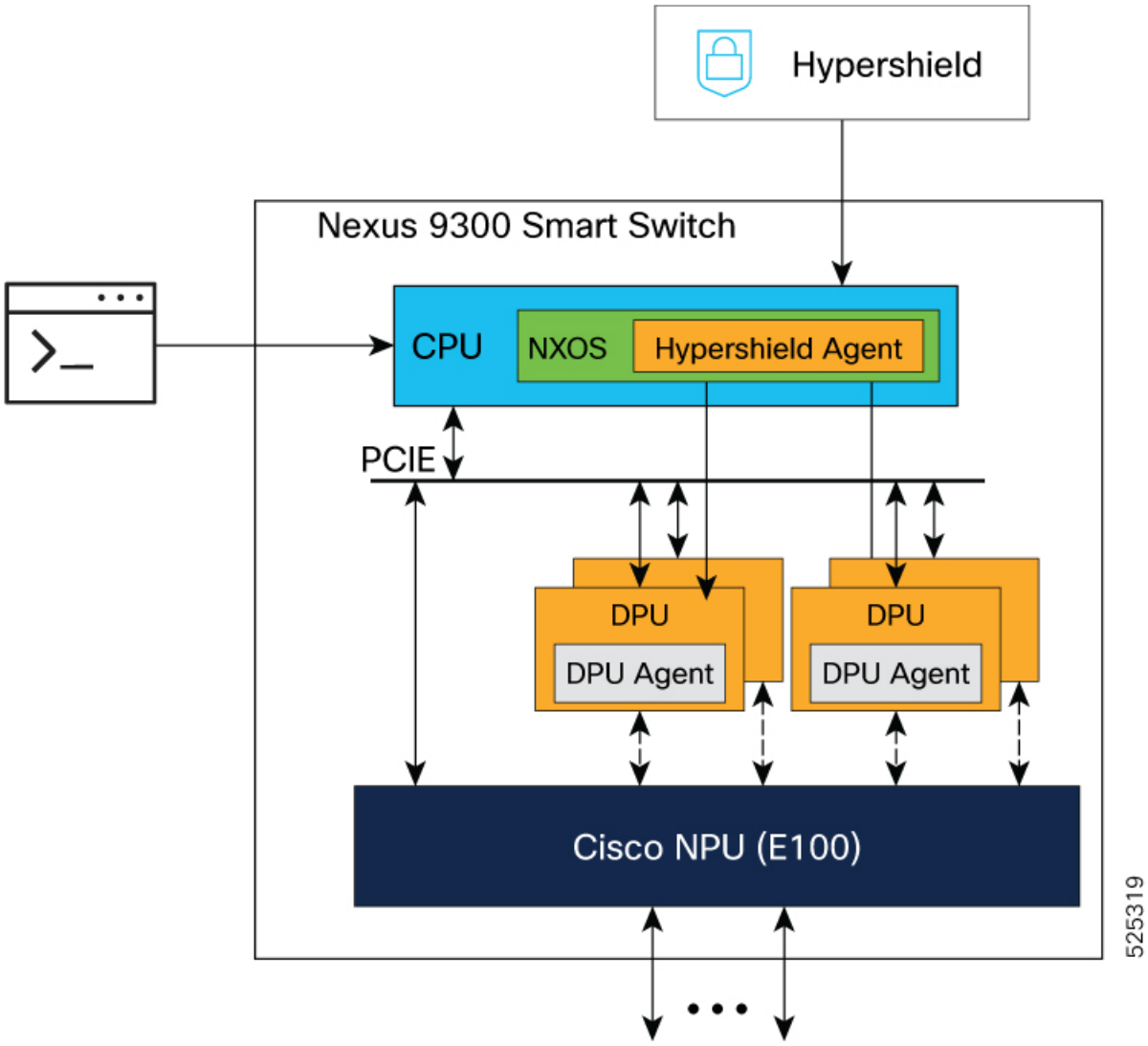
The Smart Switch architecture integrates several key components to manage and process network traffic. At the core is the Network Processing Unit (NPU) to perform routing and switching, and the Data Processing Units (DPUs) for traffic filtering. The CPU runs NX-OS and hosts the Hypershield Agent, which connects to the external Hypershield system. Management of networking configuration is performed using the NX-OS command line interface or Nexus Dashboard. The configuration of security policies is performed from Hypershield.

The Hypershield Agent in the N9300 Series Smart Switch establishes connectivity to the Hypershield system through the front panel ports and uses the IP address of a loopback interface as the source-interface.

Workflow

This flow describes how the N9300 Smart Switch works.

Figure 1: Architecture of a Smart Switch



1. When a security administrator configures a security policy in the Hypershield system, this is pushed to the Hypershield agent in the N9300 Smart switch. The Hypershield agent programs it on the DPUs.
2. The NPU performs routing and switching like any other NX-OS device. It is connected with links (for example 200G, based on the platform) to multiple DPUs (for example, the N9324C-SE1U supports 4 DPUs).
3. When you enable the required configuration on the NPU, it redirects the traffic to the DPUs, for traffic inspection. After traffic inspection, traffic is forwarded as usual by the NPU.

This architecture allows the DPU to accelerate the data plane processing for traffic filtering.

The N9300 Smart Switch works both as a network device and a security device as it includes:

- NPU, which provides the N9000 routing and switching functions that is managed using the command line interface, programmability, or Nexus Dashboard.

- DPU, which provides security functions that are managed using Hypershield.

Service-Ethernet ports

Service-Ethernet Ports are a unique type of NX-OS ethernet interface assigned to DPUs to carry traffic from the NPU to the DPU. These ports are distinct from inband or front-panel interfaces, to clearly differentiate them from other existing interface types.

The service-ethernet ports are created with default values for basic interface settings like MTU, speed, and bandwidth similar to the front-panel ports. The service-ethernet ports are always administratively UP.

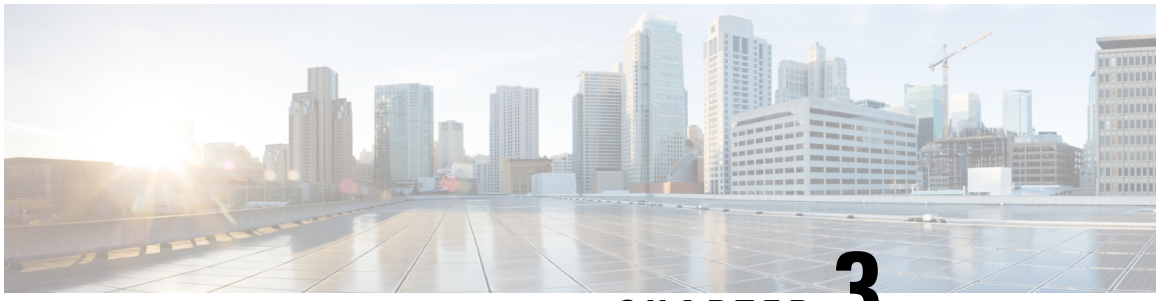
The service-ethernet ports are operational only when the DPU comes online and is detected. If the DPU goes offline, or when the service acceleration feature is unconfigured, the link may go down, and the ports corresponding to the DPU(s) are impacted.

Note These ports are configured by DPU agent every time the device is rebooted, including ISSU upgrades.

To view the status of the interfaces and DPUs they map to, use the show interface service-ethernet slot/port command.

```
switch# show interface service-ethernet 1/1
admin state is up, Connected to DPU-1
```

The service-ethernet interfaces are added to service-port-channel interfaces which can be observed in the output of the **show port-channel summary** command. The membership of the service port-channel interfaces may vary based on the platform and the load-balance configuration discussed in Configure load-balance mode.



CHAPTER 3

User Accounts and RBAC for Smart Switches

User accounts and RBAC for Smart Switches

The N9300 Smart Switches operate in network mode and DPU security mode. Both the network and security features are available in the DPU security mode. This section captures the role and user information for Smart Switches, especially for the DPU security mode which is activated by enabling the service-acceleration feature. For more information, refer to the Onboarding and Security Enablement chapter.

The switch includes several in-built or predefined user roles that control access permissions. For more information about the users and roles for the N9000 Series Switches and N9300 Smart Switches in the network mode, refer to the Configuring User Accounts and RBAC chapter in the *Cisco Nexus 9000 Series NX-OS Security Configuration Guide*.

Additional predefined role for DPU security mode

Beginning with NX-OS Release 10.6(3s)F, the N9300 Smart Switches come with an additional user role for the DPU security mode.

netsecops-admin - A specialized role available only on the N9300 Smart Switches that has the privileges to execute the commands within DPU and the Hypershield agent.

To verify the privileges of the netsecops-admin role, use the **show role name netsecops-admin** command.

Example

```
switch(config)# show role name netsecops-admin
Role: netsecops-admin
  Description: Predefined netsecops admin role for service-acceleration feature commands
  ------------------------------------------------------------------
  Rule    Perm    Type        Scope               Entity
  ------------------------------------------------------------------
  1       permit  command
switch(config)#
```

Users

The default and predefined users have roles assigned to them, and you cannot change the user roles. The users on a Smart Switch are similar to that of the N9000 Series Switches except for the netsecops-admin user. However, there is a slight change in privileges for a few users on the N9300 Smart Switches unlike rest of the N9000 Series Switches which are captured in this section.

- **admin** – The default user available on the switch with a role that provides complete read-and-write access to the entire N9000 Series Sitch. On N9300 Smart Switches, the admin user is the only user who has read-and-write access to the entire switch including network and DPU security modes.

  The admin user can perform tasks in both network and DPU security modes, while all other users are restricted to either network or DPU security mode only. As the admin user has the highest privileges for switch administration, this user can create users and assign valid roles. Thus, the admin user is the default superuser.

Note The superuser can create users with both network-admin and netsecops-admin roles. For troubleshooting service-acceleration issues, it is recommended to troubleshoot the system with users that have both roles assigned.

- **network-admin** - A user who has complete read-and-write access only to the network mode on the N9300 Smart Switches. This user can create all users including network-admins except netsecops-admins and admin and assign any role other than the netsecops-admin role.
- **netsecops-admin** - A user who has complete read-and write access only to the DPU security mode on the N9300 Smart Switches. This user's role is limited to executing commands related to DPU and Hypershield Agent. This user has no privileges to create a user or assign any roles.

  The configuration commands that a netsecops-admin user can perform are attach, username, change-password, configure, service, show, and slot. For attach and slot commands, refer to Commands executed by netsecops-admin. For the rest of the commands, refer to the Configuring User Accounts and RBAC chapter in the *Cisco Nexus 9000 Series NX-OS Security Configuration Guide*.
- **network-operator** - A user who has read-only access to the entire switch on N9000 Series Switches, but on an N9300 Smart Switch, the access is limited to the network mode on the switch. The network-admin can create this user and assign role. This role allows the user to view operational data but does not provide the rights to configure.
- **priv0-15** - These are privilege roles that can be assigned manually by the network-admin, each with specific command permissions.

Commands executed by netsecops-admin

The netsecops-admin is the only role that has access to the DPU. However, feature service acceleration must be enabled to ensure that the DPU is accessible. The netsecops-admin user can then access the DPU and execute the commands related to DPU. These commands enable advanced management and security operations on the DPUs of the switch.

Use any one of these commands on the supervisor to connect to the DPU and run the DPU commands to check and troubleshoot the DPU firmware version, NPU-DPU port status, state, and so on:

- Use the **attach dpu** *dpu-id* command to attach the user session to one DPU at a time, enabling direct SSH access to the DPU. The netsecops-admin user can then execute commands on the DPU.

  Example

  Use case: The netsecops-admin user wants to verify the firmware version using the **attach dpu** command.

```
switch# attach dpu 1
Attaching to dpu 1 ...
To exit type 'exit'...
dpu1# pdsctl show version
Firmware Version : 1.162.2
Pipeline : rudra
P4 Program : hs-dp-app
Build Time : Mon Aug 3 08:18:19 UTC 2026

dpu1# exit
Connection to dpu1 closed.
switch#
```

- Run the **slot** *slot-num* **dpu** *dpu-id internal-DPU-commands* command. This command allows executing commands directly from the switch Command Line Interface (CLI) without the **attach dpu** *dpu-id* command. It allows targeting a specific DPU in a given slot for command execution or monitoring.

