Configuring ITD

This chapter describes how to configure the Intelligent Traffic Director (ITD) on the Cisco NX-OS device.

About ITD

Intelligent Traffic Director (ITD) is an intelligent, hardware-based, multi-terabit solution that allows you to build a scalable architecture for Layer 3 and Layer 4 traffic distribution, load balancing, and redirection.

Benefits of ITD:

- Multi-terabit solution at line rate
- Transparency to end device and stateless protocol benefits
- Reduced complexities and architecture scaling for alternative features like Web Cache Communication Protocol (WCCP) and policy-based routing
- Simplified provisioning and ease of deployment
- Legacy service appliances can co-exist with new ones
- Removes the requirement for an expensive external load balancer
- No certification, integration, or qualification needed between the devices and the Cisco NX-OS switch
- Order of magnitude OPEX savings : reduction in configuration, and ease of deployment
- CAPEX savings : No service module or external L3/L4 load-balancer needed. Every Nexus port can be used as load-balancer

ITD features:

- Hardware based multi-terabit/s L3/L4 load-balancing at wire-speed
- Zero latency load-balancing
- Redirect line-rate traffic to any devices, for example web cache engines, Web Accelerator Engines (WAE), video-caches, etc
- Capability to create clusters of devices, for example, Firewalls, Intrusion Prevention System (IPS), or Web Application Firewall (WAF), Hadoop cluster
- IP-stickiness

- Hardware based multi-terabit/s L3/L4 load-balancing at wire-speed
- Zero latency load-balancing
- Redirect line-rate traffic to any devices, for example web cache engines, Web Accelerator Engines (WAE), video-caches, etc
- Capability to create clusters of devices, for example, Firewalls, Intrusion Prevention System (IPS), or Web Application Firewall (WAF), Hadoop cluster
- IP-stickiness

- Resilient (like resilient ECMP), Consistent hash
- Virtual IP based L4 load-balancing
- Weighted load-balancing and Failaction are supported among nodes
- Load-balances to large number of devices/servers
- ACL along with redirection and load balancing simultaneously
- Bi-directional flow-coherency. Traffic from A–>B and B–>A goes to same node
- The servers/appliances don’t have to be directly connected to Nexus switch
- Monitoring the health of servers/appliances with IP SLA-based probes
- N + M redundancy (N number of nodes and M number of hot-standbys)
- Automatic failure handling of servers/appliances
- VRF support, vPC support
- Supports both IPv4 and IPv6 (all platforms do not support IPv6)
- The feature does not add any load to the supervisor CPU
- Handles unlimited number of flows
- Nondisruptive node addition or deletion
- Simultaneous redirection and load balancing
- Rate sharing across multiple ITD services in the same switch

Use case examples:

- Load-balance to cluster of Firewalls.

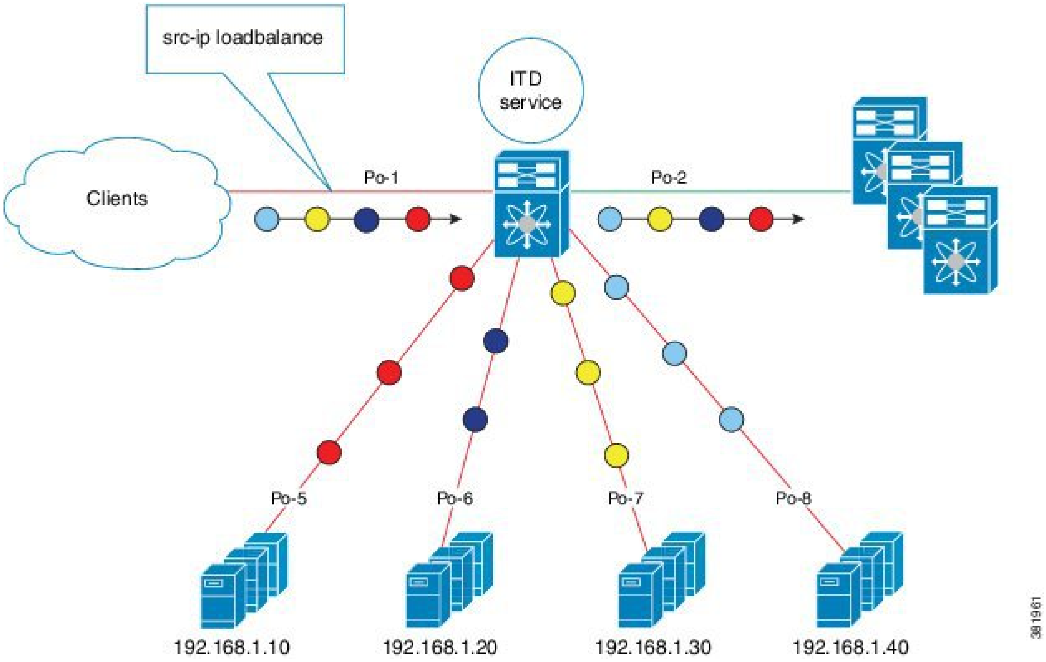
- Scale IPS, IDS and WAF by load-balancing to NX-OS devices
- Scale the NFV solution by load-balancing to low cost VM/container based NFV
- Scale the WAAS / WAE solution. Traffic redirection mechanism for the Wide Area Application Services (WAAS) or Web Accelerator Engine (WAE) solution
- Scale the VDS-TC (video-caching) solution
- Scale Layer-7 load-balancers, by distributing traffic to L7 LBs
- Replaces ECMP or the port channel to avoid rehashing . ITD is resilient, and doesn't cause re-hashing on node add/delete/failure
- Server load balancing in DSR (Direct Server Return) mode
- Scales up NG intrusion prevention systems (IPSs) and web application firewalls (WAFs) by load balancing to NX-OS devices
- Load balances to Layer 5 through Layer 7 load balancers

Deployment Modes

One-Arm Deployment Mode

You can connect servers to the switch in one-arm deployment mode. In this topology, the server is not in the direct path of client or server traffic, which enables you to plug a server into the network with no changes to the existing topology or network.

Figure 1: One-Arm Deployment Mode

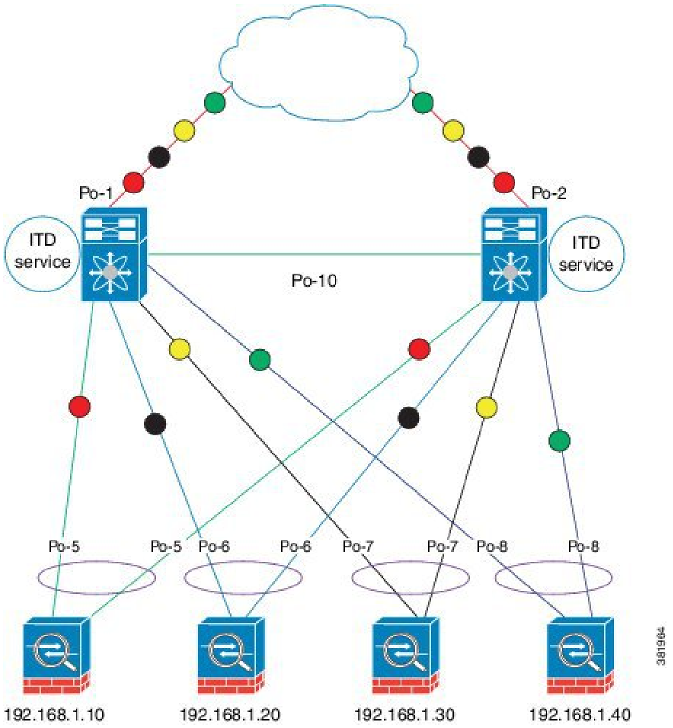


One-Arm Deployment Mode with vPC

ITD supports an appliance pool connected to a virtual port channel (vPC). The ITD service runs on each switch, and ITD programs each switch to provide flow-coherent traffic passing through the nodes.

Note It is recommended to use failaction bucket distribute for VPC scenarios (not using ITD NAT) to keep consistent behavior across peers on failures of nodes reachable over VPC.

