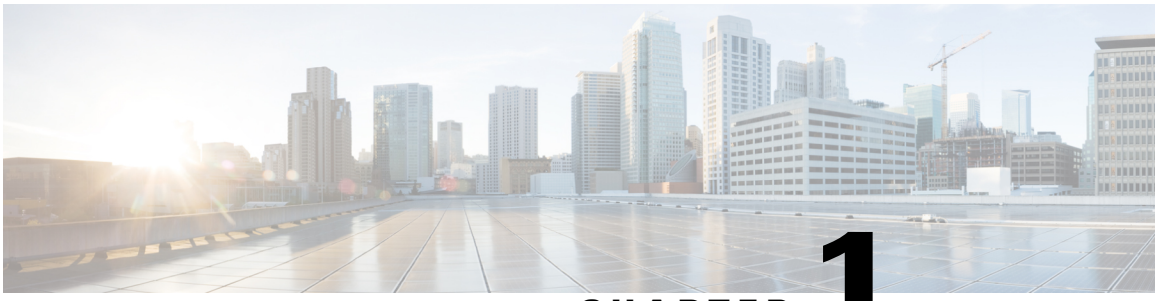
Cisco Intercloud Fabric Configuration Guide, Release 2.3.1

First Published: November 13, 2015

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

Overview

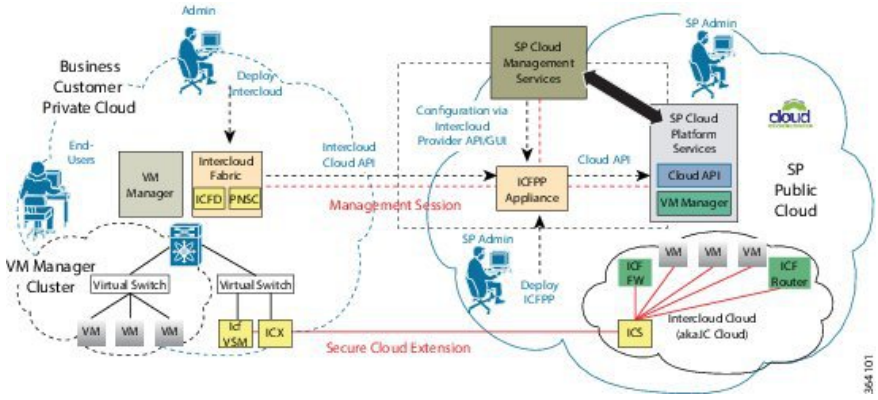
About Cisco Intercloud Fabric

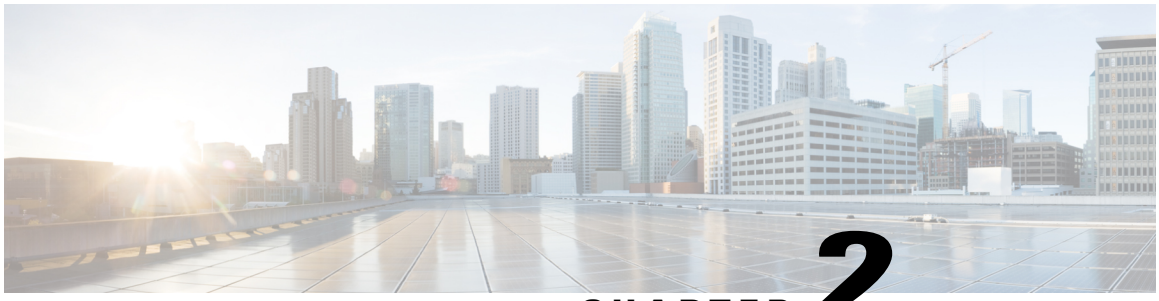
Cisco Intercloud Fabric provides a faster and flexible response to business needs and addresses the potential challenges with hybrid clouds. A hybrid cloud is an interaction between private and provider clouds where private clouds extend to provider clouds and use provider cloud resources in a secure and scalable way. Cisco Intercloud Fabric lets you place workloads across heterogeneous environments in multiple provider clouds. Cisco Intercloud Fabric provides the architectural foundation for secure hybrid clouds, which allows enterprises to easily and securely connect the private clouds to the provider cloud as needed and on demand. With a hybrid cloud, enterprises can combine the benefits of private and provider clouds. Cisco Intercloud Fabric provides the following benefits:

- Provides a single point of management and control for virtual workloads across multiple provider clouds.
- Provides a choice of cloud providers, such as Amazon Web Service, Microsoft Azure, and Cisco Intercloud Services – V.
- Provides highly secure, scalable connectivity to extend private clouds to service provider clouds.
- Enforces consistent network and workload policies throughout the hybrid cloud.

- Enables workload mobility to and from service provider clouds for virtual workloads.

Figure 1: Cisco Intercloud Fabric





CHAPTER 2

Configuring ERSPAN

This chapter contains the following sections:

Information About Encapsulated Remote SPAN

Encapsulated remote SPAN (ERSPAN) monitors traffic in multiple network devices across an IP network and sends that traffic in an encapsulated envelope to destination analyzers. ERSPAN can be used to monitor traffic remotely. ERSPAN sources can be ports, VLANs, or port profiles.

ERSPAN Sources

The interfaces from which traffic can be monitored are called ERSPAN sources. These sources include Ethernet, virtual Ethernet, port profile, and VLAN. When a VLAN is specified as an ERSPAN source, all supported interfaces in the VLAN are ERSPAN sources. When a port profile is specified as an ERSPAN source, all ports that inherit the port profile are ERSPAN sources. Traffic can be monitored in the receive direction, the transmit direction, or both directions for Ethernet and virtual Ethernet source interfaces as described by the following:

- Receive source (Rx)—Traffic that enters the switch through this source port is copied to the ERSPAN destination port.
- Transmit source (Tx)—Traffic that exits the switch through this source port is copied to the ERSPAN destination port.

Characteristics of ERSPAN Sources

An ERSPAN source has these characteristics:

- Can be port type Ethernet, virtual Ethernet, port channel, port profile, or VLAN.
- Cannot be a destination port or port profile.
- Can be configured to monitor the direction of traffic—receive, transmit, or both.
- Can be in the same or different VLANs.
- For VLAN ERSPAN sources, all active ports in the source VLAN are included as source ports.
- For port profile sources, all active interfaces attached to the port profile are included as source ports.

ERSPAN Destinations

An ERSPAN destination is an IP address on a remote device.

Characteristics of ERSPAN Destinations

- An ERSPAN destination is specified by an IP address.
- In ERSPAN, the source SPAN interface and destination SPAN interface can be on different devices interconnected by an IP network. ERSPAN traffic uses generic routing encapsulation (GRE).

Network Analysis Module

You can also use the Cisco Network Analysis Module (NAM) to monitor ERSPAN data sources for application performance, traffic analysis, and packet header analysis.

ERSPAN Sessions

You can create up to 64 total ERSPAN sessions on the Virtual Ethernet Module (VEM).

You must configure an ERSPAN session ID that is added to the ERSPAN header of the encapsulated frame to differentiate between ERSPAN streams of traffic at the termination box. You can also configure the range of flow ID numbers.

Guidelines and Limitations for ERSPAN

- ERSPAN is supported only on Intercloud Fabric Switch (ICS) (no ERSPAN sources on Intercloud Fabric Extender (ICX)).
- A maximum of 64 ERSPAN sessions can be configured on the Virtual Supervisor Module (VSM).
- A maximum of 32 source VLANs are allowed in a session.

- A maximum of 16 source port profiles are allowed in a session.
- A maximum of 128 source interfaces are allowed in a session.

Caution Overload Potential

To avoid an overload on uplink ports, use caution when configuring ERSPAN, especially when sourcing VLANs. The uplink that the VM kernel uses might get overloaded due to ERSPAN traffic. VSM-VEM communication might also be impacted. For example, when the Nexus 1000V is configured for Layer 3 connectivity, both AIPC traffic and ERSPAN traffic use the same VM kernel NIC.

- A port can be configured in a maximum of four ERSPAN sessions.
- A port can be a source in a maximum of four ERSPAN sessions.

Configuring ERSPAN

This section describes how to configure ERSPAN and includes the following procedures:

- Configuring an ERSPAN Port Profile
- Configuring an ERSPAN Session

Configuring an ERSPAN Port Profile

You can configure a port profile on the VSM to carry ERSPAN packets through the IP network to a remote destination analyzer.

You must complete this configuration for all hosts in the OpenStack Horizon server.

This procedure includes steps to configure the port profile for the following requirements:

- ERSPAN for Layer 3 control.
- An access port profile. It cannot be a trunk port profile.