  Example

  Use case: The netsecops-admin user does not want to log into the DPU but needs to verify the DPU port state-machine directly, so the user uses the slot dpu command to execute the internal DPU command and also revert to the Supervisor (SUP).

```
switch# slot 1 dpu 1 pdsctl show port fsm

Port ID: 00000111-0000-0000-4242-9ca9b8dd73e0
Timestamp State Duration (sec)
------------------------------------------------------------------
1970-01-01 00:00:10.23 ENABLED -
1970-01-01 00:00:10.124 AN_CFG 0.100659465
1970-01-01 00:00:10.124 SERDES_CFG 0.000001245
1970-01-01 00:00:10.174 WAIT_SERDES_RDY 0.050439400
1970-01-01 00:00:10.174 MAC_CFG 0.000172545
1970-01-01 00:00:10.174 WAIT_PHY_LINK_UP 0.000139070
1970-01-01 00:00:10.174 SIGNAL_DETECT 0.000002575
1970-01-01 00:00:10.174 DFE_TUNING 0.000189905
1970-01-01 00:00:10.174 DFE_START_ICAL 0.000001085
1970-01-01 00:00:10.174 DFE_WAIT_ICAL 0.000001865
1970-01-01 00:00:11.666 DFE_START_PCAL 1.491027795
1970-01-01 00:00:11.666 DFE_WAIT_PCAL 0.000007260
1970-01-01 00:00:11.666 CLEAR_MAC_REMOTE_FAULTS 0.000325270
1970-01-01 00:00:11.666 WAIT_MAC_SYNC 0.000012750
1970-01-01 00:00:11.666 WAIT_MAC_FAULTS_CLEAR 0.000015845
1970-01-01 00:00:11.666 UP 0.000007625


Port ID: 00000211-0000-0000-4242-9ca9b8dd73e0
Timestamp State Duration (sec)
------------------------------------------------------------------
1970-01-01 00:00:10.23 ENABLED -
1970-01-01 00:00:10.225 AN_CFG 0.201410325
1970-01-01 00:00:10.225 SERDES_CFG 0.000001045
1970-01-01 00:00:10.285 WAIT_SERDES_RDY 0.060090520
1970-01-01 00:00:10.285 MAC_CFG 0.000162035
1970-01-01 00:00:10.285 WAIT_PHY_LINK_UP 0.000133370
1970-01-01 00:00:10.285 SIGNAL_DETECT 0.000001940
1970-01-01 00:00:10.285 DFE_TUNING 0.000176350
```

```
1970-01-01 00:00:10.285 DFE_START_ICAL 0.000000830
1970-01-01 00:00:10.285 DFE_WAIT_ICAL 0.000001545
1970-01-01 00:00:11.718 DFE_START_PCAL 1.433245790
1970-01-01 00:00:11.719 DFE_WAIT_PCAL 0.000006890
1970-01-01 00:00:11.719 CLEAR_MAC_REMOTE_FAULTS 0.000316325
1970-01-01 00:00:11.719 WAIT_MAC_SYNC 0.000011745
1970-01-01 00:00:11.719 WAIT_MAC_FAULTS_CLEAR 0.000013710
1970-01-01 00:00:11.719 UP 0.000007615

switch#
```

- Use the **attach console dpu** *dpu-id* command to attach or connect to the console of a DPU within the switch. The console window opens and the netsecops-admin user can interact directly with the DPU's command-line interface for management or troubleshooting purposes.

Note To navigate or close the console, see the instructions on the banner at the bottom of the console. For example, to return to NX-OS, press Ctrl-A, release, then Z, X, and Enter. For more information, refer to the Cisco Hypershield User Documentation and Cisco Hypershield Reference Documentation.

Example

Use case: The netsecops-admin user wants to verify the DPU port status using the console.

```
switch# attach console dpu 1

Welcome to minicom 2.9

OPTIONS: I18n
Compiled on Sep 22 2023, 21:10:41.
Port /dev/dpu00, 00:31:44

Press CTRL-A Z for help on special keys


dpu1# pdsctl show port status
MAC-Info: MAC ID/MAC Channel/Num lanes
FEC-Type: FC - FireCode, RS - ReedSolomon
--------------------------------------------------------------------------------------------------------------------------------------
ID Name IfIndex Speed MAC-Info FEC AutoNeg MTU Pause Pause Debounce State Transceiver
NumLinkDown LinkSM Loopback
Cfg/Oper Cfg/Oper Type Tx/Rx (msecs) Admin/Oper
--------------------------------------------------------------------------------------------------------------------------------------
00000111-0000-0000-4242-9ca9b8dd73e0 Eth1/1 0x11010000 200G 0/0/4 RS/RS F/F 9440 NONE
F/F 0 UP/UP REMOVED 0 UP NONE
00000211-0000-0000-4242-9ca9b8dd73e0 Eth1/2 0x11020000 200G 0/4/4 RS/RS F/F 9440 NONE
F/F 0 UP/UP REMOVED 0 UP NONE

No. of ports : 2

dpu1#
```

- Use the **service system hypershield connect session** *[command]* to open an interactive shell session inside the Hypershield agent container running on the switch.
  - If a *command* is specified, the command is executed non-interactively inside the container and returns the output to the terminal.

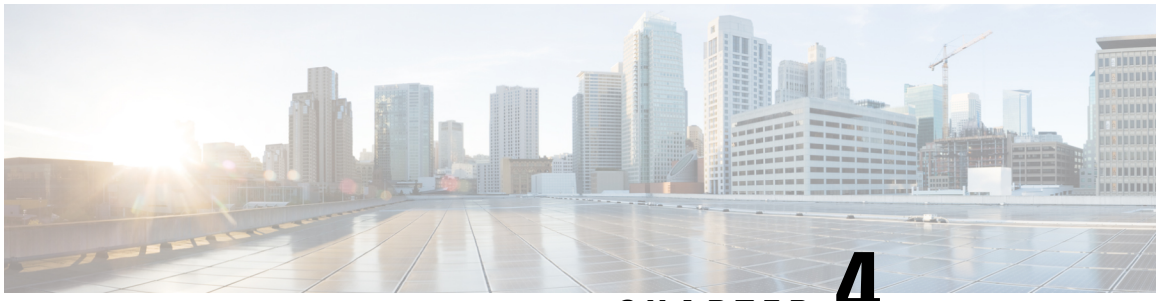
    Example- interactive session

```
switch# service system hypershield connect session
HypershieldAgent:/usr/src/app#
```

- If a *command* is not specified, it opens an interactive shell session within the Hypershield agent container.

Example – non-interactive session

```
switch# service system hypershield connect session /usr/src/app/agwctl -v
agwctl version v1.19.0-pre.8-547-gb57fe1ed
```



CHAPTER 4

Onboarding and Security Enablement

Onboarding and security enablement

Onboarding and security enablement requires the DPU to be in a powered-on state on the Smart Switches. By default, the DPU is in a powered-off state. When the DPU is in a powered-off state, the switch works in a network-only mode. When the DPU is in a powered-on state, the switch works in both network and DPU security modes.

Note Enabling the DPU security services requires registering the Smart Switch with the Hypershield Controller. Because of this you should configure the Hypershield Controller first and then onboard the Smart Switch with the Hypershield Controller.

The Smart Switch, when used in DPU security mode, provides a distributed, stateful Layer-4 segmentation and DPU security enforcement directly within the switch using embedded Data Processing Units (DPUs).

For more information about licenses, refer to the Cisco NX-OS Licensing Options Guide. For more information about Smart Licensing Using Policy (SLP) and setup, refer to the specific articles that help you learn, deploy, and troubleshoot SLP Cisco NX-OS Licensing collection page.

For troubleshooting information, refer to Cisco N9300 Series Smart Switch Troubleshooting Guide for DPU Security Mode.

Configure Hypershield Controller

Configure the Hypershield Controller before onboarding and security enablement.

The Hypershield Controller, by default, uses 10.20.0.0/16 and 10.30.0.0/16 IP subnets for setting up internal network connectivity within the Hypershield Controller Kubernetes cluster. If these subnets overlap with any routable networks in your data center environment and you want to avoid routing issues with the controller, update the `bootstrap.yaml` on the Hypershield Controller. The other option is to deploy the Hypershield Controller with a different set of subnets of /16 size instead of the default. The fields to be updated (with example overrides) in the `bootstrap.yaml` are:

```
# Override the bootstrap defaults of 10.20.0.0/16 and 10.30.0.0/16.
# 10.0.0.0/12 is not Cisco-internal routable (e.g. 10.0.0.0/16 through 10.15.0.0/16).
podCIDR: "10.8.0.0/16"
serviceCIDR: "10.9.0.0/16"
```

Workflow for onboarding and enabling security

Perform these steps to enable the DPU security mode on N9300 Smart Switches.

1. Bringup the switch by installing the software on the switch. NX-OS manages software images for both the Smart Switch and its DPUs by bundling them together.

   For more information about Software images, refer to the Cisco Nexus 9000 Series NX-OS Software Upgrade and Downgrade Guide.

2. Apply networking configurations similar to that of rest of the N9000 Series Switches.

   For more information, refer to Configuration Guides for Cisco NX-OS 10.6(x). For example, to configure VRF contexts and Layer 3 interfaces with VRF members and configure VLANs and interfaces pertaining to those VLANs, refer to Cisco Nexus 9000 Series NX-OS Interfaces Configuration Guide, and Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide.

3. Enable the service acceleration feature.

4. Configure a loopback interface for Hypershield source-interface for the Hypershield Agent to Hypershield Controller connectivity; refer to Create a loopback interface for Hypershield connectivity.

5. Add the configuration to connect to the Hypershield Controller; refer to Configure Hypershield connectivity.

6. Request and apply the token from Hypershield Controller; refer to Register the N9300 Smart switch with Hypershield.

7. Configure traffic inspection for traffic in the required VRFs and VLANs; refer to Configure Traffic redirection to the firewall service.

8. Enable the firewall service functionality; refer to Enable service firewall.

Enable Service Acceleration feature

The **feature service-acceleration** command is used to enable the DPU security mode or the DPU functionality. When enabled, NX-OS powers up the DPUs, which takes some time. As a result, NX-OS prevents the execution of no feature service-acceleration command until the DPUs are fully powered up and reach a terminal state.

The feature service-acceleration command cannot be disabled until all DPUs in the system have been powered on. Feature service-acceleration can be re-enabled after it has been disabled, without a switch reboot.

Note If you use the configure replace feature (refer to Performing Configuration Replace), the success of configure replace may depend on the timing of the configuration relative to when service-acceleration was enabled or disabled.

Perform this task to enable the service acceleration feature and power up the DPUs in the switch.

Procedure

Enable the **feature service-acceleration** command to power up the DPUs.

Example:

```
switch(config)# feature service-acceleration
```

If the **feature service-acceleration** command is *not* configured, the DPUs are powered off. The switch functions as a NXOS switch.

Note
Enabling feature service-acceleration powers up the DPUs; however, to finalize the configuration, you must define which VRFs traffic should be redirected to the DPU. See Configure traffic redirection to the firewall service, on page 22.

Verify service acceleration

Perfrom this task to verify service acceleration enablement.

Procedure

Step 1 Verify the service acceleration status using the **show run service-acceleration | grep feature** command.

Example:

```
switch# show run service-acceleration | grep feature

feature service-acceleration
```

Step 2 Use the **show interfaces brief** command to view the status of the interfaces.

Example:

```
switch# show interface brief
..!
--------------------------------------------------------------------------------
Service         VLAN    Type Mode   Status  Reason                 Speed     Port
Ethernet                                                                     Ch #
--------------------------------------------------------------------------------
SEth1/1         --      eth  routed up      none                   200G(D) --
SEth1/2         --      eth  routed up      none                   200G(D) --
SEth1/3         --      eth  routed up      none                   200G(D) --
SEth1/4         --      eth  routed up      none                   200G(D) --
```

All DPUs power up when the feature service-acceleration is enabled.

Step 3 Use the **show module** command to verify if the DPUs are powered up and online.