Figure 2: One-Arm Deployment Mode with vPC



Sandwich Deployment Mode

The sandwich deployment mode uses two switches to provide stateful handling of traffic.

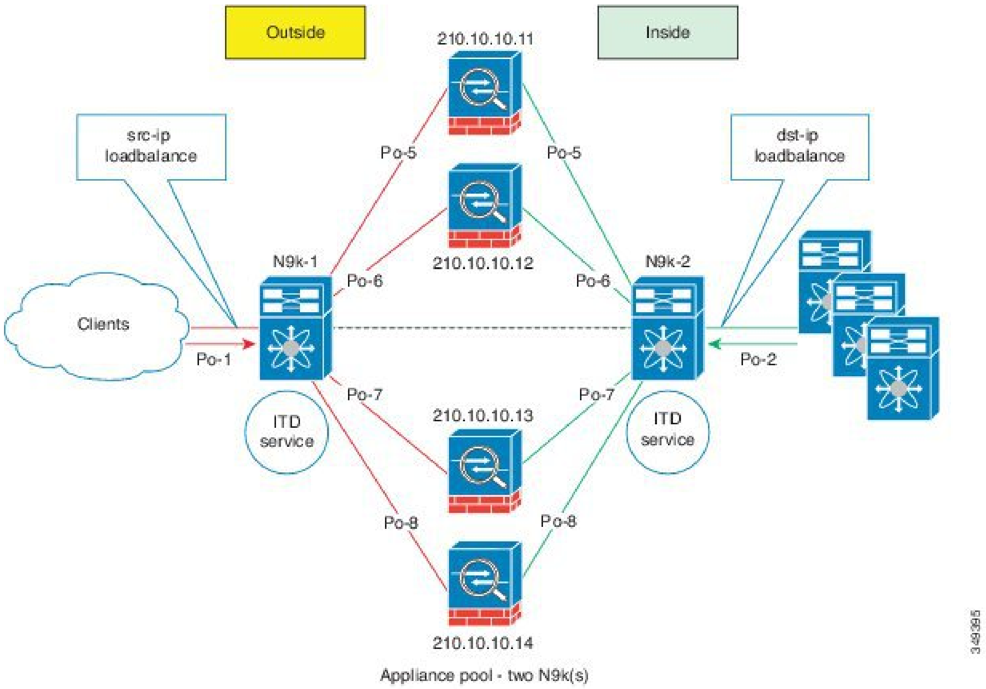
The main requirement in this mode is that both the forward and reverse traffic of a flow must go through the same appliance. Examples include firewalls and load balancer deployments, where traffic between the client and the server must flow through the same appliance.

The key features are:

- An ITD service for each network segment, one for the outside network and another for the inside network.
- A source IP address load-balancing scheme where the ITD service operates on the interface that connects to the outside world in an ingress direction.
- A destination IP address load-balancing scheme where the ITD service operates on the interface that connects to the servers in the ingress direction.

- If a user-defined access-list (include ACL) is used in the ITD service in the outside network, an access-list with reversed ACE rules should be created and applied as a user ACL in the ITD service in the inside network.

Figure 3: Sandwich Deployment Mode

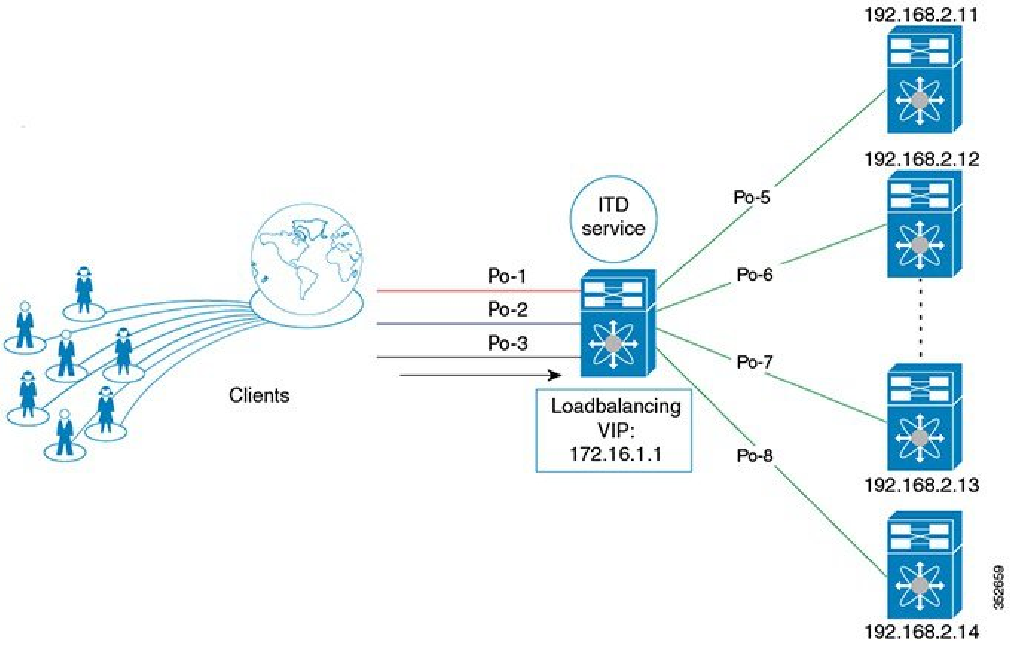


Server Load-Balancing Deployment Mode

The ITD service can be configured to host a virtual IP (VIP) on the switch. Internet traffic destined for the VIP will be load balanced to the active nodes. The ITD service is not a stateful load balancer.

Note You need to configure the ITD service manually and in a similar manner on each switch.

Figure 4: ITD Load Distribution with VIP



Device Groups

Nodes can be a physical server, virtual server, or a service appliance where traffic can be load balanced. These nodes are grouped together under a device group, and this device group can be mapped to a service.

ITD supports device groups. When you configure a device group, you can specify the following:

- The device group's nodes
- The device group's probe

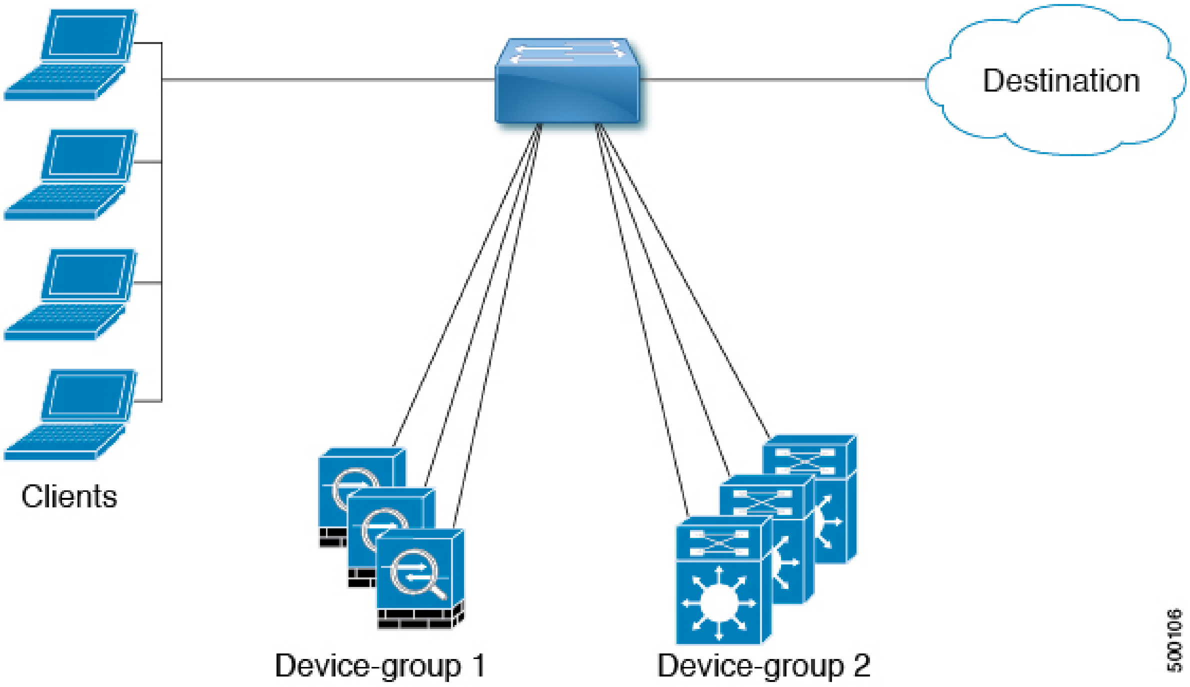
You can configure probes at the device-group level or at the node level. With node-level probing, each node can be configured with its own probe, allowing for further customization per node. Node-level probes are useful in scenarios where each node needs to be monitored differently for failure conditions.