Only one ERSPAN local Layer 3 interface can be assigned to this Layer 3 control port profile per host as follows:

- If more than one ERSPAN local Layer 3 interface is assigned to a host, the first one assigned takes effect. The second one is not considered a Layer 3 interface.
- If more than one ERSPAN local Layer 3 interface is assigned to a host, and you remove the second assigned one, the VEM does not use the first assigned one. Instead, you must remove both the ERSPAN local Layer 3 interfaces and then add one back.

Before You Begin

- Log in to the CLI in EXEC mode.
- Ensure that a name has been established for this port profile.

Note The port profile name is used to configure the ERSPAN local Layer 3 interface. An ERSPAN local Layer 3 interface is required on each KVM host to send ERSPAN-encapsulated IP packets, and must have IP connectivity to the ERSPAN destination IP address.

- Ensure that a name has been established for the OpenStack policy profile to which this profile maps. For information, see the *Cisco Nexus 1000V for KVM Virtual Network Configuration Guide*.
- Create the system VLAN that sends IP traffic to the ERSPAN destination and note the VLAN ID to use in this configuration.
- Obtain the documentation for adding a new virtual adapter.

For more information about system port profiles, see the *Cisco Nexus 1000V Port Profile Configuration Guide*.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **port-profile** *port_profile_name*	Creates the port profile and places you in global configuration mode for the specified port profile. This command saves the port profile in the running configuration. The port profile name can be up to 80 characters and must be unique for each port profile on the Cisco Nexus 1000V.
Step 3	switch(config-prot-prof)# **capability l3control**	Configures the port profile to carry ERSPAN traffic and saves the port profile in the running configuration.
Step 4	switch(config-prot-prof)# **publish port-profile***name*	Designates the port profile as an OpenStack policy profile and adds the name of the OpenStack policy profile to which this profile maps. This command saves the settings in the running configuration. The port profile is mapped to an OpenStack policy profile of the same name. When an OpenStack Horizon server connection is established, the port group created in Cisco Nexus 1000V is then distributed to the virtual switch on the OpenStack Horizon server. The *name* argument is the same as the port profile name if you do not specify a port group name. If you want to map the port profile to a different port group name, use the name option followed by the alternate name.
Step 5	switch(config-prot-prof)# **switchport mode access**	Designates the interfaces as switch access ports (the default).
Step 6	switch(config-prot-prof)# **switchport access vlan** *id*	Assigns a VLAN ID to the access port for this port profile and saves the setting in the running configuration.

	Command or Action	Purpose
		This VLAN is used to send IP traffic to the ERSPAN destination.
Step 7	switch(config-prot-prof)# **no shutdown**	Enables the interface in the running configuration.
Step 8	switch(config-prot-prof)# **state enabled**	Enables the port profile in the running configuration. This port profile is now ready to send out ERSPAN packets on all KVM hosts with ERSPAN sources.
Step 9	switch(config-prot-prof)# **show port-profile name** *port_profile_name*	(Optional) Displays the configuration for the specified port profile as it exists in the running configuration.
Step 10	switch(config-port-prof)# **copy running-config startup-config**	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.
Step 11	To configure the ERSPAN local Layer 3 interface, navigate to the `/etc/n1kv/n1kv.conf` file and enter the details such as the port name, port profile, IP address, subnet, and the MAC address. For example, virt erspan0 profile erspan-pp mode static address 30.30.30.20 netmask 255.255.255.0 mac 00:22:44:34:ab:cd.	

```
switch# configure terminal
switch(config)# port-profile erspan_profile
switch(config-port-prof)# capability l3control
switch(config-port-prof)# publish port-profile
switch(config-port-prof)# switchport mode access
switch(config-port-prof)# switchport access vlan 2
switch(config-port-prof)# no shutdown
switch(config-port-prof)# state enabled
switch(config-port-prof)# show port-profile name erspan
port-profile erspan
  description:
  status: enabled
  capability uplink: no
  capability l3control: yes
  system vlans: 2
  port-group: access
  max-ports: 32
  inherit:
  config attributes:
    switchport access vlan 2
    no shutdown
  evaluated config attributes:
    switchport access vlan 2
    no shutdown
  assigned interfaces:
n1000v(config-port-prof)# copy running-config startup-config
```

Configuring an ERSPAN Session

This procedure involves creating the SPAN session in ERSPAN source configuration mode (config-erspan-source).

SPAN sessions are created in the shut state by default.

When you create a SPAN session that already exists, any additional configuration is added to that session. To make sure the session is cleared of any previous configuration, you can delete the session first. The step to do this is included in the procedure.

Before You Begin

- Log in to the CLI in EXEC mode.
- Obtain the number of the SPAN session that you are going to configure.
- Configure an ERSPAN-capable port profile on the VSM.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **no monitor session** *session-number*	Clears the specified session.
Step 3	switch(config)# **monitor session** *session-number* **type erspan-source**	Creates a session with the given session number and places you in ERSPAN source configuration mode. This configuration is saved in the running configuration.
Step 4	switch(config-erspan-src)# **description** *description*	For the specified ERSPAN session, adds a description and saves it in the running configuration. The *description* can be up to 32 alphanumeric characters. The default is blank (no description).
Step 5	switch(config-erspan-src)#**source** {**interface** *type* {*number*| *range*} | **vlan** {*number* | *range*} | **port-profile** {*name*}} [**rx** | **tx** | **both**]	For the specified session, configures the sources and the direction of traffic to monitor and saves them in the running configuration. • For the *type* argument, specify the interface type—ethernet, port-channel, vethernet. • For the *number* argument, specify the interface slot/port or range; or the VLAN number or range to monitor. • For the *name* argument, specify the name of the existing port profile. • For the traffic direction keywords, specify as follows: ◦ **rx** (the VLAN default) indicates receive. ◦ **tx** indicates transmit.

	Command or Action	Purpose
		◦ **both** is the default keyword.
Step 6	Repeat Step 5 to configure additional ERSPAN sources.	(Optional)
Step 7	switch(config-erspan-src)# **filter vlan** {*number* | *range*}	(Optional) For the specified ERSPAN session, configures the VLANs, VLAN lists, or VLAN ranges to be monitored; and saves the VLAN arguments to the running configuration. On the monitor port, only the traffic from the VLANs that match the VLAN filter list is replicated to the destination.
Step 8	Repeat Step 7 to configure all source VLANs to filter.	(Optional)
Step 9	switch(config-erspan-src)# **destination ip** *ip_address*	Configures the IP address of the host to which the encapsulated traffic is sent in this monitor session and saves it in the running configuration.
Step 10	switch(config-erspan-src)# **ip ttl** *ttl_value*	(Optional) Specifies the IP time-to-live value, from 1 to 255, for ERSPAN packets in this monitor session and saves it in the running configuration.
Step 11	switch(config-erspan-src)# **mtu** *mtu_value*	(Optional) Specifies an MTU size (from 50 to 1500) for ERSPAN packets in this monitor session and saves it in the running configuration. The 1500 MTU size limit includes a 50 byte overhead added to monitored packets by ERSPAN. Packets larger than this size are truncated. The default is 1500. **Note** If the ERSPAN destination is a Cisco 6500 switch, truncated ERSPAN packets are dropped unless the **no mls verify ip length consistent** command is configured on the Cisco 6500.
Step 12	switch(config-erspan-src)# **header-type** *value*	Specifies the ERSPAN header type (2 or 3) used for ERSPAN encapsulation for this monitor session as follows: • 2 is the ERPSPANv2 header type (the default). • 3 is the ERSPANv3 header type. (Used with NAM setups. Any other type of destination works only with the default v2 headers.)
Step 13	switch(config-erspan-src)# **erspan-id** *flow_id*	Adds an ERSPAN ID (from 1 to 1023) to the session configuration and saves it in the running configuration.

	Command or Action	Purpose
		The session ERSPAN ID is added to the ERSPAN header of the encapsulated frame and can be used at the termination box to differentiate between various ERSPAN streams of traffic.
Step 14	switch(config-erspan-src)# **no shut**	Enables the ERSPAN session and saves it in the running configuration. By default, the session is created in the shut state.
Step 15	switch(config-erspan-src)# **show monitor session** *session_id*	(Optional) Displays the ERSPAN session configuration as it exists in the running configuration.
Step 16	switch(config-erspan-src)# **copy running-config startup-config**	(Optional) Saves the running configuration persistently through reboots and restarts by copying it to the startup configuration.

```
switch# configure terminal
switch(config)# no monitor session 3
switch(config)# monitor session 3 type erspan
switch(config-erspan-src)# description my_erspan_session_3
switch(config-erspan-src)# source interface ethernet 2/1-3, ethernet 3/1 rx
switch(config-erspan-src)# filter vlan 3-5, 7
switch(config-erspan-src)# destination ip 10.54.54.1
switch(config-erspan-src)# ip ttl 64
switch(config-erspan-src)# mtu 1000
switch(config-erspan-src)# header-type 2
switch(config-erspan-src)# erspan-id 51
switch(config-erspan-src)# no shut
switch(config-erspan-src)# show monitor session 3
switch(config-erspan-src)# copy running-config startup-config
```

Configuring the Allowable ERSPAN Flow IDs

Use this procedure to restrict the allowable range of available flow IDs that can be assigned to ERSPAN sessions.