Example:

```
switch# show module
Mod Ports                  Module-Type                           Model           Status
--- ----- ------------------------------------------------ --------------------- --------
1    56   48x25G/2x100G/6x400G Ethernet Module             N9348Y2C6D-SE1U       ok
27   0    Virtual Supervisor Module                        N9348Y2C6D-SE1U       active *

Mod  Sw                       Hw    Slot
---  ---------------------- ------ ----
1    10.6(3r)                 0.1060 NA
27   10.6(3r)                 0.1060 VSUP

Mod  MAC-Address(es)                          Serial-Num
---  ---------------------------------------  ----------
1    fc-58-9a-1d-04-d4 to fc-58-9a-1d-05-b8  FDO29090ZW6
27   fc-58-9a-1d-04-d4 to fc-58-9a-1d-05-b8  FDO29090ZW6

Mod  Online Diag Status
---  ------------------
1    Pass
27   Pass

* this terminal session
Mod DPU   Module-Type           Model         Status
--- ---  ------------- ---------------------- --------
1   1    DPU           N9348Y2C6D-SE1U-DPU   ok
1   2    DPU           N9348Y2C6D-SE1U-DPU   ok

Mod DPU     Sw            Hw      Serial-Num   Online Diag Status
--- ---  ------------- ---------- ------------ ------------------
1   1     1.162.2      ES         NA           Pass
1   2     1.162.2      ES         NA           Pass
```

Disable service-acceleration feature

Procedure

Disable service-acceleration feature and Layer 4-7 services on the switch, for it to function in DPU powered-off state, using the using the **no feature service-acceleration** command.

Example:

```
switch(config)# no feature service-acceleration
```

Create a loopback interface for Hypershield connectivity

Perform this task to create a loopback interface to be used to connect to the Hypershield Controller. To associate this loopback interface with the Hypershield agent in the service system Hypershield configuration, see Configure Hypershield connectivity, on page 20.

To ensure that the loopback address is reachable from a front-panel interface without any routing configuration on the external management network you can configure the interface as follows:

Procedure

Step 1 Create a loopback interface for the Hypershield source-interface using the **interface loopback** *instance* command.

Example:

```
switch(config)# interface loopback 100
switch(config-if)#
```

Step 2 Configure an IP address for the interface using the **ip address** *ip-address/length* command.

Example:

```
switch(config-if)# ip address 10.16.0.140/32
```

For more information about IP addresses, see the Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide.

ip-address/length: Sets an IP address for the loopback

Note

- You have to configure /32 as length of the IP address for Hypershield Agent.
- Management traffic from Hypershield Agent to Hypershield system is *not* supported over IPv6.
- The loopback interface must be configured in the default VRF.

Step 3 Configure the physical interface and assign it the IP address of the loopback interface using the **interface** *interface-type slot/port*, **medium p2p**, and **ip unnumbered loopback** *loopback instance* commands. This means the physical interface borrows the IP address of the loopback interface, allowing the loopback IP to be reachable through this physical interface without requiring additional IP addresses.

Example:

```
switch(config)# interface eth1/3/3
switch(config-if)# medium p2p
switch(config-if)# ip unnumbered loopback 1
```

Step 4 Add static route for subnet through the physical interface with the next hop. This static route directs traffic destined for the subnet to go out through the interface with the next-hop IP, ensuring proper routing of packets to and from the loopback address through the physical interface. To configure, use the **ip route** *ip-address/subnet-mask interface-type slot/port next-hop-ip-address* command.

Example:

```
switch(config)# ip route 10.16.0.0/24 eth1/3/3 10.16.0.1
```

Step 5 Use the **show run** command to verify the loopback interface configuration.

Example:

```
switch# show run
!
interface loopback1
  ip address 10.16.0.140/32
!
interface eth1/3/3
  medium p2p
  ip unnumbered loopback1
!
ip route 10.16.0.0/24 eth1/3/3 10.16.0.1
```

Configure Hypershield connectivity

Hypershield management traffic operates within the default VRF. A loopback interface must be configured for this communication. This loopback is used as the source interface in the service system hypershield configuration, its IP address is exclusively reserved for communication between the agent and the Hypershield and it can also be used for HA. The loopback IP address cannot be reused for any control protocols running on the switch. Any attempt to test reachability to other destinations using ICMP echo from the NX-OS command line interface fails, if the loopback IP address is specified as the source.

Connectivity from the Hypershield agent to the Hypershield controller must occur over a routed front panel interfaces within the default VRF and is not supported over SVIs on vPCs.

Management traffic from the Hypershield Agent to the Hypershield Controller is not supported over IPv6.

Even if the default VRF is used for the connectivity to the Hypershield Controller, this VRF can also be used for traffic forwarding and it is possible to configure the default VRF for traffic inspection.

Use the troubleshooting commands to verify the connectivity between the Hypershield Agent and the Hypershield Controller. For more information, see Cisco N9300 Series Smart Switches DPU Security Mode Troubleshooting Guide.

Perform this task to establish connectivity between the Hypershield Agent in the N9300 Smart switch and the Hypershield Controller.

Procedure

Step 1 Enable the **service system hypershield** command to set up the Hypershield instance.

Example:

```
switch(config)# service system hypershield
```

Step 2 Configure the **source-interface** command to assign to the Hypershield Agent the IP address of the loopback interface that you previously defined.

Example:

```
switch(config-svc-sys)# source-interface loopback 100
```

The loopback interface must be configured in the default VRF. See Create a loopback interface for Hypershield connectivity, on page 19.

The example shows service acceleration feature configuration.

```
switch# show run service-acceleration
!
feature service-acceleration
service system hypershield register 34C58A…
!
service system hypershield
source-interface loopback 100
...!
service firewall
  in-service
```

Register the Smart Switch with Hypershield

Registering the Smart Switch with Hypershield requires obtaining a token from Hypershield.

This token must be entered on the switch to establish communication between the switch and Hypershield. For more information, refer to Cisco Hypershield User Documentation.

Procedure

Establish communication between the switch and Hypershield with the obtained token using the **service system hypershield register** *otp* command.

Example:

```
Switch# service system hypershield register 34C58A…
```

Execute this command at the EXEC level.

otp: Indicates the token string (maximum size 4094). Type the token without any quotes.

Verify Hypershield connection status

Procedure

Verify the status of the connectivity using the **show service-acceleration status details** command.

Example:

```
switch# show service-acceleration status details

Service System: hypershield
Source Interface: loopback101 (12.0.0.1)
```

```
DPU Load-balance Mode: symmetric-hash
Agent Status: firewall-ready,redirect-installed
Agent Health Status: ok
Controller Connection Status: success
Agent HA Status: ready
Services:
Firewall: in-service
[...]
```

Configure traffic redirection to the firewall service

You can configure VRFs or VLANs for traffic inspection through the DPUs. In this document we refer to these VRFs and VLANs as service VRFs and service VLANs respectively.

Note The firewall service does not inspect non-ip traffic, multicast IP traffic, broadcast IP traffic, traffic destined for the local switch, traffic originating from the supervisor, BFD echo packets, management VRF traffic, and traffic pertaining to any VRFs or VLANs that are not added under service firewall. All other IPv4 and IPv6 traffic pertaining to the VRFs and VLANs for service firewall are sent to the DPU firewall services for inspection. The DPU inspects IPv4 and IPv6 traffic with unicast IP addresses regardless of the MAC addresses being multicast or broadcast.

Configure load-balance mode

By default, traffic in service VRFs and VLANs is load-balanced using a symmetric hardware-based hash to all available DPUs. Routed and bridged traffic for VRFs and VLANs that are configured for filtering is distributed across the DPUs using a hardware-based symmetric hash mechanism. Users have the option to enable a pinning load-balance mode to selectively pin traffic for entire VRFs or VLANs to a specific DPU.

Note Only symmetric hashing mode is supported for Inter-VRF filtering.

The load-balance mode may only be modified when the service firewall is not in the in-service state or before the service firewall is configured.

You can configure the switch to selectively pin traffic pertaining to VRFs and VLANs to specific DPUs or allow the system to select specific DPUs for specific VRFs and VLANs.

Note This is an optional step and as of NX-OS 10.6(3s)F, the **service dpu load-balancing pinning** configuration is not for production deployments.

Procedure

Step 1 Configure the load-balance mode to pinning mode.

Example:

```
switch(config)# service dpu load-balance pinning
```

Step 2 Revert the configuration to symmetric-hash mode.

Example:

```
switch(config)# service dpu load-balance symmetric-hash
```

OR

```
switch(config)# no service dpu load-balance pinning
```

The example shows service acceleration with pinning load-balance mode configured.

```
switch# show run service-acceleration
!
feature service-acceleration
service dpu load-balance pinning
!
```

Configure Service VRFs

If a VRF is configured under service firewall, this means that this VRF is subject to traffic redirection. Unicast routed IPv4 and IPv6 traffic arriving on Layer 3 physical interfaces and port-channels, physical and port-channel subinterfaces, SVIs and VXLAN fabric interfaces is inspected if they pertain to this VRF. Traffic arriving on SVIs over Layer 2 access and trunk interfaces, port-channels, and vPCs is inspected if the SVIs pertain to this VRF.

If the VRF is subject to traffic redirection, and if the service firewall is not `in-service`, the unicast traffic routed by this VRF is dropped until the service firewall is `in-service` and until a security rule to allow that traffic is in place in the DPU. Traffic redirection doesn't apply to control protocol traffic: if the VRFs are configured for service-acceleration, and the traffic is destined to the supervisor of the N9300 Smart switch, this is not inspected. Hence it is not dropped regardless of whether service firewall is in-service or not, and regardless of whether the security rules are configured.

When the service firewall is not in `in-service` state, and the DPU is not ready to inspect traffic, the Layer 3 routing protocols ensure graceful insertion and removal (GIR) behavior. To achieve this, the service-acceleration VRFs isolate the switch from the network by altering the route advertisement behavior. Protocols supporting this functionality include Border Gateway Protocol (BGP), Open Shortest Path First (OSPF), Enhanced Interior Gateway Routing Protocol (EIGRP), and Routing Information Protocol (RIP). The protocols resume regular route advertisement behavior once the service firewall is ready for use and is in `in-service` state.

The default VRF may be configured as a VRF under service firewall for traffic inspection. Traffic from the Hypershield Agent and Controller in the default VRF are excluded from inspection. The management VRF cannot be configured for traffic inspection under service firewall.

Perform this task to specify the VRFs whose traffic must be firewalled.

Procedure

Step 1 Enable the **service firewall** command.

Example:

```
switch(config-svc-sys)# service firewall
```

Step 2 Configure the VRF under the service firewall to redirect the traffic in the VRF for inspection by the firewall service using the **vrf** *vrf-name* command.

Example:

```
switch(config-svc-sys-fw)# vrf red module-affinity dynamic
```

By default, the switch uses the hardware-based symmetric hash to distribute the traffic across the DPUs. The VRF context and other required networking configuration is entered as usual on the switch.

Step 3 (Optional) Configure a specific DPU number to indicate that traffic in the VRF must be inspected by the firewall service in that DPU using the **module-affinity** command when the load-balance mode of pinning is configured.

Note
Explicitly configuring module-affinity in the 10.6(3s)F release is only for testing, not for production deployments.

Example:

```
switch(config-svc-sys-fw)# vrf blue module-affinity 1
```

Alternatively in the pinning load-balance mode, if module-affinity dynamic is configured, the system selects a DPU to inspect all traffic in the VRF.

When the system is not using the pinning load-balance mode, module-affinity can only be configured as dynamic.

This is a sample configuration example of a VRF for traffic redirection to the DPU.

```
switch(config-svc-sys)# service firewall
switch(config-svc-sys-fw)# vrf red module-affinity dynamic
switch(config-svc-sys-fw)# vrf blue module-affinity 1
```

Traffic in the VRF blue is inspected by DPU1.

This example shows service acceleration with firewalling enabled, with pinning load-balance mode

```
switch# show run service-acceleration
!
feature service-acceleration
service dpu load-balance mode pinning
!
service system hypershield
source-interface loopback 100
service firewall
   vrf blue module-affinity 1
   vrf red module-affinity dynamic
```

The example shows service acceleration with firewalling enabled, with symmetric-hash load-balance mode.

```
switch# show run service-acceleration
!
feature service-acceleration
service system hypershield register 34C58A…
!
service system hypershield
source-interface loopback 100
service firewall
vrf blue module-affinity dynamic
vrf red module-affinity dynamic
```

Configure Service VLANs

Bridged traffic within VLANs (enabled on Layer 2 access and trunk interfaces, port-channels, and vPCs) configured for service firewall undergo inspection.