Multiple Device Groups in an ITD Service

device groups are supported in an ITD service (as shown in the figure below). An ITD service generates a single route map with different sequences that point to different device groups.

Each device group represents different types of traffic requiring different services but arriving on the same ingress interface. Traffic on the interface is redirected to the appropriate device group based on the virtual IP address. Supporting multiple device groups per ITD service on the same interface allows ITD to scale.

Figure 5: Multiple Device Groups in an ITD Service



For a configuration example showing how to configure multiple device groups in an ITD service, see .

VRF Support

The ITD service can be configured in the default VRF as well as in non-default VRFs.

Ingress interfaces and device-group nodes must all belong to the same VRF for the ITD service to redirect traffic. You must ensure that all ingress interfaces and node members of the associated device group are reachable in the configured VRF.

Router ACLs

The switch supports router access control lists (RACLs) with ITD.

You can configure ITD and an RACL on the same ingress interface. The resulting RACL, which is downloaded to the TCAM, is a cross product of the ACL generated by ITD and the user-configured RACL. The permit and deny statements configured on the RACL are combined with the ACL permits and redirect entries created by ITD. This functionality helps you to filter and load distribute selected traffic.

Note ITD statistics do not function if you configure an RACL on an ITD ingress interface.

Include and Exclude ACLs

Include ACL

The include ACL feature allows you to assign an access control list (ACL) to an ITD service. Only traffic matching the ACE is load-balanced toward the nodes and other traffic follows default routing rules.

Beginning from Cisco NX-OS Release 9.3 (3), you can configure up to 8 access-lists under one ITD service. You can associate each access list with its own device-group (Multi-ACL). When specific device-group is associated with one user ACL, that device-group takes priority and overwrites the default device-group. With this feature, ITD can load-balance traffic matching different ACLs to different device-groups.

Exclude ACL

You can configure an exclude ACL to specify the traffic that you want ITD to exclude from the ITD load balancer. The traffic, which the exclude ACL selects, is RIB-routed and bypasses ITD. An exclude ACL can filter based on both source and destination fields. The exclude ACL precedes the virtual IP address.

Virtual IP Address Filtering

A virtual IP address can be used to filter traffic for ITD. A virtual IP address and subnet mask combination for traffic filtering is supported for the destination field only.

Port Number-Based Filtering

Port numbering can be used to filter traffic for ITD. The following methods are supported to filter traffic based on Layer 4 ports (for example, port 80):

- Matching destination ports

  Any source or destination IP address with destination port 80 is matched. (For example: The virtual IP address is configured as **0.0.0.0 0.0.0.0 tcp 80**.)

- Matching source ports

  Any port other than 80 bypasses ITD, and port 80 is redirected. (For example: The exclude ACL is configured as **permit tcp any neq 80 any**.)

- Matching multiple port numbers

  Multiple virtual IP address lines in ITD can be configured, one for each port.

Hot-Standby

The hot-standby feature reconfigures the switch to look for an operational hot-standby node and select the first available hot-standby node to replace the failed node. ITD reconfigures the switch to redirect the traffic segment that was originally headed toward the failed node to the hot-standby node. The service does not impose any fixed mapping of hot-standby nodes to active nodes.

When the failed node becomes operational again, it is reinstated as an active node. The traffic from the acting hot-standby node is redirected back to the original node, and the hot-standby node reverts to the pool of standby nodes.

When multiple nodes fail, traffic destined to all failed nodes gets redirected to the first available hot-standby node.

The hot-standby node can be configured only at the node level . At the node level, the hot-standby node receives traffic only if its associated active node fails.

ITD supports N + M redundancy where M nodes can act as hot-standby nodes for N active nodes.

Multiple Ingress Interfaces

You can configure the ITD service to apply traffic redirection policies on multiple ingress interfaces. This feature allows you to use a single ITD service to redirect traffic arriving on different interfaces to a group of nodes.

same ingress interface can be included in two ITD services, allowing one IPv4 ITD service and one IPv6 ITD service.

Including the same ingress interface in both IPv4 and IPv6 ITD services allows both IPv4 and IPv6 traffic to arrive on the same ingress interface. An IPv4 ITD policy is applied to redirect the IPv4 traffic, and an IPv6 ITD policy is applied to redirect the IPv6 traffic.

Note Make sure that the same ingress interface is not referenced in more than one IPv4 ITD service or more than one IPv6 ITD service. The system does not automatically enforce it and it is not supported.

System Health Monitoring

ITD monitors health of the nodes and applications running on those nodes periodically to detect any failures and to handle the failure scenarios.

ICMP, TCP, UDP probes are supported.

Health of an Interface Connected to a Node

leverages the IP service level agreement (IP SLA) feature to periodically probe each node. ITD uses the Internet Control Message Protocol (ICMP) to periodically probe each node. The probes are sent at a 10-second frequency by default and can be configured down to 1 second. They are sent simultaneously to all nodes. You can configure the probe as part of the pool group configuration.

A probe is declared to have failed after retrying three times by default. At this point, the node state becomes “Failed,” and its status becomes “PROBE_FAILED.”

Node Failure Handling

Upon marking a node as down, the ITD performs the following tasks automatically to minimize traffic disruption and to redistribute the traffic to remaining operational nodes:

- Determines if a standby node is configured to take over from the failed node.
- If the standby node is operational, it is identified the node as a candidate node for traffic handling.
- Redefines the standby node as active for traffic handling, if an operational standby node is available
- Programs automatically to reassign traffic from the failed node to the newly active standby node.

Peer Synchronization

The peer synchronization feature synchronizes the node health status across two ITD peer services in sandwich mode. It is useful in preventing traffic loss if a link on one of the ITD peer services goes down.

Each ITD service probes its peer service periodically to detect any failure. A ping is sent every second to the ITD peer service. If a reply is not received, it is retried three times. The frequency and retry count are not configurable.

Note Peer-service feature requires fail-action least-bucket or fail-action node per-bucket to be configured, to allow for synchronized fail-over of nodes across services. Additionally synchronized fail-over is not supported when either service is using hot-standby nodes or node level standbys.

Failaction Reassignment

Failaction for ITD enables traffic to the failed nodes to be reassigned to one or more active nodes. When the failed node becomes active again, it resumes serving connections. If all the nodes are down, the packets are routed automatically. All Failaction mechanisms are supported for both IPv4 and IPv6 services.

Note You must configure a probe under an ITD device group before enabling the failaction feature.

Failaction Node Reassign

When a node goes down, the traffic buckets associated with the node are reassigned to the first active node found in the configured set of nodes. If the newly reassigned node also fails, traffic is reassigned to the next available active node.

Failaction Node Least-Bucket

When a node goes down, the traffic buckets associated with the node are reassigned to an active node that is currently receiving traffic from the least number of traffic buckets. For each subsequent node failure, the active node with least traffic buckets is recomputed and all the buckets directed to a failed node are redirected to this node, thereby allowing the re-assigned buckets to be distributed over multiple active nodes.

Failaction Bucket Distribute

When the service is enabled, ITD uses an internal algorithm to preselect varied sequences of primary nodes as alternate backup paths for with different priorities for each primary node. When a node goes down, the traffic to the node will be re-directed to the first active backup node with the highest priority, and so on, for subsequent failures, thereby minimizing the convergence delays.

the primary nodes of a device-group or up to 32 primary nodes of a device-group (whichever is lesser) shall be preselected with different priorities for each node.