The available ERSPAN flow IDs are from 1 to 1023.

Before You Begin

- Log in to the CLI in EXEC mode.
- Determine the restricted range of ERSPAN flow IDs that you want to designate.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.

	Command or Action	Purpose
Step 2	switch(config)# [**no**] **limit-resource erspan-flow-id minimum** *min_val* **maximum** *max_va*l	Restricts the allowable range of ERSPAN flow IDs that can be assigned. The allowable range is from 1 to 1023. The defaults are as follows: • The minimum value = 1 • The maximum value = 1023 The **no** form of this command removes any configured values and restores default values.
Step 3	switch(config)# **show running monitor**	(Optional) Displays changes to the default limit-resource erspan-flow-id values for verification.
Step 4	switch(config)# **copy running-config startup-config**	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

```
switch# configure terminal
switch(config)# limit-resource erspan-flow-id minimum 20 maximum 40
switch(config)# show monitor
switch(config)# show running monitor
switch(config)# copy running-config startup-config
```

Configuration Example for an ERSPAN Session

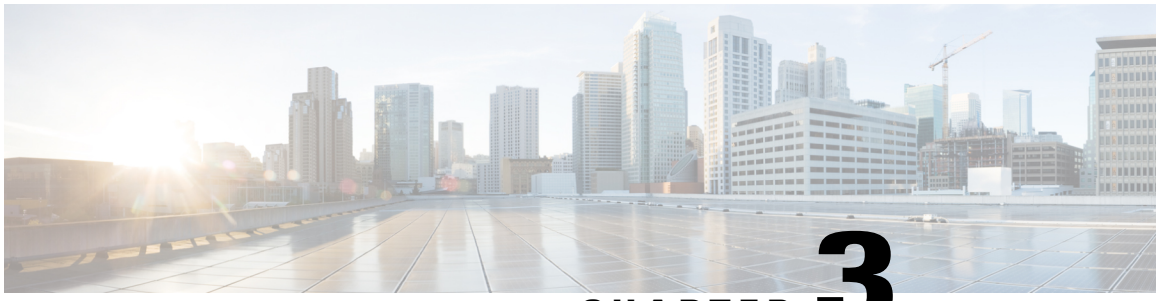
The following example shows how to create an ERSPAN session for a source Ethernet interface and destination IP address on the Cisco Nexus 1000V. Packets arriving at the destination IP are identified by the ID 999 in their header.

```
switch# monitor session 2 type erspan-source
switch(config-erspan-src)# source interface ethernet 3/3
switch(config-erspan-src)# source port-profile my_profile_src
switch(config-erspan-src)# destination ip 10.54.54.1
switch(config-erspan-src)# erspan-id 999
switch(config-erspan-src)# mtu 1000
switch(config-erspan-src)# no shut

switch(config-erspan-src)# show monitor session 2
   session 2
---------------
type              : erspan-source
state             : up
source intf       :
    rx            : Eth3/3
    tx            : Eth3/3
    both          : Eth3/3
source VLANs      :
    rx            :
    tx            :
    both          :
```

```
source port-profile :
    rx            : my_profile_src
    tx            : my_profile_src
    both          : my_profile_src
filter VLANs      : filter not specified
destination IP    : 10.54.54.1
ERSPAN ID         : 999
ERSPAN TTL        : 64
ERSPAN IP Prec.   : 0
ERSPAN DSCP       : 0
ERSPAN MTU        : 1000
ERSPAN Header Type: 2

switch(config-erspan-src)# module vem 3 execute vemcmd show span

VEM SOURCE IP: 10.54.54.10

HW SSN ID   ERSPAN ID   HDR VER   DST LTL/IP
        1                 local   49,51,52,55,56
        2         999         2   10.54.54.1
```

Feature History for ERSPAN

Feature Name	Releases	Feature Information
ERSPAN	5.2(1)SK3(2.1)	ERSPAN was introduced.



CHAPTER 3

Configuring NetFlow

This chapter contains the following sections:

Information About NetFlow

NetFlow allows you to evaluate IP and Ethernet traffic and understand how and where it flows. NetFlow gives you visibility into traffic that transits the virtual switch by characterizing traffic based on its source, destination, timing, and application information. You can use this information to assess network availability and performance, assist in meeting regulatory requirements (compliance), and help with troubleshooting. NetFlow gathers data that you can use for accounting, network monitoring, and network planning.

What Is a Flow

You create a flow using a flow record to define the criteria for your flow. All criteria must match for the packet to count in the given flow. Flows are stored in the NetFlow cache. Flow information tells you the following:

- Source address tells you who is originating the traffic.
- Destination address tells who is receiving the traffic.
- Ports characterize the application that uses the traffic.

- Class of service examines the priority of the traffic.
- The device interface tells how traffic is being used by the network device.
- Tallied packets and bytes show the amount of traffic.

Flow Record Definition

A flow record defines the information that NetFlow gathers, such as the packets in the flow and the types of counters gathered per flow. You can define new flow records or use the predefined Cisco Nexus 1000V flow record.

Predefined flow records use 32-bit counters and are not recommended for data rates above 1 Gbps. For data rates that are higher than 1 Gbps, Cisco recommends that you manually configure the records to use 64-bit counters.

The following table describes the criteria defined in a flow record.

Table 1: Flow Record Criteria

Flow Record Criteria	Description
Match	Defines the information that is matched for collection in the flow record. • ip—Data collected in the flow record matches one of the following IP options: ◦ Protocol ◦ tos (type of service) • IPv4—Data collected in the flow record matches one of the following IPv4 address options: ◦ Source address ◦ Destination address • Transport—Data collected in the flow record matches one of the following transport options: ◦ Destination port ◦ Source port • datalink—Data collected in the flow record matches one of the following data link options: ◦ mac source-address ◦ mac destination-address ◦ ethertype ◦ vlan ◦ vxlan **Note** Layer 2 fields can be matched only when IP fields are not present in the record.

Flow Record Criteria	Description
Collect	Defines how the flow record collects information. • Counter—Collects flow record information in one of the following formats: ◦ Bytes—32-bit counter (default). ◦ Bytes long—64-bit counter (recommended for data rates that are higher than 1 Gbps). ◦ Packets—32-bit counter (default). ◦ Packets long—64-bit counters (recommended for data rates that are higher than 1 Gbps). • timestamp sys-uptime—Collects the system uptime for the first or last packet in the flow. • transport tcp flags—Collects the TCP transport layer flags for the packets in the flow. **Note** 64-bit counters are recommended.

Predefined Flow Records

Cisco Nexus 1000V Predefined Flow Record: Netflow-Original

```
switch# show flow record netflow-original
Flow record netflow-original:
    Description: Traditional IPv4 input NetFlow with origin ASs
    No. of users: 0
    Template ID: 0
    Fields:
        match ipv4 source address
        match ipv4 destination address
        match ip protocol
        match ip tos
        match transport source-port
        match transport destination-port
        match interface input
        match interface output
        match flow direction
        collect routing source as
        collect routing destination as
        collect routing next-hop address ipv4
        collect transport tcp flags
        collect counter bytes long
        collect counter packets long
        collect timestamp sys-uptime first
        collect timestamp sys-uptime last
switch#
```

Note Although the following lines appear in the output of the **show flow record** command, the commands they are based on are not currently supported in Cisco Nexus 1000V. The use of these commands has no affect on the configuration.

```
collect routing source as
collect routing destination as
collect routing next-hop address ipv4
```

Cisco Nexus 1000V Predefined Flow Record: Netflow IPv4 Original-Input

```
switch# show flow record netflow ipv4 original-input
Flow record netflow ipv4 original-input:
    Description: Traditional IPv4 input NetFlow
    No. of users: 0
    Template ID: 0
    Fields:
        match ipv4 source address
        match ipv4 destination address
        match ip protocol
        match ip tos
        match transport source-port
        match transport destination-port
        match interface input
        match interface output
        match flow direction
        collect routing source as
        collect routing destination as
        collect routing next-hop address ipv4
        collect transport tcp flags
        collect counter bytes long
        collect counter packets long
        collect timestamp sys-uptime first
        collect timestamp sys-uptime last
switch#
```

Cisco Nexus 1000V Predefined Flow Record: Netflow IPv4 Original-Output

```
switch# show flow record netflow ipv4 original-output
Flow record netflow ipv4 original-output:
    Description: Traditional IPv4 output NetFlow
    No. of users: 0
    Template ID: 0
    Fields:
        match ipv4 source address
        match ipv4 destination address
        match ip protocol
        match ip tos
        match transport source-port
        match transport destination-port
        match interface input
        match interface output
        match flow direction
        collect routing source as
        collect routing destination as
        collect routing next-hop address ipv4
        collect transport tcp flags
        collect counter bytes
        collect counter packets
        collect timestamp sys-uptime first
        collect timestamp sys-uptime last
switch#
```

Cisco Nexus 1000V Predefined Flow Record: Netflow Protocol-Port

```
switch# show flow record netflow protocol-port
Flow record netflow protocol-port:
```

```
    Description: Protocol and Ports aggregation scheme
    No. of users: 0
    Template ID: 0
    Fields:
        match ip protocol
        match transport source-port
        match transport destination-port
        match interface input
        match interface output
        match flow direction
        collect counter bytes
        collect counter packets
        collect timestamp sys-uptime first
        collect timestamp sys-uptime last
switch#
```

Accessing NetFlow Data

You can use two methods to access NetFlow data:

- Command-line interface (CLI)
- NetFlow collector (a separate product from the Cisco Nexus 1000V for KVM)

CLI for NetFlow

You can use the CLI to access NetFlow data and to view what is happening in your network.