VLANs configured for service firewall might have SVIs that do not belong to service firewall VRFs or might not have SVIs configured on the switch. In such scenarios, only bridged traffic within those VLANs are inspected while routed traffic is not.

VLANs might not be configured for service firewall but could have SVIs associated with service firewall VRFs. In such cases, only routed traffic within those VLANs are inspected and bridged traffic is not.

You can enable both routed and bridged traffic for a VLAN for inspection. To do so, configure the VLAN for the service firewall and configure the VRF related to its SVI for the service firewall. Note that the users or systems can choose different DPUs for inspecting routed and bridged traffic within a VLAN.

Just like for service VRFs, if a VLAN is configured under the service firewall configuration (indicating that the VLAN is subject to traffic filtering), and if the service firewall is not `in-service`, the IPv4 and IPv6 traffic bridged for this VLAN is dropped until the service firewall is `in-service` and a security rule allowing that traffic is established in the DPU.

Perform this task to specify the VLANs whose bridged traffic must be firewalled.

Procedure

Step 1 Enable the **service firewall** command.

Example:

```
switch(config-svc-sys)# service firewall
```

Step 2 Configure the VLAN under the service firewall to redirect the bridged traffic in the VLAN for inspection by the firewall service using the **vlan id** *{ vlan-id | vlan-range }* command.

Example:

```
switch(config-svc-sys-fw)# vlan id 100 bridged-traffic module-affinity dynamic
```

Example:

```
switch(config-svc-sys-fw)# vlan id 105-110 bridged-traffic module-affinity dynamic
```

By default, the switch uses the hardware-based symmetric hash to distribute the traffic across the DPUs.

The VLAN definition and other required networking configuration is entered as usual on the switch.

Step 3 (Optional) If the pinning load-balance mode has been configured, you may configure a specific DPU number to indicate that traffic in the VLAN must be inspected by the firewall service in that DPU using the **module-affinity** command.

Note
Explicitly configuring module-affinity in the 10.6(3s)F release is only for testing, not for production deployments.

Example:

```
switch(config-svc-sys-fw)# vlan id 200 bridged-traffic module-affinity 2
```

Alternatively, in the pinning load-balance mode, if module-affinity dynamic is configured, the system selects a DPU to inspect all traffic in the VLAN.

When the system is not using the pinning load-balance mode, module-affinity can only be configured as dynamic.

This is a sample configuration example of a VLAN for traffic redirection to the DPU.

```
switch(config-svc-sys)# service firewall
switch(config-svc-sys-fw)# vlan id 100 bridged-traffoc module-affinity dynamic
switch(config-svc-sys-fw)# vlan id 200 bridged-traffic module-affinity 2
```

Bridged traffic in VLAN 200 is inspected by DPU2.

This example shows service acceleration with firewalling enabled, with pinning load-balance mode.

```
switch# show run service-acceleration
!
feature service-acceleration
service dpu load-balance mode pinning
!
service system hypershield
source-interface loopback1
 service firewall
   vrf blue module-affinity 1
   vrf red module-affinity dynamic
   vlan id 100 bridged-traffoc module-affinity dynamic
   vlan id 200 bridged-traffic module-affinity 2
```

This example shows service acceleration with firewalling enabled, with symmetric-hash load-balance mode.

```
switch# show run service-acceleration

feature service-acceleration
service dpu load-balance mode pinning

service system hypershield
  source-interface loopback 100
  service firewall
    vrf blue module-affinity 1
    vrf red module-affinity dynamic
    vlan id 100 bridged-traffoc module-affinity dynamic
    vlan id 200 bridged-traffic module-affinity dynamic
    vlan id 100,200 bridged-traffic module-affinity dynamic
```

Enable service firewall

The **in-service** command enables the service firewall functionality, and allows traffic inspection by the DPUs.

You can also use the **no in-service** command to trigger maintenance mode for the firewall functionality and, to disable traffic redirection to DPUs for any reason or modify the load-balance mode.

Procedure

Use the **in-service** command to enable the service firewall to redirect specific traffic.

Example:

```
switch(config-svc-sys-fw)# in-service
```

Verify the service firewall and redirection status

You can verify the status of the service firewall and redirection when the status is out-of-service or in-service. This section provides examples for both the statuses.

- Use the **show service-acceleration status details** command to view the firewall service when is in out-of-service state.

```
switch# show service-acceleration status details
Service System: hypershield
   Source Interface: loopback48 (10.1.1.1)
   DPU Load-balance Mode: symmetric-hash
   Agent Status: firewall-disable
   Agent Health Status: ok
   Controller Connection Status: init
   Agent HA Status: not-initialized
   Services:
     Firewall: out-of-service

VRF             Operational State       Affinity(DPU)
=====================================================================
blue            isolated                n/a
red             isolated                n/a

VLAN            Operational State       Affinity(DPU)
=====================================================================
100             VLAN set to drop        n/a
200             VLAN set to drop        n/a
```

- Use the **show service-acceleration status details** command to view the status of the firewall service when it is in `in-service` state.

  This example shows the firewall service in the `in-service` state and in pinning mode.

```
switch# show service-acceleration status details
  Service System: hypershield
    Source Interface: loopback48 (10.1.1.1)
    DPU Load-balance Mode: pinning
    Agent Status: firewall-ready,redirect-installed
    Agent Health Status: ok
```

```
    Controller Connection Status: success
    Agent HA Status: not-initialized Services:
      Firewall: in-service

      VRF                               Operational State       Affinity(DPU)
      ================================================================================
      blue                              forwarding ready        1
      red                               forwarding ready        2

      VLAN                              Operational State       Affinity(DPU)
      ================================================================================
      100                               forwarding ready        1
      200                               forwarding ready        2
```

This example shows the firewall service in the in-service state and in symmetric-hash mode.

```
show service-acceleration status details
  Service System: hypershield
    Source Interface: loopback48 (10.1.1.1)
    DPU Load-balance Mode: symmetric-hash
    Agent Status: firewall-ready,redirect-installed
    Agent Health Status: ok
    Controller Connection Status: success
    Agent HA Status: not-initialized
    Services:
      Firewall: in-service

      VRF                               Operational State       Affinity(DPU)
      ================================================================================
      blue                              forwarding ready        1-4
      red                               forwarding ready        1-4

      VLAN                              Operational State       Affinity(DPU)
      ================================================================================
      100                               forwarding ready        1-4
      200                               forwarding ready        1-4
```

- Use the command **show service-acceleration redirect-policy brief** to identify the service port-channel subinterfaces for redirecting traffic from the VRFs or VLANs to the DPUs.

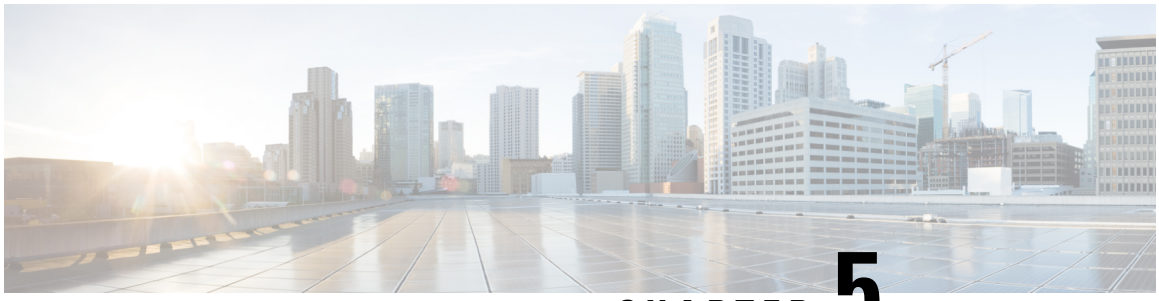
  This example shows the output when symmetric-hash load-balance mode is in use.

```
switch# show service-acceleration redirect-policy brief

VRF                            AF Type Interface[Status]    Affinity   Redirect Status

=============================================================================================
red                              IPv4    Spo512.10[UP]        1-4        Enabled

blue                             IPv4    Spo512.11[UP]        1-4        Enabled

red                              IPv6    Spo512.10[UP]        1-4        Enabled

blue                             IPv6    Spo512.11[UP]        1-4        Enabled


VLAN                           AF Type Interface[Status]    Affinity   Redirect Status

=============================================================================================
100                              IPv4    Spo512[UP]           1-4        Enabled

200                              IPv4    Spo512[UP]           1-4        Enabled

100                              IPv4    Spo512[UP]           1-4        Enabled
```

```
200                              IPv4    Spo512[UP]           1-4        Enabled
```

This example shows the output when pinning load-balance mode is in use.

```
switch# show service-acceleration redirect-policy brief

VRF                            AF Type Interface[Status]    Affinity   Redirect Status

=============================================================================================
red                              IPv4    Spo2.10[UP]        1-4        Enabled

blue                             IPv4    Spo1.11[UP]        1-4        Enabled

red                              IPv6    Spo2.10[UP]        1-4        Enabled

blue                             IPv6    Spo1.11[UP]        1-4        Enabled


VLAN                           AF Type Interface[Status]    Affinity   Redirect Status

=============================================================================================
100                              IPv4    Spo1[UP]           1-4        Enabled

200                              IPv4    Spo2[UP]           1-4        Enabled
100                              IPv4    Spo1[UP]           1-4        Enabled

200                              IPv4    Spo2[UP]           1-4        Enabled
```



CHAPTER 5

Inter-VRF Traffic Inspection in Smart Switches

Inter-VRF traffic inspection

Inter-VRF routing involves routing traffic between different VRFs by leaking routes from the source VRF to the target VRF through routing protocols. Traffic entering the target VRF is sent to the Data Processing Unit (DPU) for firewall inspection applicable to both the target and source VRFs, and then it egresses out of the source VRF. This process ensures security inspection while enabling cross-VRF communication.

Asymmetric configuration of VRFs are not supported. For example, if source VRF is a service VRF and destination VRF is a non-service VRF or if source VRF is a non-service VRF and destination VRF is a service VRF.

A route can only have a single VRF redirect destination. ECMP (Equal-Cost Multi-Path) of VRF redirect destinations is not supported.

A route cannot have a next hop in the local VRF and a VRF redirect destination simultaneously. The URIB or Client filters out and sends only the best path.

When the route lookup result is the egress VRF, for example, VRF B, this involves a second route lookup in the VRF, for example, VRF B, in the NPU, which leads to reduced bandwidth.

Only Global Consistency Checker (CC) and troubleshoot commands are supported. Single route CC is not supported for leaked routes. For more information, refer to Cisco Smart Switches Troubleshooting Guide for DPU Security Mode.

Configure Inter-VRF traffic inspection

Summary

Perform these procedures to configure inter-VRF traffic inspection:

Workflow

1. Assign source and target VRFs to service system hypershield. For more information, refer to Configure service VRFs.
2. Configure IP prefix lists and route maps for IPv4 and IPv6 filtering

3. Configure BGP Route Redistribution in VRF
4. Configure VRF Route Import and Export with Route-Targets and Route-Map

   For more information about configuring BGP route leaking, refer to Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide, Release 10.6(x).

Configure IP prefix lists and route maps for IPv4 and IPv6 filtering

Use this procedure to configure prefix lists to filter IPv4 and IPv6 prefixes and associate them with route maps to control route advertisement or filtering policies on the device. Configuring prefix list and route-maps helps in filtering the IP addresses that need to be leaked.

Procedure

Step 1 Use the **configure terminal** command to enter the global configuration mode.

Example:

```
switch# configure terminal
```

Step 2 Perform step a or b to create an IPv4 or IPv6 prefix list to permit all prefixes with prefix length up to 32 for IPv4 and up to 128 for IPv6 respectively. The prefix-list name can be any alphanumeric string up to 63 characters.

a) Use the **ip prefix-list** *prefix-list-name* **seq** *sequence-number* **permit** *ipv4-prefix* **le** *max-prefix-length* command for IPv4.