Note This algorithm is intended for relatively even traffic distribution but doesn't guarantee even distribution with node failures.

Failaction Optimization

Prior to Cisco NX-OS Release 9.2 (2), when the node goes down, the buckets associated with the node are reassigned to an active node as determined by the fail-action algorithm. However if the newly reassigned node has also failed simultaneously, the traffic buckets for the original failed node have to be re-assigned to another active node, after re-running the fail-action computation. The delay in reassigning the failed node buckets to an active node impacts the network performance.

With fail-action optimization, when a node goes down, the status of all available nodes is first proactively fetched. The re-assignment of all nodes detected as failed will then be done based on the fail-action mechanism, thereby avoiding the delays in repeated re-assignment.

Failaction Node-Per-Bucket

When a particular node fails, the node with least number of buckets are identified and the buckets are distributed across the other active nodes, starting from the node with least buckets.

ITD repeatedly identifies the least buckets node currently and assign one bucket to the node until all buckets are reassigned. Hence all buckets are distributed evenly among all remaining active nodes.

Note identifies the nodes to fail-over, based on the weights of the nodes. If a node doesn't have a weight configured a default weight of 1 is used.

No Failaction Reassignment

When failaction node reassignment is not configured, there are two possible scenarios:

No Failaction Reassignment with a Probe Configured

The ITD probe can detect the node failure or the lack of service reachability. If the node fails, the traffic is routed and does not get reassigned, as failaction is not configured. Once the node recovers, the recovered node starts to handle the traffic.

No Failaction Reassignment without a Probe Configured

Without a probe configuration, ITD cannot detect the node failure. When the node is down, ITD does not reassign or redirect the traffic to an active node.

Licensing Requirements

For a complete explanation of Cisco NX-OS licensing recommendations and how to obtain and apply licenses, see the *Cisco NX-OS Licensing Guide* and the *Cisco NX-OS Licensing Options Guide*.

Guidelines and Limitations for ITD

ITD has the following configuration guidelines and limitations:

- ITD is supported on the following platforms:

  ITDv4 support

  - Beginning from Cisco Nexus NX-OS Release 9.2 (1), Cisco Nexus C9364C, C9336C-FX2, C93240YC-FX2 switches are supported.
  - Cisco Nexus 9500 Series switches with Cisco Nexus X9432PQ, X9464PX, X9464TX, X9536PQ, X9564PX, X9564TX, and X9636PQ line cards.
  - Cisco Nexus 9332PQ, 9372PX, 9372PX-E, 9372TX, 9372TX-E, 9396PX, 9396TX, 93120TX, and 93128TX switches.
  - Cisco Nexus 9236C, 92160YC-X, and 92304QC switches, 93180YC-EX, 93108TC-EX, C93180YC-FX, C93108TC-FX and 9232C switches

  ITDv6 support

  - Cisco Nexus 9236C, 92160YC-X, and 92304QC switches, 93180YC-EX, 93108TC-EX, C93180YC-FX, C93108TC-FX and 9232C switches
  - Beginning from Cisco Nexus NX-OS Release 9.2 (1), Cisco Nexus C9364C, C9336C-FX2, C93240YC-FX2 switches are supported.

- ITD does not support using FEX ports for ingress or egress to the next-hop IP address.
- Configuration rollback is supported only when the ITD service is in shut mode in both the target and source configurations.
- ITD is not supported with VXLAN. For load balancing in VXLAN, use pervasive load balancing. For more information, see the Cisco Programmable Fabric with VXLAN BGP EVPN Configuration Guide.
- SNMP is not supported for ITD.
- IPv6 does not support node level probes, only device group level probes are supported.
- IPv6 device group level probes support TCP, ICMP protocols but do not support HTTP, UDP, and DNS protocols.
- Weighted load balancing is not supported with the ITD **failaction** commands.
- The **bucket distribution** options is available for IPv4 and IPv6.

Note Fail-action bucket distribute is not recommended for services using hot-standby nodes or node-level standby nodes.

- The following guidelines and limitations apply to the exclude ACL feature:
  - The exclude ACL supports only permit access control entries (ACEs). Deny ACEs are not supported.
  - Traffic that is matched by a permit ACE in an exclude ACL bypasses ITD.
- The following guidelines and limitations apply to the include ACL feature:
  - Only one user-defined ACL is supported.

  - Only ACEs with the **permit** method are supported in the ACL. ACEs with any other method (such as **deny** or **remark**) are ignored.
  - A maximum of 256 permit ACEs are supported in one ACL.
  - Failaction is supported among the nodes.
  - ITD supports either the include ACL feature or the virtual IP address (VIP) feature but not both.
  - If the user has configured ITD with include or exclude ACL, and the user is using source IP-based load balancing, then the subnet mask in the source IP address of the ACE cannot be more than /28 (eg, it cannot be /29, /30, /31)

    If the user has configured ITD with include or exclude ACL, and the user is using destination IP-based load balancing, then the subnet mask in the destination IP address of the ACE cannot be more than /28 (eg, it cannot be /29, /30, /31)
  - The mask position cannot be configured for the include ACL feature.

- We recommend that you classify probe traffic in a separate CoPP class. Otherwise, probe traffic will go in the default CoPP class by default and might be dropped, causing IP SLA bouncing for probe traffic. For configuration information, see Configuring CoPP for IP SLA Packets
- ITD sessions are not supported with the following:
  - Weights
  - The include and exclude ACL features
  - Node level probes
  - Device-groups with hot-standby or node level standby nodes.
  - Device-groups being used by services with peer synchronization enabled.
  - Peer sync should have same number of nodes across the two services and the order of the nodes has to be same across the two services, with symmetric flow.
  - Services with layer-4 load-balance options configured.
  - Services with multiple Virtual IPs using different device-groups.
- Disabling atomic update may allow more TCAM resources to be made available for the ITD policies, but with possible disruption in traffic during changes to policies. For further details, please refer to Security Configuration Guide 9.2(x).

ITD Support Summary

See the following table for a list of the ITD support levels.

Table 1: ITD support levels

Feature	ITDv4	ITDv6	Comments
Device group level	• TCP • ICMP • HTTP • UDP • DNS	• TCPv6 • ICMPv3	
Per Node-Probe Level	Yes	Yes	
Hot-Standby	Yes	Yes	
Weight	Yes	Yes	
Non-Disruptive Operation			
ACL Refresh	Yes	Yes	
Primary Nodes	Yes	Yes	
Hot Standby Nodes	No	No	
Service-Level			
Include ACL	Yes	Yes	
Failaction methods	• **reassign** • **least-bucket** • **node-per-bucket** • **bucket distribute**	• **reassign** • **least-bucket** • **node-per-bucket** • **bucket distribute**	
Exclude-ACL	Yes	Yes	The deny ACEs are not supported.

Feature	ITDv4	ITDv6	Comments
Supported Platforms	Cisco Nexus 9236C, 92160YC-X, 92304QC switches and, 9300-EX Series switches Cisco Nexus 9332PQ, 9372PX, 9372PX-E, 9372TX, 9372TX-E, 9396PX, 9396TX, 93120TX,and 93128TX switches. Cisco Nexus X9432PQ, X9464PX, X9464TX, X9536PQ, X9564PX, X9564TX, and X9636PQ line cards.	Cisco Nexus 93180YC-EX, 93108TC-EX, C93180YC-FX, and C93108TC-FX switches. Cisco Nexus C9364C, C9336C-FX2, C93240YC-FX2 switches	

Default Settings for ITD

This table lists the default settings for ITD parameters.

Table 2: Default ITD Parameters

Parameters	Default
Probe frequency	10 seconds
Probe retry down count	3
Probe retry up count	3
Probe timeout	5 seconds

Configuring ITD

Enabling ITD

Before you can access the ITD commands, you must enable the ITD feature.

Before you begin

Ensure that you have installed the Network Services license.

Ensure that policy-based routing (PBR) is enabled.