The CLI uses a flow monitor and a flow exporter to capture and export flow records to the NetFlow collector. Cisco Nexus 1000V supports the NetFlow Version 9 export format.

Note The Cisco Nexus 1000V supports UDP as the transport protocol for exporting data, up to two exporters per monitor.

Flow Monitor

A flow monitor creates an association between the following NetFlow components:

- Flow record—Consists of matching and collection criteria
- Flow exporter—Consists of the export criteria

This flow monitor enables a set, which consists of a record and an exporter. You can define this set once and reuse it multiple times. You can create multiple flow monitors for different needs. A flow monitor is applied to a specific interface or port profile in a specific direction.

Flow Exporter

The flow exporter is used to define the source and destination of the flow records. The source is from the VEM module and the destination is the reporting server, called the Netflow Collector. An IP packet is sent from the source to the destination with the collected information. The packet originates from the VEM, but

you can configure which IP address is placed in the source field of the IP packet. The destination requires an IP address as well as a UDP port number for which the Netflow collector listens for packets.

An exporter definition includes the following:

- Destination IP address
- UDP port number (where the collector is listening)
- Source IP address to spoof (not the actual source location, but the address placed in the IP packet sent to the collector)
- Export format version

NetFlow Collector

The NetFlow data reporting process is as follows:

1. You configure NetFlow records to define the information that NetFlow gathers.
2. You configure Netflow monitor to capture flow records to the NetFlow cache.
3. You configure NetFlow export to send flows to the collector.
4. The Cisco Nexus 1000V searches the NetFlow cache for flows that have expired and exports them to the NetFlow collector server.
5. Flows are bundled together based on space availability in the UDP export packet and based on an export timer.
6. The NetFlow collector software creates real-time or historical reports from the data.

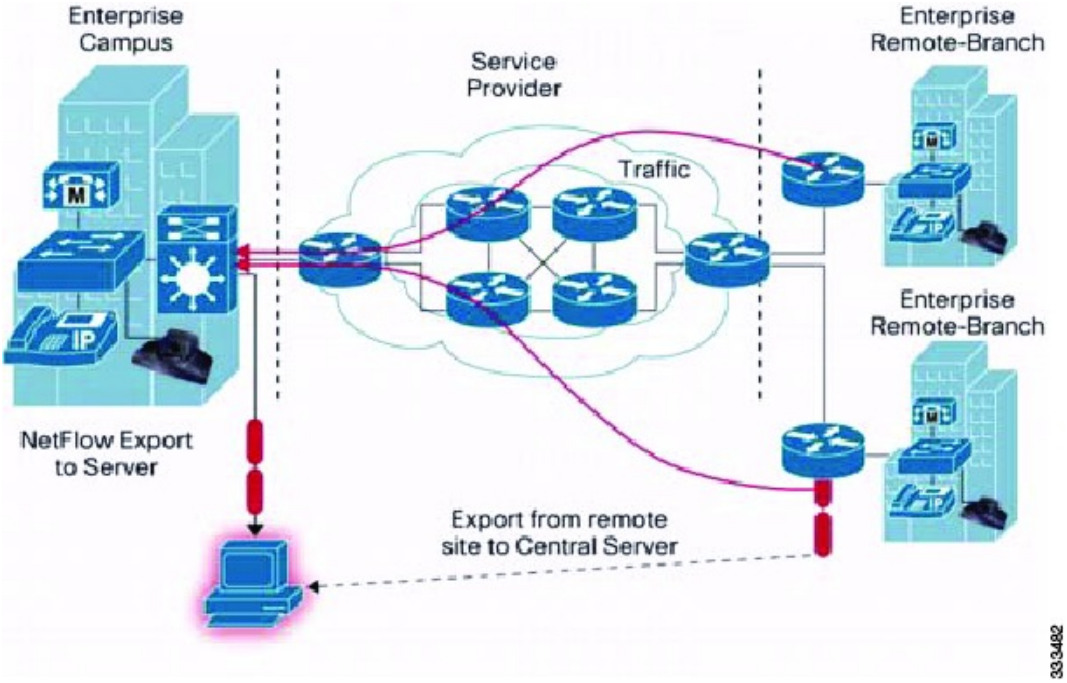
Exporting Flows to the NetFlow Collector Server

Timers determine when a flow is exported to the NetFlow collector server. See the following figure where a flow is ready for export when one of the following occurs:

- The flow is inactive for a certain amount of time, during which no new packets are received for the flow.

- The flow has lived longer than the active timer, such as a long FTP download.

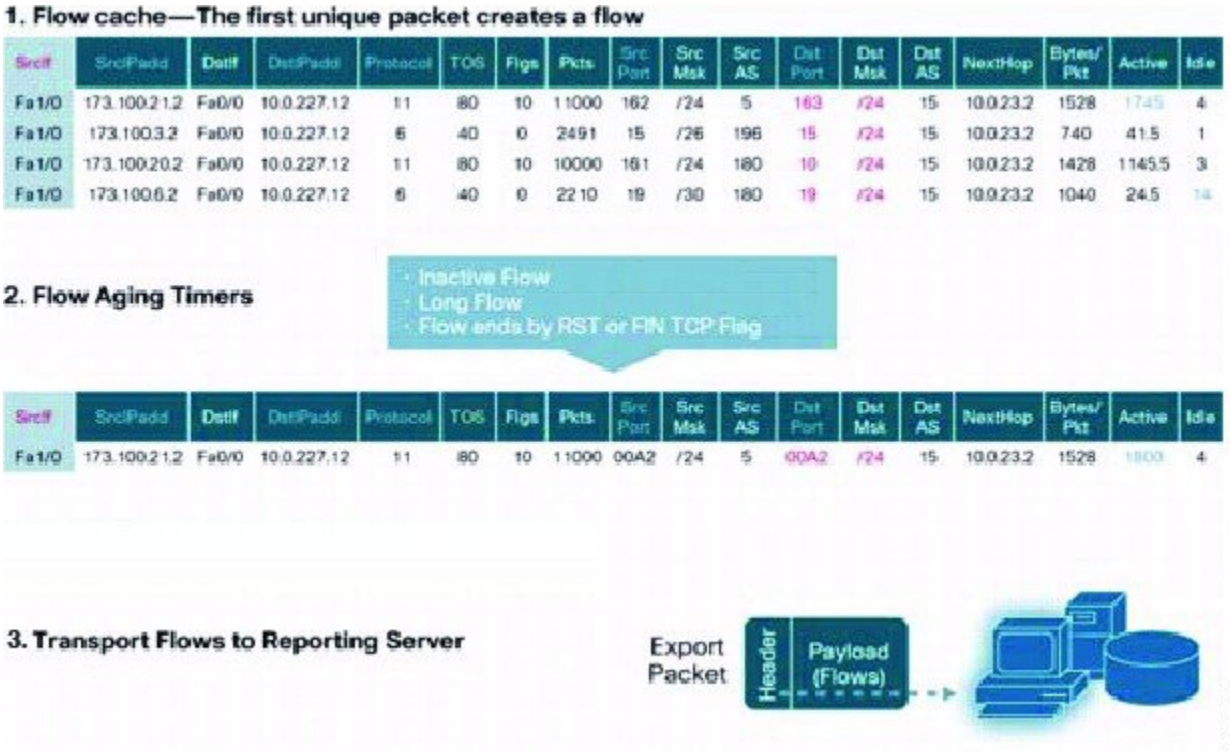
Figure 2: Exporting Flows to the NetFlow Collector Server



What NetFlow Data Looks Like

The following figure shows an example of NetFlow data.