Example:

```
switch(config)# ip prefix-list test seq 5 permit 0.0.0.0/0 le 32
```

b) Use the **ipv6 prefix-list** *prefix-list-name* **seq** *sequence-number* **permit** *ipv6-prefix* **le** *max-prefix-length* command for IPv6.

Example:

```
switch(config)# ipv6 prefix-list testv6 seq 5 permit 0::/0 le 128
```

Step 3 Use the **route-map** *route-map name* **permit** *sequence number* command to define an IPv4 or IPv6 route-map with a permit statement sequence.

Example:

```
switch(config)# route-map test permit 10
```

Example:

```
switch(config)# route-map testv6 permit 10
```

Step 4 Use the **match ip address prefix-list** *prefix-list-name* command to match IP route against the prefix list within the route map for IPv4 or IPv6.

Example:

```
switch(config)# match ip address prefix-list test
```

Example:

```
switch(config)# match ipv6 address prefix-list testv6
```

Configure BGP route redistribution in VRF

This configuration enables BGP, enters the BGP routing process for autonomous system number (AS) 100, and configures redistribution of multiple routing protocols and sources into BGP within the VRF named red. The route-map statements allow filtering or policy control of redistributed routes. The examples cover both IPv4 and IPv6 address families under the VRF context.

Procedure

Step 1 Use the **configure terminal** command to enter the global configuration mode.

Example:

```
switch# configure terminal
```

Step 2 Enable **feature BGP**.

Example:

```
switch(config)# feature BGP
```

Step 3 Enter BGP router configuration mode with AS number 100 using the **router bgp 100** command.

Example:

```
switch(config)# router bgp 100
```

Step 4 Configure IPv4 or IPv6 unicast address family for VRF **red**. Perform step a or b for IPv4 or IPv6 respectively.

a) Use the **address-family ipv4 unicast vrf** *vrf-name* command for IPv4.

Example:

```
switch(config)# address-family ipv4 unicast vrf red
```

b) Use the **address-family ipv6 unicast vrf** *vrf-name* command for IPv6.

Example:

```
switch(config)# address-family ipv6 unicast vrf red
```

Step 5 Redistribute routes in VRF red from various clients within BGP. Perform step a or b for IPv4 or IPv6 respectively.

a) For IPv4, redistribute Adjacency Manager (am), OSPF process 100, directly connected, or static routes with route-map.

Example:

```
switch(config)# redistribute am route-map test
switch(config)# redistribute ospf 100 route-map test
switch(config)# redistribute direct route-map test
switch(config)# redistribute static route-map test
```

b) For IPv6, redistribute OSPFv3 process 100, directly connected, and static routes with route map.

Example:

```
switch(config)# redistribute ospfv3 100 route-map testv6
switch(config)# redistribute direct route-map testv6
switch(config)# redistribute static route-map testv6
```

Step 6 Exit the current address family configuration mode.

Example:

```
switch(config)# exit-address-family
```

Configure VRF route import and export with route-targets and route-maps

Use this procedure to configure route import and export between two VRFs, for example, green and red, using route-targets and route maps to control route leaking.

A brief explanation on how route leaking works in this example:

- Routes exported from VRF green are tagged with 3:3. VRF red imports routes with route-target 3:3, so it receives routes exported by green.
- Conversely, routes exported from VRF red are tagged with 2:2. VRF green imports routes with route-target 2:2, so it receives routes exported by red.
- The use of export map and import map (test and testv6) allows filtering or modifying routes during export/import.
- The additional import vrf default map test in VRF red allows it to import routes from the default VRF, applying the test route map.
- This configuration enables bidirectional route leaking between VRF green and VRF red, controlled by route-targets and route maps.

Procedure

Step 1 Configure VRF green.

a) Use the **configure terminal** command to enter the global configuration mode.

Example:

```
switch# configure terminal
```

b) Enable **feature BGP**.

Example:

```
switch(config)# feature BGP
```

c) Configure VRF green using the **vrf context** *vrf-name* command.

Example:

```
switch(config)# vrf context green
```

d) Enter the IPv4 or IPv6 unicast address family configuration mode within the VRF green.

i. Use the **address-family ipv4 unicast** command for IPv4.

Example:

```
switch(config)# address-family ipv4 unicast
```

ii. Use the **address-family ipv6 unicast** command for IPv6.

Example:

```
switch(config)# address-family ipv6 unicast
```

e) Configure the VRF to import routes tagged with the route-target extended community 2:2 for IPv4 or IPv6 using the **route-target import 2:2** command.

Example:

```
switch(config)# route-target import 2:2
```

f) Configure the VRF to export routes tagged with the route-target extended community 3:3 for IPv4 or IPv6 using the **route-target export 3:3** command.

Example:

```
switch(config)# route-target export 3:3
```

g) Apply the route-map for IPv4 or IPv6 to filter or control routes being exported from the VRF green. Use the **export map** *route-map* command for IPv4 or IPv6 respectively.

Example:

```
switch(config)# export map test
```

Example:

```
switch(config)# export map test6
```

h) Apply the route-map for IPv4 or IPv6 to filter or control routes being imported into the VRF green. Use the **import map** *route-map* command for IPv4 or IPv6 respectively.

Example:

```
switch(config)# import map test
```

Example:

```
switch(config)# import map test6
```

i) Exit the current address family configuration mode.

Example:

```
switch# exit-address-family
```

Step 2 Configure VRF red.

a) Configure **feature BGP** as mentioned in Step a and b of Step 1, and then configure VRF red.

Example:

```
switch# configure terminal
switch(config)# feature BGP
switch(config)# vrf context red
```

b) Enter the IPv4 or IPv6 unicast address family configuration mode within the VRF red.

i. Use the **address-family ipv4 unicast** command for IPv4.

Example:

```
switch(config)# address-family ipv4 unicast
```

ii. Use the **address-family ipv6 unicast** command for IPv6.

Example:

```
switch(config)# address-family ipv6 unicast
```

c) Configure the VRF to import routes tagged with the route-target extended community 3:3 for IPv4 or IPv6 using the **route-target import 3:3** command.

Example:

```
switch(config)# route-target import 3:3
```

d) Configure the VRF to export routes tagged with the route-target extended community 2:2 for IPv4 or IPv6 using the **route-target export 2:2** command.

Example:

```
switch(config)# route-target export 2:2
```

e) Apply the route-map for IPv4 or IPv6 to filter or control routes being exported from the VRF red. Use the **export map** *route-map* command for IPv4 or IPv6 respectively.

Example:

```
switch(config)# export map test
```

Example:

```
switch(config)# export map test6
```

f) Use the **import vrf default map** *route-map* command to import routes from the default VRF.

Example:

```
switch(config)# import vrf default map test
```

g) Apply the route-map for IPv4 or IPv6 to filter or control routes being imported into the VRF red. Use the **import map** *route-map* command for IPv4 or IPv6 respectively.

Example:

```
switch(config)# import map test
```

Example:

```
switch(config)# import map test6
```

h) Exit the current address family configuration mode.

Example:

```
switch# exit-address-family
```

Verify inter-VRF traffic inspection

This section includes examples for inter-VRF traffic inspection.

In this example, route is learned on the source VRF red.

```
show ip route 12.10.1.0/24 vrf red
IP Route Table for VRF "red"
'*' denotes best ucast next-hop
'**' denotes best mcast next-hop
'[x/y]' denotes [preference/metric]
'%<string>' in via output denotes VRF <string>

12.10.1.0/24, ubest/mbest: 1/0
*via 1.10.1.2, [1/0], 00:02:58, static
```

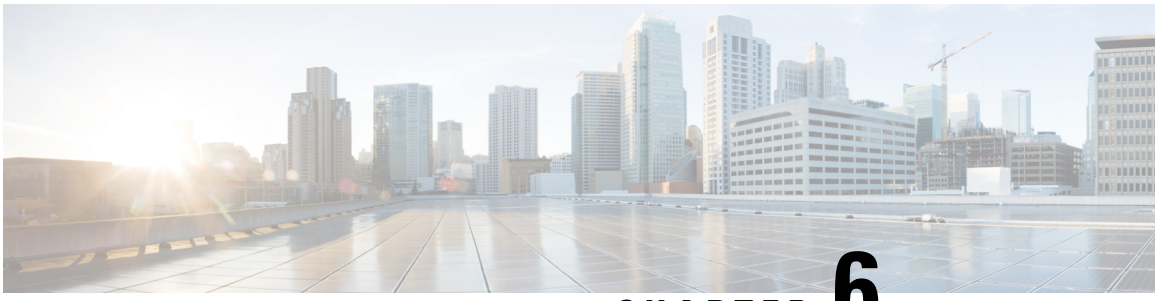
In this example, route is leaked to the target VRF green.

```
show ip route 12.10.1.0/24 vrf green
IP Route Table for VRF "green"
'*' denotes best ucast next-hop
'**' denotes best mcast next-hop
```

```
'[x/y]' denotes [preference/metric]
'%<string>' in via output denotes VRF <string>

12.10.1.0/24, ubest/mbest: 1/0
*via 1.10.1.2%red, [20/0], 00:13:58, bgp-100, external, tag 100
```

In the output of the **show vrf red detail** command, the table id or hardware VRF id for VRF red is 6 and the table id for VRF green is 4.

```
show vrf red detail
VRF-Name: red, VRF-ID: 7, State: Up
VPNID: unknown
RD: 0:0
Max Routes: 0 Mid-Threshold: 0
Table-ID: 0x80000006, AF: IPv6, Fwd-ID: 0x80000006, State: Up
Table-ID: 0x00000006, AF: IPv4, Fwd-ID: 0x00000006, State: Up
VRF Type: Service

show vrf green detail
VRF-Name: green, VRF-ID: 5, State: Up
VPNID: unknown
RD: 0:0
Max Routes: 0 Mid-Threshold: 0
Table-ID: 0x80000004, AF: IPv6, Fwd-ID: 0x80000004, State: Up
Table-ID: 0x00000004, AF: IPv4, Fwd-ID: 0x00000004, State: Up
VRF Type: Service
```



CHAPTER 6

Traffic Inspection with Smart Switches in VXLAN EVPN Fabric

Traffic Inspection with Smart Switches in VXLAN EVPN fabric

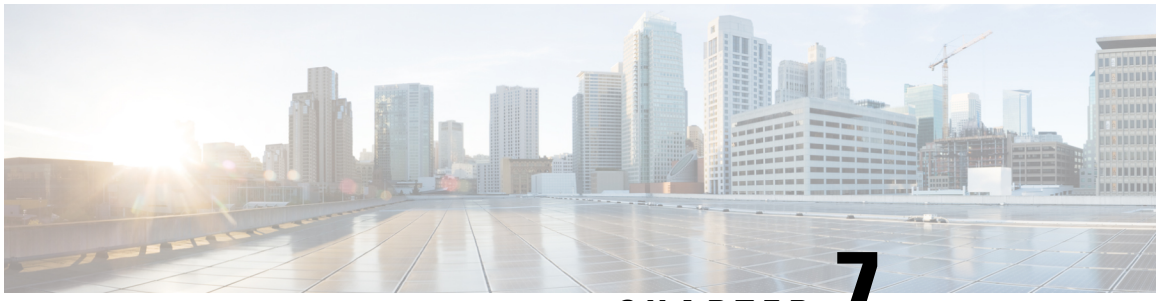
The N9300 Smart Switch can be used as a leaf switch in VXLAN EVPN fabrics to inspect unicast IPv4 and IPv6 traffic (in service VRFs or VLANs) arriving from hosts, remote leaf switches or border gateways. Smart Switches may also be used as border leaf switches or Multi-Site Border Gateways to inspect traffic arriving from remote sites.

When Smart Switches are used as VXLAN leaf switches, traffic for service VRFs and VLANs, arriving from the hosts, is inspected at the encapsulating smart-switch VTEP. After traffic inspection, the traffic is encapsulated and forwarded through the fabric to the destination VTEP. The destination smart-switch VTEP, decapsulates the traffic and then inspects the traffic for the service VRFs and VLANs. After traffic inspection, traffic is forwarded as usual to the destination hosts. Transit traffic arriving on fabric-facing interfaces is not subjected to traffic redirection.