SUMMARY STEPS

1. **configure terminal**
2. [**no**] **feature itd**
3. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **feature itd** **Example:** `switch(config)# feature itd`	Enables the ITD feature. By default, ITD is disabled.
Step 3	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring a Device Group

You can create an ITD device group and then specify the group's nodes and probe. Beginning with Cisco NX-OS Release 7.0(3)I3(1), you can configure multiple device groups.

Before you begin

Ensure that the ITD feature is enabled.

If your device is running Cisco NX-OS Release 7.0(3)I3(1) or later, ensure that the following commands are configured: **feature sla sender** and **feature sla responder**.

SUMMARY STEPS

1. **configure terminal**
2. [**no**] **itd device-group** *name*
3. [**no**] **node {ip | ipv6} {***ipv4-address* | *ipv6-address***}**
4. [**no**] **probe** *track id*
5. [**no**] **weight** *weight*
6. [**no**] **port** *port value*
7. [**no**] **mode hot-standby**
8. [**no**] **shutdown**
9. **exit**
10. Repeat Steps 3 through 5 for each node.
11. [**no**] **probe** {**icmp** | **http** | **tcp port** *port-number* | **udp port** *port-number* | **dns** [**frequency** *seconds*] [[**retry-down-count** | **retry-up-count**] *number*] [**timeout** *seconds*]

12. [no] **hold-down threshold count <count> [time <time>]**
13. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **itd device-group** *name*
	Example:
	`switch(config)# itd device-group dg1`
	`switch(config-device-group)#`
		Creates an ITD device group and enters device group configuration mode. You can enter up to 32 alphanumeric characters.

Step 3	[**no**] **node {ip | ipv6} {***ipv4-address* | *ipv6-address***}**
	Example:
	`switch(config-device-group)# node ip 20.20.20.3`
	`switch(config-dg-node)#`
	Example:
	`switch(config-device-group)# node ipv6`
	`2001::198:1:1:11`
	`switch(config-dg-node)#`
		Specifies the nodes for ITD.

Step 4	[**no**] **probe** *track id*
	Example:
	`switch (config-device-group)# probe track 30`
	`switch(config-device-group-node)#`
		Configures the user defined track ID for the probe.

Step 5	[**no**] **weight** *weight* **Example:** `switch(config-dg-node)# weight 6`	Specifies the weight of the node for ITD. The range is from 1 to 256.

Step 6	[**no**] **port** *port value*
	Example:
	`switch(config-dg-node)# node ip 10.10.10.10`
	`    port 1000`
		Specifies the port number for Feature Port Address Translation . The range is from 1 to 65535.

Step 7	[**no**] **mode hot-standby**
	Example:
	`switch (config-device-group)# node ipv6 50::1`
	`switch(config-device-group-node)# mode hot-standby`
		Configures the node as a hot-standby node for the device group.

Step 8	[**no**] **shutdown** **Example:**	Moves the node into or out of maintenance mode.

	Command or Action	Purpose

	`switch(config-dg-node)# node ip 2.1.1.1`
	`switch(config-dg-node)# shutdown`
	`switch(config-dg-node)# no shutdown`
	`switch(config-dg-node)#`	

Step 9	**exit**
	Example:
	`switch(config-dg-node)# exit`
	`switch(config-device-group)#`
		Exits device group node configuration mode.

Step 10	Repeat Steps 3 through 5 for each node.	
Step 11	[**no**] **probe** {**icmp** | **http** | **tcp port** *port-number* | **udp port** *port-number* | **dns** [**frequency** *seconds*] [[**retry-down-count** | **retry-up-count**] *number*] [**timeout** *seconds*] **Example:** `switch(config-device-group)# probe icmp frequency 100`	Configures the cluster group service probe. Beginning with Cisco NX-OS Release 7.0(3)I3(1), you can specify the following protocols as the probe for the ITD service: • ICMP • TCP • UDP • HTTP • DNS In earlier releases, ICMP is used as the probe for the ITD service. The options are as follows: • **frequency**—Specifies the frequency of the probe in seconds. The range is from 1 to 604800. • **retry-down-count**—Specifies the number of recounts undertaken by the probe when the node goes down. The range is from 1 to 5. • **retry-up-count**—Specifies the number of recounts undertaken by the probe when the node comes back up. The range is from 1 to 5. • **timeout**—Specifies the length of the timeout period in seconds. The range is from 1 to 604800.

Step 12	[**no**] **hold-down threshold count <count> [time <time>]**
	Example:
	`switch(config-itd)# itd device-group dg`
	`switch(config-device-group)# hold-down threshold count 1`
	`switch(config-device-group)# node ip 1.1.1.1`
	`switch(config-dg-node)# hold-down threshold count 3 time 200`
		Specifies the hold-down threshold failure count and threshold timer for the node or the device-group.

	Command or Action	Purpose
Step 13	(Optional) **copy running-config startup-config** **Example:** `switch(config-device-group)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring an ITD Service

Before you begin

Ensure that the ITD feature is enabled.

Ensure that the device group to be added to the ITD service has been configured.

SUMMARY STEPS

1. **configure terminal**
2. [**no**] **itd** *service-name*
3. [**no**] **device-group** *device-group-name*
4. [**no**] **ingress interface** *interface*
5. [**no**] **load-balance** {**method** {**src** {**ip** | **ip-l4port** [**tcp** | **udp**] **range** *x y*} | **dst** {**ip** | **ip-l4port** [**tcp** | **udp**] **range** *x y*}} | **buckets** *bucket-number* | **mask-position** *position*}
6. **virtual [ip | ipv6] {***ipv4-address ipv4-network-mask* | *ipv6-address ipv6-network-mask***}** [**tcp** | **udp** {*port-number* | **any**}] [**advertise** {**enable** | **disable**} [**active**]]
7. Enter one of the following commands to determine how traffic is reassigned after a node failure:
    - [**no**] **failaction node reassign**
    - [**no**] **failaction node least-bucket**
    - [**no**] **failaction bucket distribute**
8. [**no**] **vrf** *vrf-name*
9. [**no**] **exclude access-list** *acl-name*
10. (Optional) [**no**] **peer local service** *peer-service-name*
11. **no shutdown**
12. (Optional) **show itd** [*itd-name*]
13. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **itd** *service-name* **Example:**	Configures an ITD service and enters ITD configuration mode. You can enter up to 32 alphanumeric characters.

	Command or Action	Purpose

	`switch(config)# itd service1`
	`switch(config-itd)#`	

Step 3	[**no**] **device-group** *device-group-name* **Example:** `switch(config-itd)# device-group dg1`	Adds an existing device group to the ITD service. The *device-group-name* specifies the name of the device group. You can enter up to 32 alphanumeric characters. **Note** Beginning with Cisco NX-OS Release 7.0(3)I3(1), you can add multiple device groups to the ITD service.
Step 4	[**no**] **ingress interface** *interface* **Example:** `switch(config-itd)# ingress interface ethernet 4/1-10`	Adds an ingress interface or multiple interfaces to an ITD service. Use a comma (",") to separate multiple interfaces. Use a hyphen ("-") to separate a range of interfaces. Configure the required VRF and interface modes prior to associating the interface to the service.

Step 5	[**no**] **load-balance** {**method** {**src** {**ip** | **ip-l4port** [**tcp** | **udp**] **range** *x y*} | **dst** {**ip** | **ip-l4port** [**tcp** | **udp**] **range** *x y*}} | **buckets** *bucket-number* | **mask-position** *position*}
	Example:
	`switch(config-itd)# load-balance method src ip buckets 16`
		Configures the load-balancing options for the ITD service.
		The options are as follows:
		• **method**—Specifies the source or destination IP-address-based load or traffic distribution.
		• **buckets**—Specifies the number of buckets to create. One or more buckets are mapped to a node. Buckets must be configured in powers of two. The range is from 2 to 256.
		Note If you configure more buckets than the number of nodes, the buckets are applied in a round-robin fashion across all the nodes.
		• **mask-position**—Specifies the load-balance mask position number. The range is from 0 to 23.

