Configuring VM-FEX

This chapter contains the following sections:

- Information About VM-FEX, page 1
- Licensing Requirements for VM-FEX, page 3
- Default Settings for VM-FEX, page 3
- Configuring VM-FEX, page 4
- Verifying the VM-FEX Configuration, page 12

Information About VM-FEX

VM-FEX Overview

Based on the (prestandard) IEEE 802.1Qbh port extender technology, Cisco Virtual Machine Fabric Extender (VM-FEX) extends the fabric from the switch chassis to the Virtual Machine (VM). Each VM is associated with a network adapter vNIC, which is associated with a virtual Ethernet (vEthernet or vEth) port on the parent switch. This dedicated virtual interface can be managed, monitored, and spanned in the same way as a physical interface. Local switching in the hypervisor is eliminated, with all switching being performed by the physical switch.

VM-FEX Components

Server

VM-FEX is supported by Cisco UCS C-Series rack-mount servers with the VMware virtualization environment as the hypervisor.

The configuration of the server is performed using the Cisco Integrated Management Controller (CIMC) interface, which provides both a GUI and a CLI interface. The configuration of the hypervisor and virtualization services is performed using the VMware vSphere client.

For information about CIMC and VM-FEX configuration, see the following documents:

- *Cisco UCS C-Series Servers Integrated Management Controller GUI Configuration Guide*
- *Cisco UCS Manager VM-FEX for VMware GUI Configuration Guide*

Virtual Interface Card Adapter

VM-FEX is supported by the Cisco UCS P81E Virtual Interface Card (VIC), a dual-port 10 Gigabit Ethernet PCIe adapter that supports static or dynamic virtualized interfaces, including up to 128 virtual network interface cards (vNICs).

The configuration of the VIC and its vNICs is performed using the CIMC interface on the Cisco UCS C-Series servers.

FEX

The physical ports of the server can be connected directly to the switch or to a fabric extender (FEX) connected to the switch. VM-FEX is supported by the Cisco Nexus Fabric Extender.

VM-FEX and AFEX require that the FEX is connected with a fabric PO and not individual links.

Switch

VM-FEX is supported by the Cisco Nexus device. Although a single switch chassis can be connected with VM-FEX, a typical application uses a pair of switches deployed as a virtual port channel (vPC) domain.

On the switch, a vEthernet interface represents the vNIC. All operations performed by the network administrator are performed on the vEthernet interface.

VM-FEX Terminology

The following terms are used in describing VM-FEX components and interfaces:

virtual Ethernet interface

A virtual Ethernet interface (vEthernet or vEth) represents the switch port that is connected to the vNIC of a virtual machine. Unlike a traditional switch interface, a vEth interface's name does not indicate the module with which the port is associated. Where a traditional physical switch port is specified as GigX/Y, where X is the module number and Y is the port number on the module, a vEth interface is specified as vEthY. This notation allows the interface to keep the same name when the VM migrates to another physical server.

dynamic interface

A dynamic interface is a vEthernet interface that is configured automatically as a result of adapter and switch communications. The provisioning model of a dynamic interface consists of the configuration on the switch of a vEthernet port profile, which is propagated to the network adapter as a port group, followed by the association of the port group with the vNIC. The port profile is created in the switch by the network administrator, while the association with the vNIC is performed on the adapter by the server administrator.

static interface

A static interface is configured manually on the switch and the adapter. A static virtual adapter can be a vNIC or a virtual host adapter bus (vHBA). A static interface can be a vEthernet or a virtual Fibre Channel (vFC) interface bound to a static vEthernet interface.

In one method of creating a static vEthernet, the network administrator assigns a channel number (equivalent to a VN-Tag or prestandard IEEE 802.1BR tag number) to the vEthernet. The server administrator must be sure to define a vNIC on the adapter with the same channel number.

In another method, the network administrator can create a static floating vEthernet by configuring the vEthernet with a virtual switching instance (VSI) MAC address and DVPort ID.

floating vEthernet interface

In a hypervisor environment, each vNIC on the network adapter is associated with one virtual machine (VM). VMs can migrate from one physical server to another. A virtual interface that migrates with a VM and virtual network link is called a floating vEthernet interface.

fixed vEthernet interface

A fixed vEthernet interface is a virtual interface that does not support migration across physical interfaces. For fixed vEthernet (static or dynamic), an administrator can change configurations at any time. The binding of the vEthernet interface number to a channel number is persistent unless the administrator changes it.