In VXLAN EVPN topologies, the default VRF serves as the underlay VRF. This VRF is reserved for the transport of VXLAN-encapsulated traffic; it is not designed to support tenant endpoint connectivity. Consequently, we recommend not configuring the default VRF as a service VRF for Smart Switches. However, if you configure the default VRF as a service VRF, these limitations apply:

- Network connectivity for tenant VRFs and VLANs that do not require traffic inspection can be disrupted due to VRF isolation mechanisms.
- vPC fabric peering does not function correctly in this configuration.
- Inter-VRF traffic filtering between hosts in the default VRF and the non-default VRF does not work.

For more information about VXLAN EVPN fabric support, refer to the *Configure VXLAN BGP EVPN* chapter in the Cisco Nexus 9000 Series NX-OS VXLAN Configuration Guide.



CHAPTER 7

Layer 2 isolation for micro-segmentation

This chapter describes how local-proxy ARP and Layer 2 isolation support micro-segmentation and how to configure and verify the feature.

Local-proxy ARP and Layer 2 isolation

A local-proxy ARP with Layer 2 isolation is a network security mechanism that

- directs IPv4 traffic between same-subnet endpoints through the smart switch for inspection and policy enforcement
- prevents endpoints from bypassing the smart switch by blocking direct Layer 2 communication, and
- prevents unnecessary traffic forwarding to the switch by responding with the gateway MAC address only when the destination is reachable.

Components of local-proxy ARP with Layer 2 isolation

Local-proxy ARP with Layer 2 isolation uses these components to move same-subnet IPv4 traffic through the smart switches:

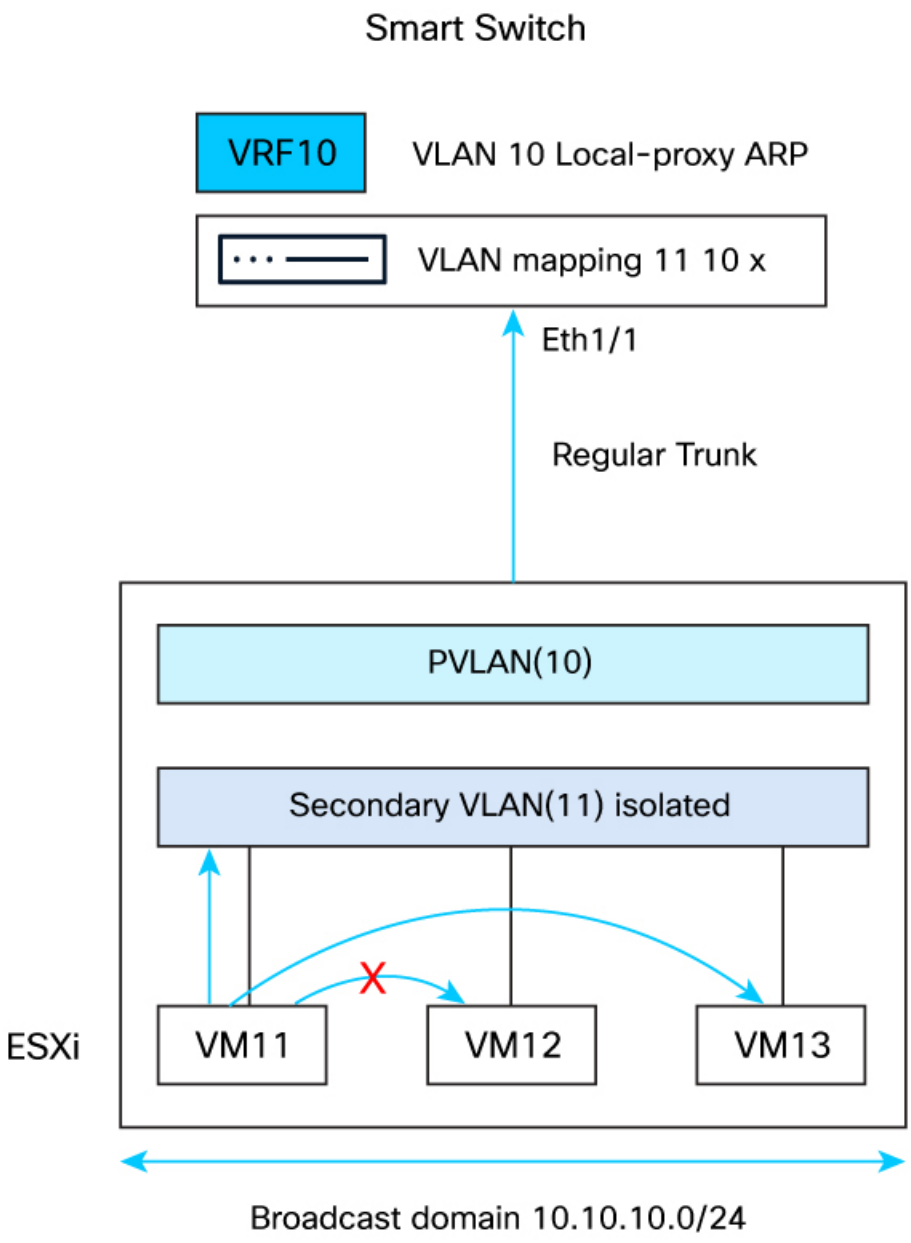
- **Private VLAN isolation**: Prevents direct Layer 2 communication between endpoints on an intermediate Layer 2 switch or distributed virtual switch.
- **Receive-only VLAN mapping**: Maps the secondary VLAN to the primary VLAN on ingress without changing the VLAN on egress.
- **Enhanced local-proxy ARP**: Returns the smart switch gateway MAC address when the destination is known and reachable.
- **Service VLAN Layer 2 isolation**: Blocks direct Layer 2 traffic that may use learned MAC or ARP information, to prevent bypassing the DPU stateful security inspection.

Topology for local-proxy ARP and Layer 2 isolation

The topology illustrates how local-proxy ARP and Layer 2 isolation secure communication between virtual machines in the same IPv4 subnet:

- The ESXi private VLAN (PVLAN) configuration prevents direct Layer 2 communication between virtual machines in isolated secondary VLAN 11.
- A regular trunk between the ESXi host and Ethernet1/1 on the smart switch carries primary VLAN 10 and secondary VLAN 11.
- Ethernet1/1 maps traffic received in secondary VLAN 11 to primary service VLAN 10. The receive-only mapping does not translate VLAN 10 to VLAN 11.
- The VLAN 10 switched virtual interface (SVI) provides the default gateway for the 10.10.10.0/24 subnet and uses enhanced local-proxy ARP.
- VRF 10 redirects routed traffic to the service firewall, and the service VLAN isolation policy prevents bridged traffic from bypassing inspection.
- The smart switch routes eligible same-subnet IPv4 traffic through the service firewall while maintaining Layer 2 isolation.

Figure 2: Local-proxy ARP topology with Layer 2 isolation



Key benefits

- Maintains Layer 2 isolation for security.
- Supports local-proxy ARP for intra-subnet communication.
- Enforces DPU stateful filtering.

Guidelines and limitations for local-proxy ARP and Layer 2 isolation

Review these guidelines and limitations before configuring same-subnet IPv4 traffic inspection through a service firewall.

Traffic support

- Beginning with NX-OS Release 10.6(3s)F, local-proxy ARP with Layer 2 isolation supports IPv4 bridged traffic only. Neighbor Discovery proxy is not supported.
- Layer 2 isolation drops bridged IPv4, IPv6, and non-IP traffic. IPv6 endpoint communication is not supported because Neighbor Discovery proxy is not supported.
- Local-proxy ARP with Layer 2 isolation supports smart switches in a virtual port channel (vPC) topology.

Configuration requirements

- Configure an SVI for each isolated service VLAN. Associate the SVI with a virtual routing and forwarding (VRF) instance enabled for routed-traffic inspection.
- Configure enhanced local-proxy ARP with the `reachable` keyword. The smart switch responds only after it confirms destination reachability.
- A service VLAN can use either the default redirect action or the isolate action. An isolated VLAN does not support a nondefault module affinity.
- The system does not validate SVI, VRF, or enhanced local-proxy ARP dependencies for the isolation action. Configure and verify each dependency separately.

VLAN mapping behavior

- To change an existing mapping between bidirectional and receive-only operation, remove the mapping and create it again with the required direction.
- Receive-only VLAN mapping supports single-tagged packets when the inner VLAN ID is 0.
- Spanning Tree Protocol (STP) is not supported on interfaces configured with **switchport vlan mapping** *x y* **rx**. STP requires symmetric bidirectional VLAN mapping.

Software downgrade considerations

- Before downgrading to a release earlier than NX-OS Release 10.6(3s)F, remove mappings that use the **rx** keyword.

Configure local-proxy ARP with Layer 2 isolation

Use this task when the subnet gateway is an SVI on smart switches. This configuration maps isolated secondary VLANs to the primary VLAN, forcing same-subnet traffic through Layer 3 firewall inspection and enabling enhanced security.

Follow these steps to configure local-proxy ARP with Layer 2 isolation:

Before you begin

- Configure a private VLAN on the intermediate Layer 2 switch or distributed virtual switch to prevent direct endpoint communication.
- Allow the primary VLAN on a promiscuous trunk. Otherwise, allow both VLANs on a regular trunk and use receive-only VLAN mapping on the smart switch.
- Create the translated VLAN on the smart switch and allow it on the Layer 2 trunk.
- Configure VRF `red` for routed-traffic inspection.

Procedure

Step 1 Enter global configuration mode.

Example:

```
switch# configure terminal
```

Step 2 Select the Layer 2 trunk interface that connects to the intermediate switch or distributed virtual switch.

Example:

```
switch(config)# interface Ethernet1/1
```

Step 3 Enable VLAN mapping and map the secondary VLAN to the primary VLAN on ingress.

Example:

```
switch(config-if)# switchport mode trunk
switch(config-if)# switchport vlan mapping enable
switch(config-if)# switchport vlan mapping 11 10 rx
switch(config-if)# switchport trunk allowed vlan 10
switch(config-if)# no shutdown
```

For more information, refer to Configure rx-only VLAN mapping.

Step 4 Configure the SVI for the primary VLAN and enable enhanced local-proxy ARP.

Example:

```
switch(config)# interface Vlan10
switch(config-if)# vrf member red
switch(config-if)# ip address 10.10.10.254/24
switch(config-if)# ip local-proxy-arp reachable
switch(config-if)# no shutdown
```

For more information, refer to Configure proxy.

Step 5 Configure the primary VLAN as an isolated service VLAN under the service firewall.

Example:

```
switch(config)# service system hypershield
switch(config-svc-sys)# source-interface loopback2
switch(config-svc-sys)# service firewall
switch(config-svc-pol-fw)# vlan id 10 bridged-traffic isolate
```

For more information, refer to Configure VLAN layer 2 isolation.

Step 6 Configure the SVI VRF for traffic redirection to the firewall service.

Example:

```
switch(config-svc-pol-fw)# vrf red module-affinity dynamic
```

For more information, refer to Configure Service VRFs, on page 23.

Step 7 Enable the service firewall.

Example:

```
switch(config-svc-pol-fw)# in-service
```

For more information, refer to Enable service firewall, on page 27.

The smart switch routes eligible same-subnet IPv4 traffic through the service firewall and drops bridged traffic on isolated service VLANs.

Configure local-proxy ARP on an EVPN SVI

Use this task to enable local-proxy ARP on an Ethernet VPN (EVPN) switched virtual interface (SVI) configured with the anycast gateway. Local-proxy ARP allows the gateway to direct same-subnet traffic efficiently.

Follow these steps to enable local-proxy ARP on an EVPN SVI:

Procedure

Step 1 Enter global configuration mode.

Example:

```
switch# configure terminal
```

Step 2 Select the SVI.

Example:

```
switch(config)# interface Vlan10
```

Step 3 Enable the anycast gateway and local-proxy ARP on the SVI.