Figure 3: NetFlow Cache Example



High Availability for NetFlow

The Cisco Nexus 1000V supports stateful restarts for NetFlow. After a reboot or supervisor switchover, the Cisco Nexus 1000V applies the running configuration.

Guidelines and Limitations for NetFlow

- In Cisco Nexus 1000V, the mgmt0 interface IP address of the VSM is configured by default as the source IP address for an exporter.
- Predefined flow records use 32-bit counters, which are recommended for data rates above 1 Gbps. For data rates that are higher than 1 Gbps, Cisco recommends that you manually configure the records to use 64-bit counters.
- The Cisco Nexus 1000V includes the following predefined flow records:
  - netflow-original—The Cisco Nexus 1000V predefined traditional IPv4 input NetFlow with origin ASs

Note The routing-related fields in this predefined flow record are ignored.

- netflow ipv4 original-input—The Cisco Nexus 1000V predefined traditional IPv4 input NetFlow
- netflow ipv4 original-output—The Cisco Nexus 1000V predefined traditional IPv4 output NetFlow
- netflow protocol-port—The Cisco Nexus 1000V predefined protocol and ports aggregation scheme

- Up to 8,000 NetFlow instances are allowed per Distributed Virtual Switch (DVS).
- Up to 300 NetFlow instances are allowed per host.
- A maximum of one flow monitor per interface per direction is allowed.
- Up to two flow exporters are allowed per monitor.
- Up to 64 NetFlow monitors, exporters, or records are allowed per DVS.
- NetFlow is not supported on port channels or interfaces in a port channel.

Default Settings for NetFlow

Table 2: Default NetFlow Parameters

Parameters	Default
NetFlow version	9
Source	Line card export with spoofed mgmt0 IP address of the VSM
Match	Direction and interface (incoming/outgoing)
Flow monitor active timeout[1]	1800
Flow monitor inactive timeout [2]	45
DSCP	Default/best-effort (0)
VRF	Management (1)

1 Cisco recommends that the difference between the flow active timeout and the flow inactive timeout be a minimum of 1600 seconds.
2 Cisco recommends that the difference between the flow active timeout and the flow inactive timeout be a minimum of 1600 seconds.

Enabling the NetFlow Feature

Before You Begin

You are logged in to the CLI in EXEC mode.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **feature netflow**	Enables the NetFlow feature.
Step 3	switch(config)# **show feature**	(Optional) Displays the available features and whether or not they are enabled.
Step 4	switch(config)# **copy running-config startup-config**	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

This example shows how to enable the NetFlow feature:

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

Configuring Netflow

Defining a Flow Record

Before You Begin

- You know which of the options you want this flow record to match.
- You know which options you want this flow record to collect.

Note Although the following lines appear in the output of the **show flow record** command, the commands they are based on are not currently supported in Cisco Nexus 1000V. The use of these commands has no effect on the configuration.

```
collect routing source as
collect routing destination as
collect routing next-hop address ipv4
```

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **flow record** *name*	Creates a flow record by name, and places you in the CLI Flow Record Configuration mode for that specific record.
Step 3	switch(config-flow-record)# **description** *string*	(Optional) Adds a description of up to 63 characters to the flow record and saves it to the running configuration.
Step 4	switch(config-flow-record)# **match** {**ip** {**protocol** | **tos**} | **ipv4** {**destination** | **source**} | **transport** {*destination-port* | *source-port*} | **datalink** {{**mac** {*source-address* | *destination-address*}} | **ethertype** | **vlan** | **vxlan** }}	Defines the flow record to match one of the following and saves it in the running configuration. • ip—Matches one of the following IP options: ◦ protocol ◦ tos (type of service) • ipv4—Matches one of the following IPv4 address options: ◦ source address ◦ destination address • transport—Matches one of the following transport options: ◦ destination port ◦ source port • datalink—Data collected in the flow record matches one of the following data link options: ◦ mac source-address ◦ mac destination-address ◦ ethertype ◦ vlan ◦ vxlan **Note** NetFlow does not support mixing data link fields with other field types in the same record.
Step 5	switch(config-flow-record)# **collect** {**counter** {**bytes** [**long**] | **packets** [**long**]} | **timestamp** **sys-uptime**{*first* | *last*} | **transport** **tcp flags**}	Specifies a collection option to define the information to collect in the flow record and saves it in the running configuration. • counter—Collects flow record information in one of the following formats:

	Command or Action	Purpose
		◦ bytes: collected in 32-bit counters unless the long 64-bit counter is specified. ◦ packets: collected in 32-bit counters unless the long 64-bit counter is specified. **Note** Cisco recommends that the 64-bit counters be used for systems with data rates in excess of 1 Gbps. • timestamp sys-uptime—Collects the system uptime for the first or last packet in the flow. • transport tcp flags—Collects the TCP transport layer flags for the packets in the flow.
Step 6	switch(config-flow-record)# **show flow record** [**name**]	(Optional) Displays information about flow records.
Step 7	switch(config-flow-record)# **copy running-config startup-config**	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

The following example shows how to create a flow record:

```
switch# configure terminal
switch(config)# flow record RecordTest
switch(config-flow-record)# description Ipv4flow
switch(config-flow-record)# match ipv4 destination address
switch(config-flow-record)# collect counter packets
switch(config-flow-record)# show flow record RecordTest
Flow record RecordTest:
    Description: Ipv4flow
    No. of users: 0
    Template ID: 0
    Fields:
        match ipv4 destination address
        match interface input
        match interface output
        match flow direction
        collect counter packets
switch(config-flow-record)#
```

Defining a Flow Exporter

A flow exporter defines where and how flow records are exported to the NetFlow collector server.

- Export format version 9 is supported.
- A maximum of two flow exporters per monitor are permitted.

Before You Begin

- You know the destination IP address of the NetFlow collector server.

- You know the transport UDP port that the collector is listening on.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)#**flow exporter** *name*	Creates a flow exporter, saves it in the running configuration, and places you in CLI Flow Exporter Configuration mode.
Step 3	switch(config-flow-exporter)# **description** *string*	Adds a description of up to 63 characters to this flow exporter and saves it in the running configuration.
Step 4	switch(config-flow-exporter)# **destination** *ipv4-address*	Specifies the IP address of the destination interface for this flow exporter and saves it in the running configuration.
Step 5	switch(config-flow-exporter)# **dscp** *value*	Specifies the differentiated services codepoint value for this flow exporter, between 0 and 63, and saves it in the running configuration.
Step 6	switch(config-flow-exporter)# **source lc-exp** *ipv4-address*/*subnet-mask*	(Optional) Specifies the IP address to spoof, from which the flow records are sent to the NetFlow collector server, and saves it in the running configuration.
Step 7	switch(config-flow-exporter)# **transport udp** *port-number*	Specifies the destination UDP port, between 1 and 65535, used to reach the NetFlow collector, and saves it in the running configuration.
Step 8	switch(config-flow-exporter)# **version** {**9**}	Specifies NetFlow export version 9, saves it in the running configuration, and places you in the export version 9 configuration mode.
Step 9	switch(config-flow-exporter-version-9)# **option** {**exporter-stats** | **interface-table**} **timeout** *value*	Specifies one of the following version 9 exporter resend timers and its value, between 1 and 86400 seconds, and saves it in the running configuration: • exporter-stats • interface-table
Step 10	switch(config-flow-exporter-version-9)# **template data timeout** *seconds*	Sets the template data resend timer and its value, between 1 and 86400 seconds, and saves it in the running configuration.
Step 11	switch(config-flow-exporter-version-9)# **show flow exporter** [*name*]	(Optional) Displays information about the flow exporter.

	Command or Action	Purpose
Step 12	switch(config-flow-exporter-version-9)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

The following example shows how to create a flow exporter:

```
switch# configure terminal
switch(config)# flow exporter ExportTest
switch(config-flow-exporter)# description ExportHamilton
switch(config-flow-exporter)# destination 192.0.2.1
switch(config-flow-exporter)# dscp 2
switch(config-flow-exporter)# source lc-exp 192.0.2.2/24
switch(config-flow-exporter)# transport udp 200
switch(config-flow-exporter)# version 9
switch(config-flow-exporter-version-9)# option exporter-stats timeout 1200
switch(config-flow-exporter-version-9)# template data timeout 1200
switch(config-flow-exporter-version-9)# show flow exporter ExportTest
Flow exporter ExportTest:
    Description: ExportHamilton
    Destination: 192.0.2.1
    VRF: management (1)
    Destination UDP Port 200
    Source IP Address 192.0.2.2
    Export from Line Card
    DSCP 2
    Export Version 9
        Exporter-stats timeout 1200 seconds
        Data template timeout 1200 seconds
    Exporter Statistics
        Number of Flow Records Exported 0
        Number of Templates Exported 0
        Number of Export Packets Sent 0
        Number of Export Bytes Sent 0
        Number of Destination Unreachable Events 0
        Number of No Buffer Events 0
        Number of Packets Dropped (No Route to Host) 0
        Number of Packets Dropped (other) 0
        Number of Packets Dropped (LC to RP Error) 0
        Number of Packets Dropped (Output Drops) 1
        Time statistics were last cleared: Never
switch(config-flow-exporter-version-9)# copy running-config startup-config
switch(config-flow-exporter-version-9)#
```

Defining a Flow Monitor

A flow monitor is associated with a flow record and a flow exporter.