Step 6	**virtual [ip | ipv6] {***ipv4-address ipv4-network-mask* | *ipv6-address ipv6-network-mask***}** [**tcp** | **udp** {*port-number* | **any**}] [**advertise** {**enable** | **disable**} [**active**]]
	Example:
	`switch(config-itd)# virtual ip 100.100.100.100 255.255.255.255 udp 443 advertise enable active`
	Example:
	`switch(config-itd)# virtual ipv6 100::100 128 udp 443`
		Configures the virtual IPv4 or IPv6 address of the ITD service.
		The **tcp** and **udp** options specify that the virtual IP address will accept flows from the specified protocol. The port range is from 0 to 65535.
		The **advertise** {**enable** | **disable**} [**active**] option specifies whether the virtual IP route is advertised to neighboring devices.
		Note For IPv6 ITD, the **advertise enable** and the **advertise enable active** options are available in the CLI but they are not supported.

	Command or Action	Purpose

Step 7	Enter one of the following commands to determine how traffic is reassigned after a node failure:
	• [**no**] **failaction node reassign**
	• [**no**] **failaction node least-bucket**
	• [**no**] **failaction bucket distribute**
	Example:
	`switch(config-itd)# failaction node reassign`
	Example:
	`switch(config-itd)# failaction node least-bucket`
	Example:
	`switch(config-itd)# failaction bucket distribute`
		The **failaction node reassign** command assigns failed node traffic to the first available active node. When a node is down, the traffic or bucket associated with the node is reassigned to the first active node found in the configured set of nodes. If the newly reassigned node also fails, the traffic or bucket is reassigned to the next available active node. Once the failed node becomes active again, traffic is diverted back to the new node, and the node resumes serving the connections.
		The **failaction node least-bucket** command assigns the failed node buckets to the active node with the least number of buckets after a node failure. For multiple node failures, this command determines the least-bucketed node and assigns all of the buckets of the failed node to the least-bucketed active node.
		The **failaction bucket distribute** command distributes buckets according to an internal algorithm, which, selects backup nodes with priorities for each bucket, when service is enabled. ITD pre-programs the backup nodes, when service is enabled. In case of node failure, the traffic will be redirect to backup node that is active and with highest priority to minimize the convergence time.
		Note This algorithm is intended for relatively even traffic distribution but doesn't guarantee even distribution.
		Note The **failaction bucket distribute** command is supported for both IPv4 and IPv6.

Step 8	[**no**] **vrf** *vrf-name* **Example:** `switch(config-itd)# vrf RED`	Specifies the VRF for the ITD service.
Step 9	[**no**] **exclude access-list** *acl-name* **Example:** `switch(config-itd)# exclude access-list acl1`	Specifies the traffic that you want ITD to exclude from the ITD load balancer.
Step 10	(Optional) [**no**] **peer local service** *peer-service-name* **Example:** `switch(config-itd)# peer local service service-A`	Specifies one of the two ITD peer services in sandwich mode that are located on the same (local) switch. You must create another ITD service and use this command to specify the second ITD peer service. Once you run this command on both services, the node health status is synchronized across the two services.

	Command or Action	Purpose
		Note The nodes in the two device groups must have the same ordering. Specifically, the first entry in both device groups must be for the same sandwiched mode so that the ordering is preserved.
Step 11	**no shutdown** **Example:** `switch(config-itd)# no shutdown`	Enables the ITD service.
Step 12	(Optional) **show itd** [*itd-name*] **Example:** `switch(config-itd)# show itd`	Displays the status and configuration for specified ITD instances.
Step 13	(Optional) **copy running-config startup-config** **Example:** `switch(config-itd)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Assigning an ACL to an ITD Service

You can use the include access control list (ACL) feature to assign an ACL to an ITD service. For each access control entry (ACE) with the **permit** method in the ACL, this feature filters the unwanted traffic and generates IP access lists and a route map to load-balance the permitted traffic. Load balancing is supported using either the source or destination IP address.

Before you begin

Ensure that the ITD feature is enabled.

Ensure that the device group to be added to the ITD service has been configured.

Ensure that the ACL to be assigned to the ITD service has been configured.

SUMMARY STEPS

1. **configure terminal**
2. [**no**] **itd** *itd-name*
3. [**no**] **device-group** *device-group-name*
4. [**no**] **ingress interface** *interface*
5. [**no**] **load-balance** {**method** {**src** {**ip** | **ip-l4port** [**tcp** | **udp**] **range** *x y*} | **dst** {**ip** | **ip-l4port** [**tcp** | **udp**] **range** *x y*}} | **buckets** *bucket-number*}
6. [**no**] **failaction node-per-bucket**
7. **access-list** *acl-name*
   - For IPv4: **access-list** *acl4-name*
   - For IPv6: **access-list IPv6** *acl6-name*

8. [**no**] **shutdown**
9. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **itd** *itd-name*
	Example:
	`switch(config)# itd service1`
	`switch(config-itd)#`
		Configures an ITD service and enters ITD configuration mode. You can enter up to 32 alphanumeric characters.

Step 3	[**no**] **device-group** *device-group-name* **Example:** `switch(config-itd)# device-group dg1`	Adds an existing device group to the ITD service. The *device-group-name* specifies the name of the device group. You can enter up to 32 alphanumeric characters.
Step 4	[**no**] **ingress interface** *interface* **Example:** `switch(config-itd)# ingress interface ethernet 4/1-10`	Adds an ingress interface or multiple interfaces to an ITD service. Use a comma (",") to separate multiple interfaces. Use a hyphen ("-") to separate a range of interfaces.

Step 5	[**no**] **load-balance** {**method** {**src** {**ip** | **ip-l4port** [**tcp** | **udp**] **range** *x y*} | **dst** {**ip** | **ip-l4port** [**tcp** | **udp**] **range** *x y*}} | **buckets** *bucket-number*}
	Example:
	`switch(config-itd)# load-balance method src ip buckets 16`
		Configures the load-balancing options for the ITD service.
		The options are as follows:
		• **method** —Specifies the source or destination IP-address-based load or traffic distribution.
		• **buckets** —Specifies the number of buckets to create. One or more buckets are mapped to a node. Buckets must be configured in powers of two. The range is from 2 to 256.
		Note If you configure more buckets than the number of nodes, the buckets are applied in a round-robin fashion across all the nodes.

Step 6	[**no**] **failaction node-per-bucket** **Example:** `switch(config-itd)# failaction node-per-bucket`	When a node failure happens the buckets assigned to this node will be distributed across the remaining active nodes. If weights are assigned to nodes, the distribution will be based on weights of the nodes.
Step 7	**access-list** *acl-name* • For IPv4: **access-list** *acl4-name* • For IPv6: **access-list IPv6** *acl6-name*	Assigns an ACL to the ITD service.

	Command or Action	Purpose

	Example:
	IPv4:
	`switch(config-itd)# access-list itd_d`
	Example:
	IPv6:
	`switch(config-itd)# access-list ipv6 itd1_d`	

Step 8	[**no**] **shutdown** **Example:** `switch(config-itd)# no shutdown`	Enables the ITD service.
Step 9	(Optional) **copy running-config startup-config** **Example:** `switch(config-itd)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Nondisruptively Adding or Deleting Nodes

You can configure an ITD session that enables you to add or delete nodes in a device group without shutting down the ITD service. Doing so minimizes traffic disruption, which can occur when you shut down the ITD service.

Before you begin

Ensure that the ITD feature is enabled.

Ensure that the device group and the ITD service have been configured.

SUMMARY STEPS

1. **configure terminal**
2. **itd session device-group** *device-group-name*
3. [**no**] **{node ip | node ipv6}** {*ipv4-address* | *ipv6-address*}
4. (Optional) **probe track id**
5. {**commit** | **abort**}
6. (Optional) **show itd session device-group** [*name*]
7. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

	Command or Action	Purpose

Step 2	**itd session device-group** *device-group-name*
	Example:
	`switch(config)# itd session device-group dg1`
	`switch(config-session-device-group)#`
		Creates an ITD session for the specified device group.