Licensing Requirements for VM-FEX

The following table shows the licensing requirements for this feature:

Product	License Requirement
Cisco NX-OS	A VM-FEX license is required for each Cisco Nexus device. The license package name is VMFEX_ FEATURE_PKG. A grace period of 120 days starts when you first configure the licensed feature. For a complete explanation of the Cisco NX-OS licensing scheme and how to obtain and apply licenses, see the *Cisco NX-OS Licensing Guide*.

Default Settings for VM-FEX

The following table lists the default settings for parameters that are relevant to VM-FEX:

Parameters	Default
Virtualization feature set	Disabled
FEX	Disabled
VM-FEX	Disabled

Parameters	Default
LLDP	Enabled
vPC	Disabled
svs vethernet auto-setup	Enabled
FCoE	Disabled

Configuring VM-FEX

Overview of the VM-FEX Configuration Steps

The following steps outline the necessary sequence of procedures for configuring VM-FEX between the switch and the server hosting the VMs. Procedures to be performed on the switch are described in this document. For procedures to be performed on the server or the VMware vCenter, refer to the server and vCenter documentation.

Procedure

Step 1 Server: Create vNICs on VIC adapter.
a) Create two static vNICs to be used as uplinks from the host.
b) Create up to 112 VM-FEX interfaces.
c) Reboot the server.

Step 2 Switch: Enable VM-FEX and other required services.
See Enabling Features Required for VM-FEX, on page 5.

Step 3 Switch: Configure two static vEthernet interfaces and bind them to the physical port and channel.
See Configuring the Fixed Static Interfaces, on page 6.

Step 4 Switch: Define port profiles to be associated with the VMs.
See Configuring a Port Profile for the Dynamic Interfaces, on page 9.

Step 5 Switch: Verify that the two static vEthernet interfaces are active and associated with the vEthernet interfaces of the switch.
See Verifying the Status of the Virtual Interfaces, on page 12.

Step 6 Switch and vCenter: Install XML certificate from switch to vCenter.
a) Switch: Enable HTTP using the **feature http** command in global configuration mode.
b) From a web browser, access the IP address of the switch and download the displayed XML certificate.
c) Switch: Disable HTTP using the **no feature http** command in global configuration mode.
d) vCenter: Install the XML certificate plugin.

Step 7 Switch: Enable vPC and register the vPC system to the vCenter as a distributed virtual switch (DVS).
See Configuring an SVS Connection to the vCenter Server, on page 10.

Step 8 vCenter: Create a datacenter on the vCenter.

Step 9 Switch: Activate and verify the SVS connection to the vCenter.
See Activating an SVS Connection to the vCenter Server, on page 11 and Verifying the Connection to the vCenter Server, on page 14.

Step 10 vCenter: Verify that the port profiles (port groups) are propagated to the vCenter.

Step 11 Server: Add resources to the DVS.

a) Add the ESX host to the DVS.
b) Add the static vNICs as uplinks to the DVS.
c) Associate VMs to the port groups defined by the switch.
d) Activate the VMs.

Step 12 Switch: Verify that the dynamic vNICs are active, assigned to VMs, and connected to the vEthernet interfaces of the switch.
See Verifying the Status of the Virtual Interfaces, on page 12.

Step 13 Server: Verify that the interfaces are active and assigned to the VMs.

Step 14 vCenter: Verify that the dynamic vNICs are active.

Enabling Features Required for VM-FEX

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	**install feature-set virtualization**	Installs the virtualization feature set on the switch.
Step 3	**feature-set virtualization**	Enables the virtualization feature set on the switch. This feature set enables the use of static vEthernet interfaces.
Step 4	**feature fex**	Enables FEX features on the switch.
Step 5	**feature vmfex**	Enables VM-FEX features on the switch. This feature set enables the use of dynamic vEthernet interfaces.
Step 6	**feature vpc**	Enables a virtual port channel (vPC) on the switch.
Step 7	**vethernet auto-create**	(Optional) Globally enables the automatic creation of virtual Ethernet interfaces. This feature is not required if the fixed vEthernet interfaces are statically configured.
Step 8	**feature fcoe**	(Optional) Enables Fibre Channel over Ethernet (FCoE) on the switch.

	Command or Action	Purpose
Step 9	**end**	(Optional) Return to privileged EXEC mode.
Step 10	**copy running-config startup-config**	(Optional) Copies the running configuration to the startup configuration.
Step 11	**reload**	(Optional) Reloads the switch.