A maximum of one flow monitor per interface or port profile per direction is permitted.

Before You Begin

- You know the name of an existing flow exporter to associate with this flow monitor.
- You know the name of an existing flow record to associate with this flow monitor. You can use either a flow record you previously created, or one of the following Cisco Nexus 1000V predefined flow records:
    - netflow-original

- netflow ipv4 original-input
- netflow ipv4 original-output
- netflow protocol-port

Note Cisco recommends that you use the predefined flow records for systems with a lower data rate. For systems operating at a higher data rate of more than 1 Gbps, Cisco recommends that you manually configure the flow record and use the 64-bit long counters.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **flow monitor** *name*	Creates a flow monitor by name, saves it in the running configuration, and places you in the CLI Flow Monitor Configuration mode.
Step 3	switch(config-flow-monitor)# **description** *string*	(Optional) For the specified flow monitor, adds a descriptive string of up to 63 alphanumeric characters, and saves it in the running configuration.
Step 4	switch(config-flow-monitor)# **exporter** *name*	For the specified flow monitor, adds an existing flow exporter and saves it in the running configuration.
Step 5	switch(config-flow-monitor)# **record** { [*name* | **netflow** {**ipv4**}] | **netflow-original** | **original-input** |**original-output** |**protocol-port**}	For the specified flow monitor, adds an existing flow record and saves it in the running configuration. • name: The name of a flow record you have previously created, or the name of a Cisco-provided, predefined flow record. • netflow: Traditional NetFlow collection schemes. • ipv4: Traditional IPv4 NetFlow collection schemes.
Step 6	switch(config-flow-monitor)# **show flow monitor** [*name*]	(Optional) Displays information about existing flow monitors.
Step 7	switch(config-flow-monitor)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

The following example shows how to create a flow exporter:

```
switch# configure terminal
switch(config)# flow monitor MonitorTest
switch(config-flow-monitor)# description Ipv4Monitor
switch(config-flow-monitor)# exporter ExportTest
```

```
switch(config-flow-monitor)# record RecordTest
switch(config-flow-monitor)# show flow monitor MonitorTest
Flow Monitor MonitorTest:
    Use count: 0
    Flow Record: RecordTest
    Flow Exporter: ExportTest
switch(config-flow-monitor)#
```

Assigning a Flow Monitor to an Interface

Before You Begin

- You know the name of the flow monitor you want to use for the interface.
- You know the interface type and its number.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *interface-type interface-number*	Places you in the CLI Interface Configuration mode for the specified interface.
Step 3	switch(config-if)# **ip flow monitor** *name* {**input** | **output**}	For the specified interface, assigns a flow monitor for input or output packets and saves it in the running configuration.
Step 4	switch(config-if)# **show flow interface** *interface-type interface-number*	(Optional) For the specified interface, displays the NetFlow configuration.
Step 5	switch(config-if)# **copy running-config startup-config**	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

The following example shows how to assign a flow monitor to an interface:

```
switch# configure terminal
switch(config)# interface veth 2
switch(config-if)# ip flow monitor MonitorTest output
switch(config-if)# show flow interface veth 2
Interface Vethernet2:
    Monitor: MonitorTest
    Direction: Output
switch(config-if)#
```

Adding a Flow Monitor to a Port Profile

Before You Begin

- You are logged in to the CLI in EXEC mode.
- You have already created the flow monitor.
- If using an existing port profile, you have already created the port profile and you know its name.
- If creating a new port profile, you know the type of interface (Ethernet or vEthernet), and you know the name you want to give it.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **port-profile** [**type** {**ethernet** | **vethernet**}] *name*	Enters port profile configuration mode for the named port profile.
Step 3	switch(config-port-prof)# **ip flow monitor** *name* {**input** | **output**}	Applies a named flow monitor to the port profile for either incoming (input) or outgoing (output) traffic.
Step 4	switch(config-port-prof)# **show port-profile** [**expand-interface**] [**name** *profile-name*]	(Optional) Displays the configuration for verification.
Step 5	switch(config-port-prof)# **copy running-config startup-config**	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

This example shows how to add a flow monitor to a port profile:

```
switch# configure terminal
switch(config)# port-profile AccessProf
switch(config-port-prof)# ip flow monitor access4 output
switch(config-port-prof)# show port-profile name AccessProf
port-profile AccessProf
  type: vethernet
  status: disabled
  capability l3control: no
  pinning control-vlan: -
  pinning packet-vlan: -
  system vlans: none
  port-group:
  max ports: 32
  inherit:
  config attributes:
    ip flow monitor access4 output
  evaluated config attributes:
    ip flow monitor access4 output
  assigned interfaces:
switch(config-port-prof)#
```

Verifying the NetFlow Configuration

Use one of the following commands to verify the configuration:

Command	Purpose
show flow cache	Displays information about NetFlow flow cache.
show flow exporter [*name*]	Displays information about NetFlow flow exporter.
show flow interface [*interface-type number*]	Displays information about NetFlow interfaces.
show flow monitor [*name* [**cache module** *number* | **statistics module** *number*]]	Displays information about NetFlow flow monitors. **Note** The **show flow monitor cache module** command differs from the **show flow monitor statistics module** command in that the cache command also displays cache entries.
show flow record [*name*]	Displays information about NetFlow flow records.
show flow timeout	Displays the NetFlow flow timeout setting.

Example: show flow exporter

```
switch(config-flow-exporter-version-9)# show flow exporter ExportTest
Flow exporter ExportTest:
    Description: ExportHamilton
    Destination: 192.0.2.1
    VRF: management (1)
    Destination UDP Port 200
    Source IP address 192.0.2.2
    Export from Line Card
    DSCP 2
    Export Version 9
        Exporter-stats timeout 1200 seconds
        Data template timeout 1200 seconds
    Exporter Statistics
        Number of Flow Records Exported 0
        Number of Templates Exported 0
        Number of Export Packets Sent 0
        Number of Export Bytes Sent 0
        Number of Destination Unreachable Events 0
        Number of No Buffer Events 0
        Number of Packets Dropped (No Route to Host) 0
        Number of Packets Dropped (other) 0
        Number of Packets Dropped (LC to RP Error) 0
        Number of Packets Dropped (Output Drops) 1
        Time statistics were last cleared: Never
switch(config-flow-exporter-version-9)#
```

Example: show flow interface

```
switch(config-if)# show flow interface veth2
Interface Vethernet2:
    Monitor: MonitorTest
    Direction: Output
switch(config-if)#
```

Example: show flow monitor

```
switch(config-flow-monitor)# show flow monitor
Flow Monitor MonitorTest:
    Use count: 1
    Flow Record: test
    Flow Exporter: ExportTest
Flow Monitor MonitorIpv4:
    Use count: 70
    Flow Record: RecordTest
    Flow Exporter: ExportTest
switch(config-flow-monitor)#
```

Example: show flow monitor cache module

```
switch(config-port-prof)# show flow monitor mDocs cache module 5
Cache type:                           Normal
Cache size (Bytes):                   224
Active Flows:                         8
Flows added:                          8
Packets added:                        228
Flows aged:                           0
    - Watermark aged                  0
    - Inactive timeout                0
    - Active timeout                  0
    - Event aged                      0
    - Emergency aged                  0
    - Permanent                       0
    - Immediate aged                  0
    - Session aged                    0
    - Fast aged                       0
    - Counters Overflow               0

            *  Denotes interface no longer exists, so just the IF Handle is displayed


  IPV4 SRC ADDR    IPV4 DST ADDR            INTF INPUT           INTF OUTPUT  FLOW DIRN
bytes       pkts
===============  ===============  ====================  ====================  =========
==========  ==========
   192.168.0.15     192.168.0.11                Veth4                 Veth6      Input
5390        55
   192.168.0.11     192.168.0.15                Veth6                 Veth4      Input
5390        55
   192.168.0.14     192.168.0.10                Veth1                 Veth5      Input
5292        54
   192.168.0.10     192.168.0.14                Veth5                 Veth1      Input
5292        54
```