Step 3	[**no**] **{node ip | node ipv6}** {*ipv4-address* | *ipv6-address*}
	Example:
	`switch(config-session-device-group)# node ip 2.2.2.1`
	Example:
	`switch(config-session-device-group)#  node ipv6 10:1::1:2`
		Adds the specified node to the ITD device group. The **no** form of this command deletes the specified node from the ITD device group.
		Repeat this step for each node that you want to add or delete.
		Note The maximum limit on the buckets per node is 32. During an ITD session, when the nodes are deleted (either through normal or nondisruptive commands) and when the number of buckets per node goes over 32 for the remaining active nodes, the following error message is displayed:
		`ERROR: Cannot delete node, exceeding maximum 32 buckets per Node. Shut service to make changes`

Step 4	(Optional) **probe track id**
	Example:
	`switch(config)# itd session device-group dg2`
	`switch(config-session-device-group)#node ip 1.1.1.5`
	`switch(config-session-device-group)#probe track 60`
		Adds a new node with an user-defined track.

Step 5	{**commit** | **abort**}
	Example:
	`switch(config-session-device-group)# commit`
	`switch(config)#`
		The **commit** command updates the ITD device group with the new or modified set of nodes, reassigns buckets, and cleans up the ITD session configuration.
		The **abort** command ignores the ITD session configuration and does not update the ITD device group.
		Note Enter the **commit** command for a nondisruptive session before rebooting. Entering the **copy running-config startup-config** command and rebooting the switch save the ITD device group configuration, but **commit** does not take effect.

Step 6	(Optional) **show itd session device-group** [*name*] **Example:** `switch(config)# show itd session device-group dg1`	Displays all configured ITD sessions or the ITD session for the specified device group.
Step 7	(Optional) **copy running-config startup-config** **Example:**	Copies the running configuration to the startup configuration.

	Command or Action	Purpose
	`switch(config)# copy running-config startup-config`	

Nondisruptively Adding or Deleting ACEs in Include ACLs

You can add or delete the access control entries (ACEs) in the include ACL without shutting down the ITD service. Doing so minimizes traffic disruption, which can occur when you shut down the ITD service.

Before you begin

Ensure that the ITD feature is enabled.

Ensure that the device group and the ITD service have been configured.

Ensure that an ACL has been assigned to the ITD service.

SUMMARY STEPS

1. **configure terminal**
2. **itd session access-list** *acl-name* **refresh**
3. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**itd session access-list** *acl-name* **refresh** **Example:** `switch(config)# itd session access-list test1 refresh`	Internally reads the include ACL and programs the TCAM. ITD checks the old and new ACL ACEs and updates the ITD-generated ACLs. **Note** To update the ACL (for example: add, remove, or update ACEs) during an active IPv6 ITD session, enter the **shutdown** and **no shutdown** commands under the ITD service configuration. The **refresh** option is not supported for IPv6. **Note** This command is required only for include ACLs.
Step 3	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Verifying the ITD Layer-3 Configuration

To display the ITD layer-3 configuration, perform one of the following tasks:

Command	Purpose
show itd [*itd-name*] [**brief** | **vrf** [*vrf-name*]]	Displays the status and configuration for specified ITD instances. • Use the *itd-name* argument to display the status and configuration for a specific ITD instance. • Use the **brief** keyword to display summary status and configuration information. • Use the **vrf** keyword to display the VRFs for the specified ITD instance.
show itd session device-group [*name*]	Displays all of the configured ITD sessions or the ITD session for the specified device group.
show running-config services	Displays the configured ITD device group and services.

These examples show how to verify the ITD configuration:

```
switch# show itd

Name           Probe LB Scheme  Status   Buckets
-------------- ----- ---------- -------- -------
WEB            ICMP  src-ip     ACTIVE   2

Device Group                                         VRF-Name
-------------------------------------------------- -------------
WEB-SERVERS


Pool                           Interface    Status Track_id
------------------------------ ------------ ------ ---------
WEB_itd_pool                   Po-1         UP       -

Virtual IP     Netmask/Prefix           Protocol   Port
---------------------------------------- ---------- -----
10.10.10.100 / 255.255.255.255          IP           0

  Node  IP                  Config-State Weight Status     Track_id
  ------------------------- ------------ ------ ---------- ---------
  1     10.10.10.11         Active       1      OK           -

        Bucket List
        -----------------------------------------------------------------------
        WEB_itd_vip_1_bucket_1


  Node  IP                  Config-State Weight Status     Track_id
  ------------------------- ------------ ------ ---------- ---------
  2     10.10.10.12         Active       1      OK           -
```

```
Bucket List
-----------------------------------------------------------------------
WEB_itd_vip_1_bucket_2
```

Configuration Examples for ITD

This example shows how to configure an ITD device group:

```
switch(config)# feature itd
switch(config)# itd device-group dg
switch(config-device-group)# node ip 210.10.10.11
switch(config-dg-node)# weight 6
switch(config-dg-node)# exit
switch(config-device-group)# node ip 210.10.10.12
switch(config-dg-node)# weight 6
switch(config-dg-node)# exit
switch(config-device-group)# node ip 210.10.10.13
switch(config-dg-node)# weight 2
switch(config-dg-node)# exit
switch(config-device-group)# node ip 210.10.10.14
switch(config-dg-node)# weight 2
switch(config-dg-node)# exit
switch(config-device-group)# probe icmp
```

This example shows how to configure multiple ITD device groups (http_servers and telnet_servers). A virtual IP address is configured per device group, and the load-distribution buckets are per virtual IP address.

```
switch(config)# itd device-group http_servers
  probe icmp
  node ip 10.10.10.9
  node ip 10.10.10.10

switch(config)# itd device-group telnet_servers
  probe icmp
  node ip 1.1.1.1
  node ip 1.1.1.2

switch(config)# itd test
virtual ip 40.1.1.100 255.255.255.255 tcp 23 device-group telnet_servers
virtual ip 30.1.1.100 255.255.255.255 tcp 80 device-group http_servers
  ingress interface Eth3/1
  no shut
```

This example shows how to configure hot-standby nodes for IPv4.

```
switch(config)# feature itd
switch(config)# itd device-group dg4-101
switch(config-device-group)# probe tcp port 8001 frequency 1 timeout 1
switch(config-device-group)# node ip 197.1.1.17
switch(config-dg-node)# node ip 197.1.1.18
switch(config-dg-node)# node ip 197.1.1.47
switch(config-dg-node)# mode hot-standby
switch(config-dg-node)# node ip 197.1.1.48
switch(config-dg-node)# mode hot-standby
```

This example shows how to configure hot-standby nodes for IPv6.

```
switch(config)# feature itd
switch(config)# itd device-group dg6-101
switch(config-device-group)# probe tcp port 8001 frequency 1 timeout 1
switch(config-device-group)# node ipv6 2001::197:1:1:11
switch(config-dg-node)# node ipv6 2001::197:1:1:12
switch(config-dg-node)# node ipv6 2001::197:1:1:2f
switch(config-dg-node)# mode hot-standby
switch(config-dg-node)# node ipv6 2001::197:1:1:30
switch(config-dg-node)# mode hot-standby
```

This example shows how to configure node-level probes (rather than device-group-level probes). With node-level probing, each node can be configured with its own probe, allowing for further customization per node.

```
switch(config)# feature itd
switch(config)# itd device-group Servers
switch(config-device-group)# node ip 192.168.1.10
switch(config-dg-node)# probe icmp frequency 10 retry-down-count 5
switch(config-device-group)# node ip 192.168.1.20
switch(config-dg-node)# probe icmp frequency 5 retry-down-count 5
switch(config-device-group)# node ip 192.168.1.30
switch(config-dg-node)# probe icmp frequency 20 retry-down-count 3
```

This example shows how to configure a virtual IPv4 address:

```
switch(config)# feature itd
switch(config)# itd s4-101
switch(config-itd)# device-group dg_v4
switch(config-device-group)# ingress interface Vlan913
switch(config-device-group)# virtual ip 100.100.100.100 255.255.255.255 udp 443 advertise
enable active
```

This example shows how to configure a virtual IPv6 address:

```
switch(config)# feature itd
switch(config)# itd s6-101
switch(config-itd)# device-group dg_v6
switch(config-device-group)# ingress interface Vlan913
switch(config-device-group)# virtual ipv6 100::100 128 tcp 443
```