Example:

```
switch(config-if)# fabric forwarding mode anycast-gateway proxy
```

Note

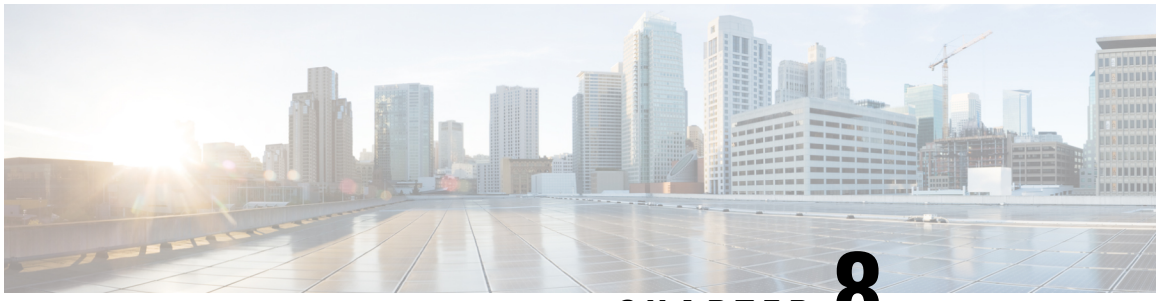
The **ip local-proxy-arp** command is not available on an SVI that uses **fabric forwarding mode anycast-gateway**, with or without the **proxy** keyword.

Verification commands for local-proxy ARP and Layer 2 isolation

Use these commands to verify the VLAN mapping, ARP entries, Layer 2 isolation, and service-acceleration status.

Table 1: Verification commands for local-proxy ARP and Layer 2 isolation

Command	Purpose
show interface Ethernet1/1 vlan mapping	Displays VLAN mappings on the specified interface.
show ip arp static vrf *vrf-name*	Displays static ARP entries for the specified VRF.
show vlan filter	Displays the VLAN filter configuration.
show vlan access-list *access-list-name*	Displays the VLAN access-list configuration.
show service-acceleration status details	Displays detailed service-acceleration status.
show ip interface vlan *vlan-id*	Displays whether local-proxy ARP is enabled on the VLAN interface.



CHAPTER 8

High Availability and Redundancy

High Availability and Redundancy

The High Availability (HA) feature ensures that the services remain available for both bridged and routed traffic during switch or DPU failure or outage. High Availability is achieved by enabling redundant paths for traffic flows and synchronizing the flow state across these paths, which is critical for stateful services.

HA peers

HA peering requires identical hardware platforms running the same software version. HA peering between two different hardware platforms is not supported, for example, if one Smart Switch is N9324C-SE1U and the other Smart Switch is N9348Y2C6D-SE1U.

The HA infrastructure ensures that the service firewall state is synchronized between configured peers. If the firewall service is not ready or becomes unavailable on a switch, HA disables the network paths through that switch and directs traffic to the other available peer. When two switches are configured for HA (Active/Active HA model), the service firewall state is synchronized between the two peering switches. Currently, only two smart switches can be configured as HA peers.

Note Due to VXLAN headers and the MAC-IP-UDP encapsulation for inline flow sync of traffic between HA peers, a host should not send jumbo frames larger than 9216 - 10B (for HA) - 30B (for macsec) = 9174 B of 9134B in case of VXLAN + MACsec. Effectively, the maximum packet size that is currently supported is 9000 bytes.

Connectivity between HA peers

HA communication requires Layer 3 connectivity with IPv4 addresses in the default VRF which you must provide between the two switches that make the HA pair. The security subsystem implements stateful inspection through flow and state synchronization.

Layer 3 connectivity between HA peers is required to carry:

- control plane traffic between the Hypershield Agents on the HA peers, for keepalives and state information exchange

- control plane traffic between the DPU pairs (DPU1 – DPUn) of the HA peers, for keepalives and flow state sync
- data traffic for inflow flow-sync for flow learns, and
- data traffic for inline flow-state sync for stateful traffic inspection.

The HA connectivity can be established using dedicated or shared links between the HA peers:

- Dedicated Port Channel: A specific port channel set up exclusively for the HA synchronization between peers.
- Shared Port Channel: Utilizes an existing port channel, such as a vPC peer link, already connecting the peers.

Note The links should be at least 25G and directly connected, but in most deployments the minimum bandwidth may be 2 x 400G unless traffic flows between the Smart Switches are symmetric. The port channel provides link redundancy. SVIs over Layer 2 port channels can also be used to establish this connectivity.

Guidelines and limitations

- HA connectivity over routed or VXLAN fabrics is not supported in 10.6(3s)F.
- NTP or PTP must be configured and be in sync between the two HA switches so that the time and flows are synchronized on both the switches
- For HA to work correctly, the DPU load-balancing configuration must be identical on both Smart Switches in the same HA pair, and the service VRF and service VLAN configuration must be identical on both devices.
- Even if HA is configured, temporary traffic interruptions might occur for asymmetric flows if there is a mismatch in policy versions, or flow-sync is incomplete or in-progress. You may also experience temporary traffic interruptions when traffic deflections occur in the network.
- Interruptions can be observed while establishing or recovering HA connectivity between switches with active firewall services.

Loss of connectivity between HA peers

If HA peers lose connectivity (a split-brain scenario), both switches disable the traffic deflection mechanism and operate as standalone firewall services. This state can result in two types of traffic issues:

- Dropped Traffic: Traffic arriving at a HA peer that is not ready can be dropped.
- Asymmetric Flow Failures: Independent inspection by both peers may lead to asymmetric flow failures.

Service Traffic Deflection

When high availability is configured, if the firewall service subsystem is not ready to process traffic despite the networking infrastructure being fully operational, traffic is deflected to the HA peer that is ready, to minimize traffic loss. This deflection is also necessary during system initialization, and when the firewall is

not in service. NX-OS manages this deflection for both routed and bridged traffic that is configured for traffic inspection.

Additionally, when the configured HA peers are detected as having mismatched software versions, incompatible platforms, differences in DPU load-balance mechanisms, or if the DPU pairs are unable to establish steady connectivity to each other, one of the HA peers having a fully functional firewall service takes-over all traffic as a firewall service that is `ready` and the other peer yields and transitions to a firewall service that is `not-ready`. Traffic is then deflected in a similar manner towards the ready firewall service.

Routed Traffic Deflection

Deflection of routed traffic to the HA peer is handled in the following ways:

- VRF contexts configured for routed-traffic inspection are isolated when the local service firewall is not ready.
  - This allows for routing protocols such as BGP, OSPF to withdraw routes or advertise routes with higher metrics than the HA peer, to the upstream devices.
  - Traffic arriving from the Layer 3 fabric is thereby deflected towards the HA peer.
  - On HA peers configured with HSRP, the HSRP groups in the isolated VRF transition gracefully to `INIT`, when the local service firewall is not ready. This allows the other peer to entirely take over the traffic for the HSRP gateway.
- On HA peers that are part of a vPC domain, the vPCs handling traffic for any VLANs, whose SVIs pertain to VRFs configured for service firewall are suspended, if the local service firewall is not ready, while the peer is ready.
  - This vPC suspension mechanism is conducted independent of the vPC role and is entirely based on the service firewall readiness.
  - vPC peers support active-active forwarding for first hop redundancy protocols such as HSRP, VRRP, and for anycast gateway in VXLAN. When a vPC is suspended, the traffic from dually homed hosts is directed only to the ready firewall service.
  - In a VXLAN fabric, type-5 routes for VRFs configured for routed-traffic inspection are always advertised with the PIP by the vPC leaf switches, regardless of the advertise-pip configuration. When VRF isolation takes effect, traffic is deflected to the ready HA peer, as only a single preferred path is available.
  - When HA state is ready/ready, the EVPN route is advertised with VIP from both peers. However, when one of the peer is put in maintenance mode (no-inservice) the peer switch advertises the routes with PIP. When the peer switch is put back to in-service mode, the routes are again advertised with VIP. These transitions from VIP to PIP or PIP to VIP can cause momentary traffic loss on the box which actively forwards traffic. PIP is only advertised when the HA state is ready/not ready.
  - In a VXLAN fabric, type-2 routes for VPC hosts in VRFs configured for routed-traffic inspection are advertised with the VIP when both HA peers are ready. These routes are withdrawn by the HA peer that is not ready and will be advertised with the PIP of the ready peer.

Bridged Traffic Deflection

Deflection of bridged traffic to the HA peer is handled in these ways:

- On HA peers that are part of a VPC domain, the VPCs handling traffic for any VLANs that are configured for bridged traffic inspection, are suspended when the local service firewall is not ready, and the peer is ready.
- In a VXLAN fabric, MAC routes and Type-3 Inclusive Multicast Ethernet Tag (IMET) routes in the VLANs configured for bridged-traffic inspection are advertised with the VIP when both HA peers are ready. These routes are withdrawn by the HA peer that is not ready and are advertised with the PIP of the ready peer.

Show commands

The output of the **show service-acceleration status details** command can show the VLANs in `forwarding ready` state when bridged traffic in those VLANs can be successfully redirected to the DPUs for inspection. When high-availability is configured and HA peering is successfully established, bridged traffic from vPCs or VXLAN fabric can arrive at the Smart Switch only when the firewall service state for the Smart Switch is ready.

Smart Switches deployed as vPC peers

Smart Switches can be deployed as vPC peers. Users must ensure configuration consistency for the feature service-acceleration and service firewall configuration across the two vPC peers.

Users must ensure firewall policies to permit traffic are available for all the interface IPs used by the individual vPC peers, when VPC peer-gateway is enabled and the VRFs pertaining to these interface IPs are enabled for service firewall. Traffic that is destined to interface IPs of a vPC peer, that hashes to the other peer when peer gateway feature is enabled, is inspected by the DPU on the other peer, before being sent over the peer link.

Configure vPC delay-restore timers of 300s and vPC auto-recovery reload-delay timers of 360s to ensure that the HA peers are able to detect each other during switch reloads, before the vPCs are recovered.

Traffic loss may occur if vPCs are down on the smart switch with the firewall service that is ready and vPCs are suspended on the non-ready HA peer.

If the vPCs carry traffic associated with VRFs or VLANs that are not configured for firewall inspection, such traffic may be impacted or deflected when the traffic deflection mechanisms take effect.

If HA is configured on vPC peers, traffic deflection for hosts that are singly homed or connected through orphan ports to only one vPC peer is not supported. When the firewall is not in-service, traffic from orphan hosts can drop. In other failover scenarios of peer incompatibility, where the firewall is in-service on both peers, traffic from orphan hosts can be independently inspected, without flow-sync taking effect.

Traffic deflection topologies

This section discusses three scenarios of Smart Switches with high-availability configured:

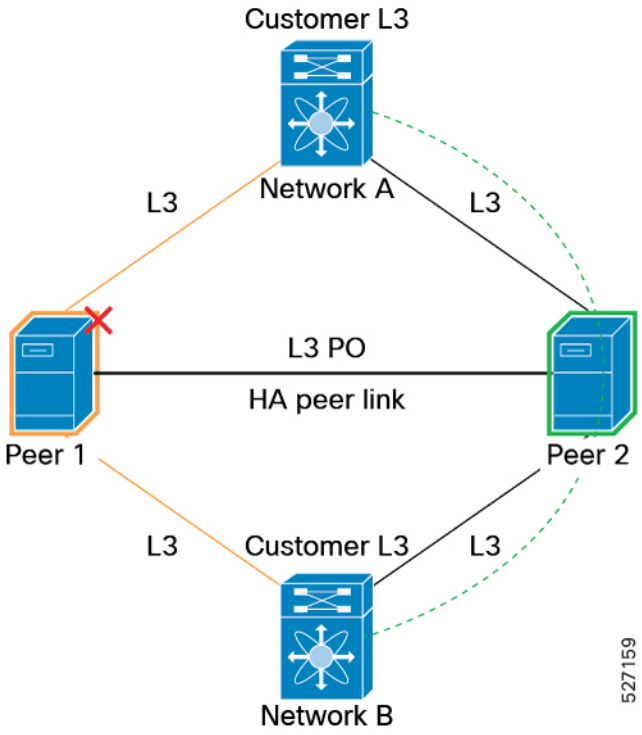
- Smart Switches as transit routers
- Smart Switches as Active/Standby HSRP GW
- Smart Switches as vPC peers

Smart Switches as transit routers

The diagram illustrates a High Availability (HA) topology where two peer devices (Peer 1 and Peer 2) act as transit routers between customer networks (Network A and Network B) and the broader infrastructure. The

devices are connected through a Layer 3 Port Channel (L3 PO), which serves as the HA peer link for state synchronization and control signaling.