This example shows how to enable the features required for VM-FEX:

```
switch# configure terminal
switch(config)# install feature-set virtualization
switch(config)# feature-set virtualization
switch(config)# feature fex
switch(config)# feature vmfex
switch(config)# feature vpc
switch(config)# vethernet auto-create
switch(config)# feature fcoe
switch(config)# end
switch# copy running-config startup-config
switch# reload
```

Configuring the Fixed Static Interfaces

You can configure two physical interfaces and binds two virtual interfaces to each physical interface, creating fixed static vEthernet interfaces. For more information on configuring fixed static interfaces, see the Adapter-FEX Configuration Guide for your device.

With redundant switches, you can perform the following procedure with identical settings on both the primary and secondary switches.

Before You Begin

- VM-FEX and other required services must be enabled on the switches.
- Two static vNICs must be configured on the VIC adapter installed in the host server.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	**interface ethernet** *slot/port*	Enters interface configuration mode for the first Ethernet port. **Note** If this is a 10G breakout port, the *slot*/*port* syntax is *slot*/*QSFP-module*/*port*.

	Command or Action	Purpose
Step 3	**shutdown**	Disables local traffic on the interface. **Note** Shutting down the interface before enabling VN-Tag mode prevents the dynamic creation of a fixed vEthernet interface.
Step 4	**switchport mode vntag**	Enables port extender support on the interface.
Step 5	**interface ethernet** *slot/port*	Enters interface configuration mode for the second Ethernet port. **Note** If this is a 10G breakout port, the *slot*/*port* syntax is *slot*/*QSFP-module*/*port*.
Step 6	**shutdown**	Disables local traffic on the interface.
Step 7	**switchport mode vntag**	Enables port extender support on the interface.
Step 8	**interface vethernet** *interface-number*	Enters configuration mode for the first virtual interface for the first Ethernet port.
Step 9	**bind interface ethernet** *slot/port* **channel** *channel-number*	Binds the virtual interface to the physical interface and the specified port channel. **Note** The port channel numbers of the virtual interfaces must match those configured on the vNICs. **Note** If this is a 10G breakout port, the *slot*/*port* syntax is *slot*/*QSFP-module*/*port*.
Step 10	**no shutdown**	Enables local traffic on the interface.
Step 11	**interface vethernet** *interface-number*	Enters configuration mode for the second virtual interface for the first Ethernet port.
Step 12	**bind interface ethernet** *slot/port* **channel** *channel-number*	Binds the virtual interface to the physical interface and the specified port channel. **Note** If this is a 10G breakout port, the *slot*/*port* syntax is *slot*/*QSFP-module*/*port*.
Step 13	**no shutdown**	Enables local traffic on the interface.
Step 14	**interface vethernet** *interface-number*	Enters configuration mode for the first virtual interface for the second Ethernet port.
Step 15	**bind interface ethernet** *slot/port* **channel** *channel-number*	Binds the virtual interface to the physical interface and the specified port channel. **Note** If this is a 10G breakout port, the *slot*/*port* syntax is *slot*/*QSFP-module*/*port*.
Step 16	**no shutdown**	Enables local traffic on the interface.
Step 17	**interface vethernet** *interface-number*	Enters configuration mode for the second virtual interface for the second Ethernet port.

	Command or Action	Purpose
Step 18	**bind interface ethernet** *slot/port* **channel** *channel-number*	Binds the virtual interface to the physical interface and the specified port channel. **Note** If this is a 10G breakout port, the *slot*/*port* syntax is *slot*/*QSFP-module*/*port*.
Step 19	**no shutdown**	Enables local traffic on the interface.
Step 20	**interface ethernet** *slot/port*	Enters configuration mode for the first Ethernet port. **Note** If this is a 10G breakout port, the *slot*/*port* syntax is *slot*/*QSFP-module*/*port*.
Step 21	**no shutdown**	Enables local traffic on the interface.
Step 22	**interface ethernet** *slot/port*	Enters configuration mode for the second Ethernet port. **Note** If this is a 10G breakout port, the *slot*/*port* syntax is *slot*/*QSFP-module*/*port*.
Step 23	**no shutdown**	Enables local traffic on the interface.
Step 24	With redundant switches, repeat this procedure with identical settings on the secondary switch.	