Example: show flow monitor statistics module

```
switch(config)# show flow monitor m1 statistics module 3
Cache type:                           Normal
Cache size:                           0
Active Flows:                         1
Flows added:                          149
Packets added:                        350
Flows aged:                           148
    - Watermark aged                  0
    - Active timeout                  0
    - Inactive timeout                148
    - Event aged                      0
    - Emergency aged                  0
    - Permanent                       0
    - Immediate aged                  0
    - Session aged                    0
    - Fast aged                       0
    - Counters Overflow               0
switch(config)#
```

Example: show flow record

```
switch(config-flow-record)# show flow record RecordTest
Flow record RecordTest:
    Description: Ipv4flow
    No. of users: 0
    Template ID: 0
    Fields:
        match ipv4 destination address
        match interface input
        match interface output
        match flow direction
        collect counter packets
switch(config-flow-record)#
```

NetFlow Example Configuration

The following example shows how to configure flow monitor using a new flow record and apply it to an interface:

```
switch# configure terminal
switch(config)# flow record RecordTest
switch(config-flow-record)# description Ipv4flow
switch(config-flow-record)# match ipv4 destination address
switch(config-flow-record)# collect counter packets
switch(config-flow-record)# exit
switch(config)# flow exporter ExportTest
switch(config-flow-exporter)# description ExportHamilton
switch(config-flow-exporter)# destination 192.0.2.1
switch(config-flow-exporter)# dscp 2
switch(config-flow-exporter)# source lc-exp 192.0.2.2/24
switch(config-flow-exporter)# transport udp 200
switch(config-flow-exporter)# version 9
switch(config-flow-exporter-version-9)# option exporter-stats timeout 1200
switch(config-flow-exporter-version-9)# template data timeout 1200
switch(config-flow-exporter-version-9)# exit
switch(config-flow-exporter)# exit
switch(config)# flow monitor MonitorTest
switch(config-flow-monitor)# description Ipv4Monitor
switch(config-flow-monitor)# exporter ExportTest
switch(config-flow-monitor)# record RecordTest
switch(config-flow-monitor)# exit
switch(config)# interface veth 2
switch(config-if)# ip flow monitor MonitorTest output
switch(config-if)# show flow interface veth 2
Interface Vethernet2:
    Monitor: MonitorTest
    Direction: Output
switch(config-if)#
```

The following example shows how to configure flow monitor using a predefined record and apply it to an interface:

```
switch# configure terminal
switch(config)# flow exporter ExportTest
switch(config-flow-exporter)# description ExportHamilton
switch(config-flow-exporter)# destination 192.0.2.1
switch(config-flow-exporter)# dscp 2
switch(config-flow-exporter)# source lc-exp 192.0.2.2/24
switch(config-flow-exporter)# transport udp 200
switch(config-flow-exporter)# version 9
switch(config-flow-exporter-version-9)# option exporter-stats timeout 1200
switch(config-flow-exporter-version-9)# template data timeout 1200
switch(config-flow-exporter-version-9)# exit
switch(config-flow-exporter)# exit
switch(config)# flow monitor MonitorTest
switch(config-flow-monitor)# description Ipv4Monitor
```

```
switch(config-flow-monitor)# exporter ExportTest
switch(config-flow-monitor)# record netflow-original
switch(config-flow-monitor)# exit
switch(config)# interface veth 2
switch(config-if)# ip flow monitor MonitorTest output
switch(config-if)# show flow interface veth 2
Interface Vethernet2:
    Monitor: MonitorTest
    Direction: Output
switch(config-if)#
```

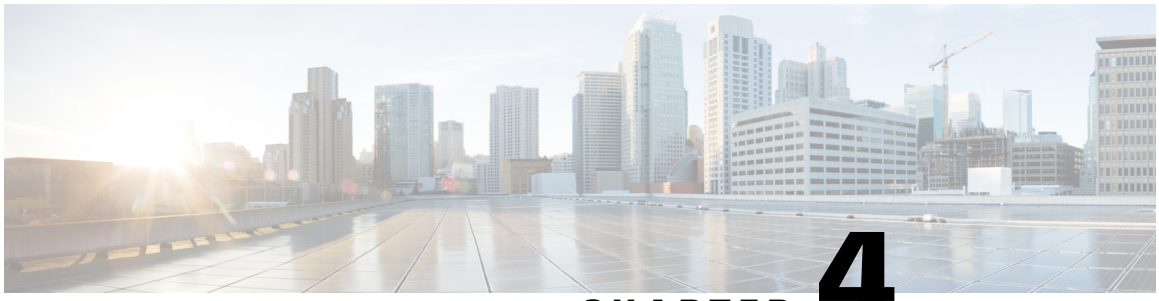
Related Documents for NetFlow

Related Topic	Document Title
Cisco NetFlow Overview	http://www.cisco.com/c/en/us/products/ios-nx-os-software/ios-netflow/index.html

Feature History for NetFlow

This table includes only the updates for those releases that have resulted in additions or changes to the feature.

Feature Name	Releases	Feature Information
NetFlow	2.2(1)	Distributed NetFlow was introduced.



CHAPTER 4

Configuring Cloud Security Groups

This chapter contains the following sections:

Information About Cloud Security Groups

Cisco Intercloud Fabric security enhancements further secure the hybrid cloud environments by enforcing security on the hybrid connection control points. Security is enforced through the use of filtering capabilities automatically applied to both ends of the site-to-site encrypted tunnel; ICX on the private cloud and ICS on the public cloud. Traffic is controlled so that only cloud resources under the management control of Cisco Intercloud Fabric are allowed to communicate to one another, thus preventing using the site-to-site tunnel as a transit link to the Internet or cloud provider's networks. By using Cloud Security Groups, hybrid cloud environments have a default policy applied to prevent traffic that does not belong to any subnet extended to the public cloud from leaving the private cloud or entering the private cloud from the public cloud.

Cloud Security Groups are classified into Enterprise Security Groups and Public Security Groups:

- Enterprise Security Groups are ICX-based security groups that control traffic leaving the private cloud by only allowing traffic from source IP addresses that belong to the networks that are extended to the public clouds.
- Public Security Groups are S-based security groups that control traffic destined to the private cloud from the public cloud by only allowing traffic from source IP addresses that belong to the enterprise IP space extended to the public cloud.

Note Cisco Intercloud Fabric has management ports and interfaces that are enabled by default. It is highly recommended you restrict device access from authorized hosts and protocols using only Infrastructure ACLs. For example:

```
!
ip access-list ACL-INFRASTRUCTURE-IN
!!---Permit secure connections for network management
permit tcp host <trusted-management-stations><icfSwitch> eq 22
!
interface mgmt0
  ip access-group ACL-INFRASTRUCTURE-IN in
!
```

Refer to Cisco Security White Paper Securing the Management Plane for more information.

Guidelines and Limitations for Cloud Security Groups

You can configure Cloud Security Groups from the CLI only. Intercloud Fabric does not expose the Cloud Security Groups configuration.

Updating the Cloud Security Groups Configuration

You can manually configure Cloud Security Groups by creating an ACL that is applied to the ICX and ICS trunk tunnel.

Before You Begin

- You must have administrator privileges to configure Cloud Security Groups.

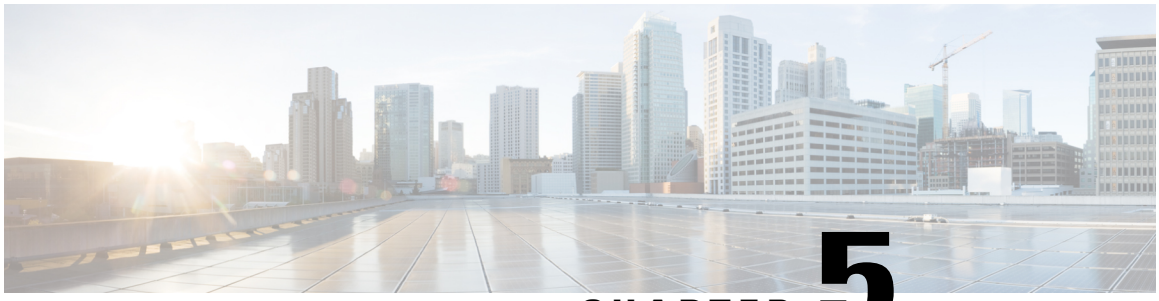
Procedure

	Command or Action	Purpose
Step 1	**ip access-list** <ACL_name> **20 deny ip any any 10 permit ip X.X.Y.0/24 X.X.Y.0/24 copy r s exit**	Creates the ACL rule. • For **<ACL_Name>**, specify the name of the ACL. • The **permit ip** only allows traffic to or from this network.
Step 2	**show ip access-lists**	Verifies the created ACL list.
Step 3	**show run Port-profiles**	Checks the port profiles.
Step 4	**port-profile** <ICX_Tunnel_Trunk> **ip port access-group** <ACL_name> **in ip port access-group** <ACL_name> **out exit**	Applies the ACL to the ICX trunk tunnel. • For **<ACL_Name>**, specify the name of the ACL. • The **ip port access-group <ACL_name> in** applies to inbound traffic.