Figure 3: Topology for Smart Switches as transit routers



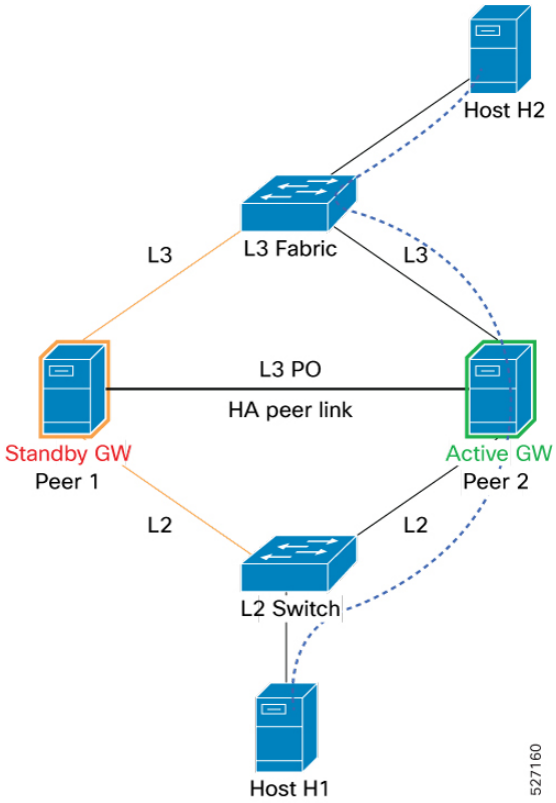
In this topology, VRF isolation influences route advertisements and deflects traffic towards Peer 2.

This topology uses VRF-aware routing to ensure that traffic is always directed to the healthy node. By isolating the routing tables, the network can perform graceful traffic redirection, ensuring that even if one transit router (Peer 1) goes offline, the other (Peer 2) seamlessly assumes the routing responsibility without disrupting the customer networks.

Smart Switches as Active/Standby HSRP GW

This diagram illustrates a High Availability (HA) Gateway deployment, typically used in enterprise networks to ensure continuous connectivity for hosts. The setup consists of two peers, Peer 1 and Peer 2, connected through a Layer 3 Port Channel (L3 PO), which acts as the HA peer link.

Figure 4: Topology for Smart Switches as Active/Standby HSRP GW



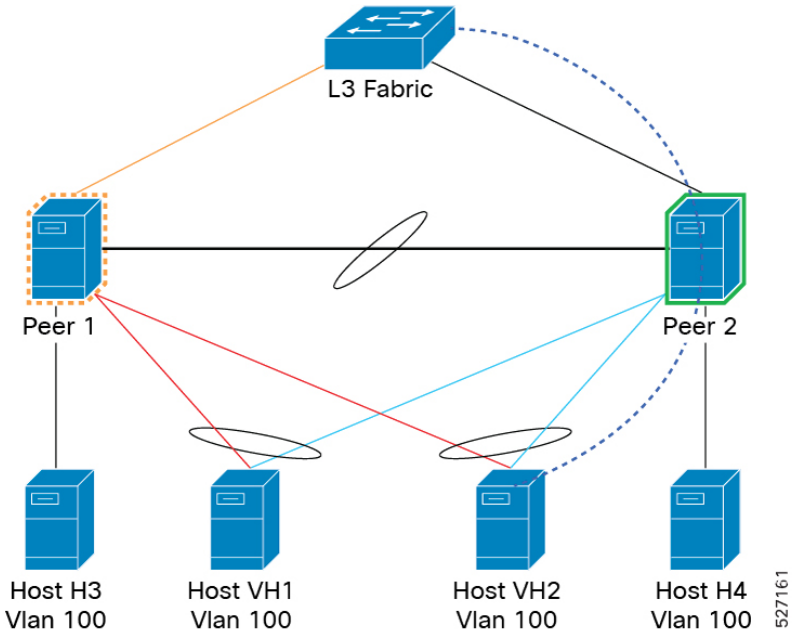
In this topology, HSRP moves to INIT on Peer 1 and all traffic is deflected to Peer 2. VRF isolation deflects the traffic from Layer 3 fabric to Peer 2.

This architecture ensures that even if one peer becomes unavailable, the network intelligence—driven by HSRP and VRF isolation—seamlessly reroutes traffic to the active peer, maintaining uninterrupted communication for the connected hosts.

Smart Switches as vPC peers

This diagram illustrates a network maintenance or failure scenario in a Virtual Port Channel (vPC) environment. The setup features two peers, Peer 1 and Peer 2, connected to a common Layer 3 Fabric and various hosts (H3, VH1, VH2, H4), all operating within VLAN 100.

Figure 5: Topology for Smart Switches as vPC peers



In this topology, vPCs on Peer 1 are suspended to deflect traffic from hosts to Peer 2 and VRF isolation deflects the traffic from Layer 3 fabric to Peer 2.

The combination of suspending vPCs on the inactive peer and leveraging VRF isolation to redirect traffic from the L3 fabric ensures that all communication—both from the hosts and from the core network—is seamlessly shifted to Peer 2. This allows for the graceful isolation of Peer 1 without disrupting the connectivity of the hosts in VLAN 100.

Configure connectivity for High Availability

Use this procedure to configure the High Availability (HA) connectivity, establish the loopback IP reachability through static routes or routing protocols over this HA connection.

Procedure

Step 1 Perform any one of these procedures based on the interface.

- Configure High Availability connectivity over Layer 3 interfaces
- Configure High Availability connectivity over Layer 2 interfaces or vPC peer link

Step 2 Create a loopback interface for the HA source-interface using the **interface loopback** *instance* command.

Example:

```
switch(config)# interface loopback 101
switch(config-if)#
```

Step 3 Configure an IP address for the interface using the **ip address** *ip-address/length* command.

Example:

```
switch(config-if)# ip address 192.0.2.2/32
```

For more information about IP addresses, refer to Cisco N9000 Series NX-OS Unicast Routing Configuration Guide.

ip-address/length: Sets an IP address for the loopback.

Note

- You have to configure /32 as length of the IP address. Only IPv4 addresses are supported.
- The loopback interface created for Controller connectivity can be used for HA source interface.
- The loopback interface must be configured in the default VRF.

Step 4 Verify the loopback interface configuration.

Example:

```
switch# show run
!
interface loopback101
ip address 192.0.2.2/32
```

Note
To establish connection between the loopback IP of the two HA peers, use static routes or routing protocols over this HA connection. For more information, refer to Cisco N9000 Series NX-OS Unicast Routing Configuration Guide.

Configure High Availability connectivity over Layer 2 interfaces

Provision and configure HA connectivity over vPC peer link or Layer 2 port channel over vPC peer link if the HA peers are vPC peers and part of the same vPC domain. The vPC peer link must allow the VLAN used for HA connectivity. Otherwise, a Layer 2 port channel can be used if the HA peers are not vPC peers.

Procedure

Step 1 Create VLAN in global configuration mode using the **vlan** *vlan-id* command.

Example:

```
switch# configure terminal
switch(config)# vlan vlan20
```

Step 2 Use the **name** *vlan name* command to assign a descriptive name to the VLAN to help identify its function as High Availability.

Example:

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

Step 3 Use the **interface vlan** *vlan-id* command to enter the configuration mode for the Switch Virtual Interface (SVI).

Example:

```
switch(config)# interface vlan 100
```

Step 4 Use the **description** *descriptive label* **SVI** command to add a label to the SVI to clarify its role as the High Availability link's virtual interface.

Example:

```
switch(config-if)# description ha-link SVI
```

Step 5 Use the **ip address** *ip-address/length* command to assign an IP address and subnet mask to the SVI, allowing the switch to route traffic for this VLAN. This provides HA loopback reachability over SVI.

Example:

```
switch(config-if)# ip address 192.168.100.1/24
```

Step 6 Use the **no shutdown** command to enable the interface.

Example:

```
switch(config-if)# no shutdown
switch(config-if)# exit
```

Step 7 Use the **interface port-channel** *port-channel-number* command to access the configuration mode for an existing Port-Channel (vPC peer-link).

Example:

```
switch(config)# interface port-channel 20
```

Step 8 Use the **switchport trunk allowed vlan add** *vlan-id* command to add HA VLAN to the allowed VLAN.

Example:

```
switch(config-if)# switchport trunk allowed vlan add 100
```

Step 9 Use the **exit** command to exit the current interface configuration mode and return to the global configuration mode.

Example:

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

What to do next

For the next steps, refer to Configure connectivity for High Availability, on page 55.

Configure High Availability connectivity over Layer 3 interfaces

Provision and configure HA connectivity over dedicated Layer 3 port channel.

Procedure

Step 1 Use the **configure terminal** command to enter the global configuration mode.

Example:

```
switch# configure terminal
```

Step 2 Use the **interface port-channel** *port-channel-number* command to specify the port-channel interface to configure and enter the interface configuration mode.

Example:

```
switch(config)# interface port-channel 10
```

Step 3 Use the **description** *descriptive label* command to assign a descriptive label to the interface, making it easier to identify it as a High Availability link during troubleshooting or management.

Example:

```
switch(config-if)# description ha-link
```

Step 4 Use the **no switchport** command to configure the interface as a Layer 3 (routed) port instead of a Layer 2 (switching) port, allowing it to support IP addressing and routing protocols.

Example:

```
switch(config-if)# no switchport
```

Step 5 Use the **ip address** *ip-address/length* command to assign an IP address and subnet mask to the routed interface, enabling it to communicate with the peer device on the other end of the link.

Example:

```
switch(config-if)# ip address 10.1.1.1/30
```

Step 6 Use the **exit** command to exit the current interface configuration mode and return to the global configuration mode.

Example:

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

What to do next

For the next steps, refer to Configure connectivity for High Availability, on page 55.

Configure High Availability

After setting up High Availability (HA) connectivity and establishing the loopback IP reachability, set up the high-availability configuration for the Hypershield service subsystem.

Perform these steps to establish HA connectivity and configure HA.

Procedure

Step 1 Use the **service system hypershield** command to enter service system mode and access the configuration mode for the Hypershield system.

Example:

```
switch(config)# service system hypershield
```

Step 2 Use the **high-availability** command to enter the sub-mode specifically for high-availability settings.

Example:

```
switch(config-svc-sys)# high-availability
```

Step 3 Set the source interface using the **source-interface** command to designate the interface whose IP address is used as the source address for the HA connectivity.

Example:

```
switch(config-svc-sys-ha)# source-interface loopback101
```

Note
The source interface should be a loopback interface.

Step 4 Use the **peer** command to specify the HA IP address of the peer switch. Note that the peer IP address must match the IP address of the configured HA source-interface on the peer.

Example:

```
switch(config-svc-sys-ha)# peer 192.168.10.2
```

Note
Only a single peer IP may be configured at any time and must be an IPv4 address.

Step 5 Activate the configuration using the **no shutdown** command to enable HA.

Example:

```
switch(config-svc-sys-ha)# no shutdown
```

Note
High Availability remains inactive until this command is executed and service firewall is configured.

Step 6 . Use the **exit** command to exit the configuration mode.

Example:

```
switch(config-svc-sys-ha)# exit
switch(config-svc-sys)# exit
switch(config)# exit
```

Verify High Availability

Run the show commands and verify the service-acceleration and High Availability configuration.

Use the **show running-config service-acceleration** command to view the configuration of the service acceleration feature on the switch.

```
switch# show running-config service-acceleration
feature service-acceleration
service system hypershield
  source-interface loopback100
  high-availability
    source-interface loopback101
    peer 192.168.10.2
    no shutdown
  service firewall
    vrf blue module-affinity dynamic
    vrf red module-affinity dynamic
    vlan id 100 bridged-traffic module-affinity dynamic
    in-service
```

Use the **show service-acceleration high-availability status** command to view the high-availability status of the service acceleration system.

```
switch# show service-acceleration high-availability status
  Service System: hypershield
```

```
HA Source Interface: loopback101 (192.0.2.2)
HA Admin State: no-shutdown
Agent Status for Service Firewall: ready
Agent HA Status: ready
Peers:
  PeerIP              Peer Service State  HA State with Peer  Reason
  ------------------- ------------------  ------------------  --------------
  192.168.10.2           ready               ha-ok               all criteria met
```