This example shows how to configure two physical interfaces, binds two virtual interfaces to each physical interface, and enables the interfaces:

```
switch-1# configure terminal
switch-1(config)# interface ethernet 1/17
switch-1(config-if)# shutdown
switch-1(config-if)# switchport mode vntag
switch-1(config-if)# interface ethernet 1/18
switch-1(config-if)# shutdown
switch-1(config-if)# switchport mode vntag

switch-1(config-if)# interface vethernet 1
switch-1(config-if)# bind interface ethernet 1/17 channel 10
switch-1(config-if)# no shutdown
switch-1(config-if)# interface vethernet 3
switch-1(config-if)# bind interface ethernet 1/17 channel 11
switch-1(config-if)# no shutdown

switch-1(config-if)# interface vethernet 2
switch-1(config-if)# bind interface ethernet 1/18 channel 10
switch-1(config-if)# no shutdown
switch-1(config-if)# interface vethernet 4
switch-1(config-if)# bind interface ethernet 1/18 channel 11
switch-1(config-if)# no shutdown

switch-1(config-if)# interface ethernet 1/17
switch-1(config-if)# no shutdown
switch-1(config-if)# interface ethernet 1/18
switch-1(config-if)# no shutdown

switch-1(config-if)#
```

What to Do Next

Verify the status of the connection between the static interfaces and the static vNICs on the host server.

Configuring a Port Profile for the Dynamic Interfaces

You can configure a port profile for dynamic virtual interfaces. This port profile is exported to the VMware vCenter distributed virtual switch (DVS) as a port-group.

With redundant switches, you can perform the following procedure with identical settings on both the primary and secondary switches.

Before You Begin

- Dynamic vNICs must be configured on the VIC adapter installed in the host server.
- The VLAN specified in the port profile must be created.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	**port-profile type vethernet** *profilename*	Enters configuration mode for the specified port profile, creating it if necessary.
Step 3	**switchport mode access**	(Optional) Configures the interface to be in access mode.
Step 4	**switchport access vlan** *vlan-id*	(Optional) Specifies the VLAN when the interface is in access mode.
Step 5	**dvs-name** {**all** | *name*}	Specifies the vCenter DVS to which the port profile is exported as a port-group. With the keyword **all**, the port profile is exported to all DVSs in the vCenter.
Step 6	**port-binding dynamic**	(Optional) Specifies dynamic port binding. The port is connected when the VM is powered on and disconnected when the VM is powered off. Max-port limits are enforced. The default is static port binding.
Step 7	**state enabled**	Enables the port profile.

This example configures a port profile for dynamic virtual interfaces:

```
switch-1# configure terminal
switch-1(config)# port-profile type vethernet vm-fex-vlan-60
switch-1(config-port-prof)# switchport mode access
switch-1(config-port-prof)# switchport access vlan 60
switch-1(config-port-prof)# dvs-name all
switch-1(config-port-prof)# port-binding dynamic
switch-1(config-port-prof)# state enabled
```

```
switch-1(config-port-prof)#
```

Configuring an SVS Connection to the vCenter Server

You can configure a secure connection from the switch to the vCenter Server.

With redundant switches, perform this procedure on both the primary and the secondary switches. In normal operation, only the primary switch connects to the vCenter, with the secondary switch connecting only upon a failure of the primary.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	**svs connection** *svs-name*	Enables and enters configuration mode for an SVS connection from the switch to the vCenter Server.
Step 3	**protocol vmware-vim**	Enables the VMware Infrastructure Software Development Kit (VI SDK), which allows clients to communicate with the vCenter.
Step 4	**vmware dvs datacenter-name** *dc-name*	Creates a VMware distributed virtual switch (DVS) in the specified datacenter.
Step 5	**dvs-name** *dvs-name*	Configures a name for the DVS in the vCenter Server.
Step 6	Choose one: • **remote ip address** *ipv4-addr* [**port** *port-num*] [**vrf** {*vrf-name* | **default** | **management**}] • **remote hostname** *host-name* [**port** *port-num*] [**vrf** {*vrf-name* | **default** | **management**}]	Specifies the hostname or IP address for the vCenter Server. Optionally, specifies the port number and VRF.
Step 7	**install certificate** {**bootflash:**[*//server/*] | **default**}	Installs a certificate that is used to connect to the vCenter Server. The *server* argument specifies the boot flash memory location to install the certificate. The argument value can be **module-1**, **sup-1**, **sup-active**, or **sup-local**.
Step 8	**extension-key:** *extn-ID*	Configures the extension key to be used to connect to the vCenter Server. **Note** With redundant switches, perform this step only on the primary switch. The key is automatically synchronized with the secondary switch.