	Command or Action	Purpose
		• The **ip port access-group <ACL_name> out** applies to outbound traffic.
Step 5	**port-profile** <ICS_Tunnel_Trunk> **ip port access-group** <ACL_name> **in ip port access-group** <ACL_name> **out exit**	Applies the ACL to the ICS trunk tunnel. • For **<ACL_Name>**, specify the name of the ACL. • The **ip port access-group <ACL_name> in** applies to inbound traffic. • The **ip port access-group <ACL_name> out** applies to outbound traffic.

The following example shows how to manually configure an ACL, verify the ACL list, check the port profiles, and apply the rule to both the ICX and ICS trunk tunnel:

```
ip access-list <ACL_name>
20 deny ip any any
10 permit ip X.X.Y.0/24 X.X.Y.0/24
copy r s
exit
show ip access-lists
show run Port-profiles
port-profile <ICX_Tunnel_Trunk>
ip port access-group <ACL_name> in
ip port access-group <ACL_name> out
exit
port-profile <ICS_Tunnel_Trunk>
ip port access-group <ACL_name> in
ip port access-group <ACL_name> out
exit
```



CHAPTER 5

Managing the Supported Providers List

This chapter contains the following sections:

Information About the Supported Providers List

The supported providers list allows you to manually update the Cisco Intercloud Fabric Director (ICFD) providers list as new providers are supported. The supported providers list can be used to manually create a customized ICFD providers list for distribution to customers.

Guidelines and Limitations for the Supported Providers List

Use extreme caution when deploying this feature in a live system and when other users are using the ICFD UI. The service stop/start will cause unwanted service outages. We recommended that you deploy this feature prior to infrastructure setup.

Updating the Supported Providers List

You can manually update the ICFD providers list as new providers are supported.

Before You Begin

- Ensure that you have access credentials for the ICFD ShellAdmin user.
- From Cisco.com, download the *create_patch*_providerslist.sh and supported_providers files to your desktop.

Note During the installation of the patch, the browser to ICFD is not available. When the service is restarted, you can view the new supported providers list as part of the icfCloud creation.

Procedure

Step 1 Log in to ICFD using ShellAdmin user credentials and enter your password to access the CLI mode. The Intercloud Fabric Shell Menu, Standalone Node menu appears.

Step 2 At the SELECT> prompt, enter **3**, and then click **return** to stop services.

Step 3 At the SELECT> prompt, enter **16**, and then click **return** to apply patch.

Step 4 At the **Do you want to take database backup before applying patch [y/n]?** prompt, enter **N**, and then click **return**.

Step 5 At the Patch URL: prompt, enter your **<web server address/patch file path and filename>**.

Step 6 At the SELECT> prompt, enter **4**, and then click **return** to start services.

Note It can take up to 1 minute for all services to come up.

Step 7 At the SELECT> prompt, enter **2**, and then click **return** to display the services status.
Check that all services are RUNNING.

Step 8 Log in to ICFD using admin credentials and choose **Intercloud > IcfCloud > Account Credentials**.
Review the Cloud Type drop-down list for the updated supported providers list.

The following example shows how to manually update the ICFD supported providers list:

```
SELECT> 3
SELECT> 16
Do you want to take database backup before applying patch [y/n]? N
Patch URL: http://10.7.1.7/patch/icfd-providers-patch-2.1.1.zip
SELECT> 4
SELECT> 2
```

Creating a Customized Supported Providers List

You can manually create a customized ICFD providers list for distribution to customers.

Before You Begin

- Ensure that you have access credentials for the ICFD ShellAdmin user.

Note During the installation of the patch, the browser to ICFD is not available. When the service is restarted, you can view the new supported providers list as part of the icfCloud creation.

Procedure

Step 1 From your desktop, download the script file and the supported_providers file from Cisco.com.

Step 2 Update the supported_providers file, in its existing format, to include new supported providers on the list.

Step 3 Copy the file to your Linux system.

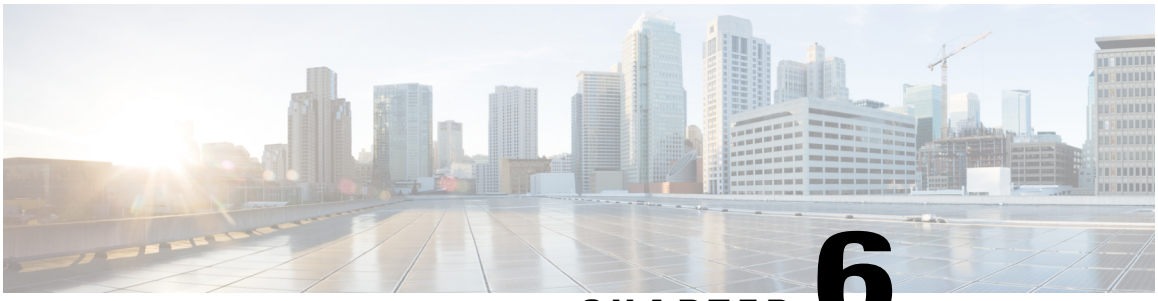
Step 4 Run the script file from within your Linux system to generate the patch:

Example:
<script_name> -p <supported*providers*_file>

Step 5 Copy the generated patch file to a web server that has network reachability with ICFD.

The following example shows how to manually create a customeized ICFD supported providers list:

```
<script_name> -p <supportedproviders_file>
```



CHAPTER 6

Configuring Network Parameters

This chapter contains the following sections:

Information About Network Parameters

Network parameters in Cisco Intercloud Fabric Director (ICFD) lets you manually modify parameters—IP address, network mask, default gateway, and so on—to update or correct misconfigured parameters.

Guidelines and Limitations for Network Parameters

Use extreme caution when deploying this feature in a live system and when other users are using the ICFD UI. Modifying network parameters in a live system could cause unwanted service outages. We recommend that you modify any IP-related configuration or DNS configuration prior to infrastructure setup.

Modifying Network Parameters

Before You Begin

- Ensure that you have access credentials for the ICFD ShellAdmin user.

Procedure

Step 1 Log in to ICFD using ShellAdmin user credentials and enter your password to access the CLI mode. The Intercloud Fabric Shell Menu, Standalone Mode menu appears.

Step 2 At the SELECT> prompt, enter **14**, and then click **return**. The current network interface details are displayed.

Step 3 Click **return**. The Intercloud Fabric Shell Menu, Standalone Mode menu appears.

Step 4 At the SELECT> prompt, enter **13** to change network interface parameters.

Step 5 At the **Do you want to Configure DHCP/STATIC IP [D/S] ?:** prompt, enter **S** to configure the static configuration.

Step 6 At the **Enter the ethernet interface that you want to configure E.g. eth0 or eth1:** prompt, enter **eth0** to select the Ethernet interface that you want to configure.

Step 7 At the **Select the IP version you want to configure [a)IPv4, b) IPv6] a/b:** prompt, enter **a** to select the IP version you want to configure.

Step 8 At the **Do you want to configure IPv4 STATIC IP for eth0 [y/n]?** prompt, enter **y** to specify that you want to configure IPv4 static ip address.

Step 9 Enter the address details as prompted and click **return**:

- IP Address
- Netmask
- Gateway
- DNS Server1
- DNS Server2—Click **return** if you do not have a secondary DNS server2

Step 10 Review the configuration details, and then enter **y** at the prompt to change the specified network parameters. [**OK**] is displayed as each change is verified.

Step 11 Click **return**. The Intercloud Fabric Shell Menu, Standalone Mode menu appears.

The following example shows how to manually modify ICFD network parameters:

```
SELECT> 13
Do you want to Configure DHCP/STATIC IP [D/S]?: S
Configuring STATIC configuration...

Enter the ethernet interface that you want to configure E.g. eth0 or eth1: eth0
Select the IP version you want to configure [a) IPv4, b) IPv6] a/b: a
Do you want to configure IPv4 STATIC IP for eth0 [y/n]? y
Configuring STATIC IP for eth0...
 IP Address: 10.36.7.15
 Netmask: 255.255.0.0
 Gateway: 10.36.0.2
 DNS Server1: 171.70.168.183
 DNS Server2:
Configuring Network with : INTERACE(eth0), IP(10.36.7.15), Netmask(255.255.0.0),
Gateway(10.36.0.2),DNS Server1(171.70.168.183), DNS Serverx 2()

Do you want to continue [y/n]? : y
Configuring STATIC IP for eth0
Successfully configured static IP address 10.36.7.15
Shutting down interface eth0:                          [  OK  ]
Shutting down loopback interface:                      [  OK  ]
Bringing up loopback interface:                        [  OK  ]
```

```
Bringing up interface eth0:                          [  OK  ]
Press return to continue …
```