This example shows how to configure the SVS connection on the primary switch and the secondary switch:

```
switch-1# configure terminal
switch-1(config)# svs connection 2VC
switch-1(config-svs-conn)# protocol vmware-vim
switch-1(config-svs-conn)# vmware dvs datacenter-name DC1
switch-1(config-svs-conn)# dvs-name Pod1
switch-1(config-svs-conn)# remote ip address 192.0.20.125 port 80 vrf management
switch-1(config-svs-conn)# install certificate default
switch-1(config-svs-conn)# extension-key: Cisco_Nexus_6004_1543569268
switch-1(config-svs-conn)#

switch-2# configure terminal
switch-2(config)# svs connection 2VC
switch-2(config-svs-conn)# protocol vmware-vim
switch-2(config-svs-conn)# vmware dvs datacenter-name DC1
switch-2(config-svs-conn)# dvs-name Pod1
switch-2(config-svs-conn)# remote ip address 192.0.20.125 port 80 vrf management
switch-2(config-svs-conn)# install certificate default
switch-2(config-svs-conn)#
```

What to Do Next

Activate the connection on the primary switch only.

Activating an SVS Connection to the vCenter Server

You can activate a connection from the switch to the vCenter Server.

Before You Begin

- The vCenter Server must be running and reachable.
- You must have already registered an extension with the vCenter Server.
- The SVS connection must be configured on the switch.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	**svs connection** *svs-name*	Enables and enters configuration mode for an SVS connection from the switch to the vCenter Server.
Step 3	[**no**] **connect**	Initiates a connection with the vCenter Server. **Note** With redundant switches, perform this step on both the primary and secondary switches. Only the primary will connect. The switch connects to the vCenter and becomes a DVS.

This example shows how to connect to a vCenter Server:

```
switch-1# configure terminal
switch-1(config)# svs connection 2VC
```

```
switch-1(config-svs-conn)# connect
Note: Command execution in progress..please wait
switch-1(config-svs-conn)#
```

Verifying the VM-FEX Configuration

Verifying the Status of the Virtual Interfaces

Use the following commands to display status information for virtual interfaces.

Command	Purpose
show interface vethernet *interface-number* [**detail**]	Displays the status of the virtual interface. Perform this procedure on each static virtual interface to verify that the interface is active and bound to the physical interface.
show interface virtual status vm-fex	Displays information about all floating virtual interfaces.
show interface virtual summary vm-fex	Displays summary information about virtual Ethernet interfaces.
show interface virtual status bound interface ethernet *port/slot*	Displays information about virtual interfaces on a bound Ethernet interface.
show interface virtual summary bound interface ethernet *port/slot*	Displays summary information about virtual interfaces on a bound Ethernet interface.

This example shows how to display status and configuration information about a static interface:

```
switch-1# show interface vethernet 1

Vethernet1 is up
Bound Interface is Ethernet1/17
Hardware is Virtual, address is 0005.73fc.24a0
Port mode is access
Speed is auto-speed
Duplex mode is auto
300 seconds input rate 0 bits/sec, 0 packets/sec
300 seconds output rate 0 bits/sec, 0 packets/sec
Rx
0 unicast packets  0 multicast packets  0 broadcast packets
0 input packets  0 bytes
0 input packet drops
Tx
0 unicast packets  0 multicast packets  0 broadcast packets
0 output packets  0 bytes
0 flood packets
0 output packet drops

switch-1# show interface vethernet 1 detail

vif_index: 20
--------------------------
  veth is bound to interface Ethernet1/17 (0x1a010000)
```

```
  priority: 0
  vntag: 16
  status: active
  channel id: 10
  registered mac info:
      vlan 0 - mac 00:00:00:00:00:00
      vlan 0 - mac 58:8d:09:0f:0b:3c
      vlan 0 - mac ff:ff:ff:ff:ff:ff

switch-1#
```

This example shows how to display status and summary information about all virtual interfaces:

```
switch-1# show interface virtual status vm-fex

Interface VIF-index   Bound If      Chan  Vlan  Status     Mode     Vntag
---------------------------------------------------------------------------
Veth32769 VIF-37      Eth1/20       ----   101  Up         Active       7
Veth32770 VIF-39      Eth1/20       ----     1  Up         Active       8
Veth32771 VIF-41      Eth1/20       ----     1  Up         Standby      9
Veth32772 VIF-43      Eth1/20       ----     1  Up         Active      10
Veth32773 VIF-47      Eth1/20       ----     1  Up         Active      12
Veth32774 VIF-48      Eth1/20       ----     1  Up         Standby     13
Veth32775 VIF-49      Eth1/20       ----     1  Up         Active      14


switch-1# show interface virtual summary vm-fex

Veth       Bound      Channel/  Port                     Mac               VM
Interface  Interface  DV-Port   Profile                  Address           Name
-------------------------------------------------------------------------------------
Veth32769  Eth1/20    7415    Unused_Or_Quarantine_Veth  00:50:56:9b:33:a7  ESX145_1_RH55.
Veth32770  Eth1/20    7575    Unused_Or_Quarantine_Veth  00:50:56:9b:33:a8  ESX145_1_RH55.
Veth32771  Eth1/20    7576    Unused_Or_Quarantine_Veth  00:50:56:9b:33:a9  ESX145_1_RH55.
Veth32772  Eth1/20    7577    Unused_Or_Quarantine_Veth  00:50:56:9b:33:aa  ESX145_1_RH55.
Veth32773  Eth1/20    7578    Unused_Or_Quarantine_Veth  00:50:56:9b:33:ac  ESX145_1_RH55.
Veth32774  Eth1/20    7579    Unused_Or_Quarantine_Veth  00:50:56:9b:33:ad  ESX145_1_RH55.
Veth32775  Eth1/20    7580    Unused_Or_Quarantine_Veth  00:50:56:9b:33:ae  ESX145_1_RH55.
Veth32776  Eth1/20    7607    Unused_Or_Quarantine_Veth  00:50:56:9b:33:ab  ESX145_1_RH55.

switch-1#
```

This example shows how to display status and summary information about fixed vEthernet interfaces:

```
switch-1# show interface virtual status bound interface ethernet 1/20

Interface VIF-index   Bound If      Chan  Vlan  Status     Mode     Vntag
---------------------------------------------------------------------------
Veth32769 VIF-16      Eth1/20          1     1  Up         Active       2
Veth32770 VIF-17      Eth1/20          5     1  Up         Active      46
Veth32771 VIF-18      Eth1/20          8     1  Up         Active      49
Veth32772 VIF-19      Eth1/20          9     1  Up         Active      50
Veth32773 VIF-20      Eth1/20         11     1  Up         Active      52
Veth32774 VIF-21      Eth1/20         12     1  Up         Active      53
Veth32775 VIF-22      Eth1/20         13     1  Up         Active      54
Veth32776 VIF-23      Eth1/20         14     1  Up         Active      55
Veth32777 VIF-24      Eth1/20         15     1  Up         Active      56
Total 9 Veth interfaces

switch-1# show interface virtual summary bound interface ethernet 1/20

Veth       Bound        Channel/  Port        Mac               VM
Interface  Interface    DV-Port   Profile     Address           Name
---------------------------------------------------------------------------
Veth32769  Eth1/20          1    sample
Veth32770  Eth1/20          5    sample
Veth32771  Eth1/20          8    sample
Veth32772  Eth1/20          9    sample
Veth32773  Eth1/20         11    sample
Veth32774  Eth1/20         12    sample
Veth32775  Eth1/20         13    sample
```

```
Veth32776  Eth1/20        14    sample
Veth32777  Eth1/20        15    sample
Total 9 Veth interfaces

switch-1#
```

Verifying the Connection to the vCenter Server

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	**show svs connections** [*svs-name*]	Displays the current SVS connections.

This example shows how to display the details of the SVS connection:

```
switch-1# configure terminal
switch-1(config)# show svs connections

Local Info:
-----------
connection 2VC:
    ip address: 192.0.20.125
    remote port: 80
    vrf: management
    protocol: vmware-vim https
    certificate: default
    datacenter name: DC1
    extension key: Cisco_Nexus_6004_1945593678
    dvs name: Pod1
    DVS uuid: cd 05 25 50 6d a9 a5 c4-eb 9c 8f 6b fa 51 b1 aa
    config status: Enabled
    operational status: Connected
    sync status: in progress
    version: VMware vCenter Server 6.0.2 build-388657


Peer Info:
----------
    hostname: -
    ip address: -
    vrf:
    protocol: -
    extension key: Cisco_Nexus_6004_1945593678
    certificate: default
      certificate match: TRUE
    datacenter name: DC1
    dvs name: Pod1
    DVS uuid: cd 05 25 50 6d a9 a5 c4-eb 9c 8f 6b fa 51 b1 aa
    config status: Disabled
    operational status: Connected

switch-1(config)#
```