This example shows how to configure weighted load balancing to proportionally distribute traffic. In this example, nodes 1 and 2 would get three times as much traffic as nodes 3 and 4.

```
switch(config)# feature itd
switch(config)# itd device-group dg
switch(config-device-group)# probe icmp
switch(config-device-group)# node ip 210.10.10.11
switch(config-dg-node)# weight 3
switch(config-device-group)# node ip 210.10.10.12
switch(config-dg-node)# weight 3
switch(config-device-group)# node ip 210.10.10.13
switch(config-device-group)# node ip 210.10.10.14
```

This example shows how to configure an exclude ACL to specify the traffic that you want ITD to exclude from the ITD load balancer. For example, developer VLANs and test-bed VLANs that do not require firewall inspection can bypass ITD.

```
switch(config)# feature itd
switch(config)# itd Service_Test
switch(config-itd)# device-group test-group
switch(config-itd)# ingress interface vlan10
switch(config-itd)# exclude access-list ITDExclude
switch(config-itd)# no shutdown
switch(config)# ip access-list ITDExclude
switch(config-acl)# 10 permit ip 5.5.5.0/24 any
switch(config-acl)# 20 permit ip 192.168.100.0/24 192.168.200.0/24
```

This example shows how to create acl1 and assign it to an ITD service. The **show** commands display the generated IP access lists and route map.

```
  switch(config)# ip access-list acl1
switch(config-acl)# 2460 permit tcp 100.1.1.0/24 any
switch(config-acl)# exit

switch(config)# itd test
switch(config-itd)# device-group dg1
switch(config-itd)# ingress interface Eth3/1
switch(config-itd)# load-balance method src ip
switch(config-itd)# access-list acl1
switch(config-itd)# show itd test
Legend:
 ST(Status): ST-Standby,LF-Link Failed,PF-Probe Failed,PD-Peer Down,IA-Inactive

Name           LB Scheme  Status   Buckets
-------------- ---------- -------- -------
test           src-ip     ACTIVE   4

Exclude ACL
-------------------------------


Device Group                                       Probe  Port
-------------------------------------------------- ----- ------
dg1                                                 ICMP

Pool                           Interface    Status Track_id
------------------------------ ------------ ------ ---------
test_itd_pool                  Eth3/1       UP       1

ACL Name/SeqNo                   IP/Netmask/Prefix            Protocol Port
-------------------------------- ---------------------------- -------- ----
acl1/2460                        100.1.1.0/24                  TCP      0

  Node  IP            Cfg-S   WGT Probe Port     Probe-IP   STS Trk# Sla_id
  ------------------- ------- --- ---- ----- -------------- --- ---  -------
  1     1.1.1.1       Active   1  ICMP                      OK   2   10002
Bucket List
--------------------------------------------------------------------------
test_itd_ace_1_bucket_1

  Node  IP            Cfg-S   WGT Probe Port     Probe-IP   STS Trk# Sla_id
  ------------------- ------- --- ---- ----- -------------- --- --- -------
  2     1.1.1.2       Active   1  ICMP                      OK   3   10003

Bucket List
--------------------------------------------------------------------------
test_itd_ace_1_bucket_2

  Node  IP            Cfg-S   WGT Probe Port     Probe-IP   STS Trk# Sla_id
  ------------------- ------- --- ---- ----- -------------- --- --- -------
  3     10.10.10.9    Active   1  ICMP                      OK   4   10004

Bucket List
--------------------------------------------------------------------------
test_itd_ace_1_bucket_3
```

```
  Node  IP            Cfg-S   WGT Probe Port     Probe-IP   STS Trk# Sla_id
  ------------------- ------- --- ---- ----- -------------- --- --- -------
  4     10.10.10.10   Active   1  ICMP                      OK   5   10005

Bucket List
--------------------------------------------------------------------------
test_itd_ace_1_bucket_4
```

Beginning with Cisco NX-OS Release 7.0(3)I7(3), ITD supports IPv6. This example shows how to create acl and assign it to an ITDv4 as well as ITDv6 service . The **show** commands display the generated IP access lists and route map.

```
switch(config)# IPv6 access list acl6-101
switch(config-acl)# 10 permit udp 2405:200:1412:2000::/96 any
switch(config-acl)# exit
switch(config)# IP access list acl4-101
switch(config)# 10 permit tcp 10.0.0.0/10 any
switch(config-acl)# exit

switch(config-itd)# device-group dg6-101
switch(config-itd)# ingress interface Vlan913
switch(config-itd)# failaction node reassign
switch(config-itd)# load-balance method src ip
switch(config-itd)#  access-list ipv6 acl6-101
switch(config-itd)# no shut


switch(config-itd)# device-group dg4-101
switch(config-itd)#  ingress interface Vlan913
switch(config-itd)# failaction node reassign
switch(config-itd)# load-balance method src ip
switch(config-itd)#  access-list acl4-101
switch(config-itd)# no shut
```

This example shows how to configure an ITD service to assign failed node buckets to the active node with the least number of buckets after a node failure.

```
switch(config-itd)# show run services

!Command: show running-config services
!Time: Thu Sep 22 22:22:01 2016

version 7.0(3)I5(1)
feature itd


itd session device-group dg


itd device-group dg
  probe icmp
  node ip 1.1.1.1
  node ip 2.2.2.2
  node ip 3.3.3.3


itd test
  device-group dg
  ingress interface Eth1/1
  failaction node least-bucket
  no shut
```

```
switch(config-itd)#

switch(config-itd)# show itd

Legend:
 ST(Status): ST-Standby,LF-Link Failed,PF-Probe Failed,PD-Peer Down,IA-Inactive

Name           LB Scheme  Status   Buckets
-------------- ---------- -------- -------
test           src-ip     ACTIVE   4

Exclude ACL
-------------------------------


Device Group                                       Probe  Port
-------------------------------------------------- ----- ------
dg                                                  ICMP

Pool                           Interface    Status Track_id
------------------------------ ------------ ------ ---------
test_itd_pool                  Eth1/1       UP       1


  Node  IP            Cfg-S   WGT Probe Port     Probe-IP   STS Trk# Sla_id
  ------------------- ------- --- ---- ----- --------------- -- --- -------
  1           1.1.1.1  Active   1 ICMP                       OK   2   10002

      Bucket List
      --------------------------------------------------------------------------
      test_itd_bucket_1, 4

  Node  IP            Cfg-S   WGT Probe Port     Probe-IP   STS Trk# Sla_id
  ------------------- ------- --- ---- ----- --------------- -- --- -------
  2           2.2.2.2  Active   1 ICMP                       OK   3   10003

      Bucket List
      --------------------------------------------------------------------------
      test_itd_bucket_2

  Node  IP            Cfg-S   WGT Probe Port     Probe-IP   STS Trk# Sla_id
  ------------------- ------- --- ---- ----- --------------- -- --- -------
  3           3.3.3.3  Active   1 ICMP                       OK   4   10004

      Bucket List
      --------------------------------------------------------------------------
      test_itd_bucket_3


switch(config-itd)#


# Brought down Node 3, and the failed node buckets are send to Node 2.

switch# show itd

Legend:
 ST(Status): ST-Standby,LF-Link Failed,PF-Probe Failed,PD-Peer Down,IA-Inactive

Name           LB Scheme  Status   Buckets
-------------- ---------- -------- -------
test           src-ip     ACTIVE   4
```

```
Exclude ACL
-------------------------------


Device Group                                       Probe  Port
-------------------------------------------------- ----- ------
dg                                                  ICMP

Pool                           Interface    Status Track_id
------------------------------ ------------ ------ ---------
test_itd_pool                  Eth1/1       UP       1

  Node  IP            Cfg-S   WGT Probe Port     Probe-IP   STS Trk# Sla_id
  ------------------- ------- --- ---- ----- --------------- -- --- -------
  1           1.1.1.1  Active   1 ICMP                       OK   2   10002

      Bucket List
      --------------------------------------------------------------------------
      test_itd_bucket_1, 4

  Node  IP            Cfg-S   WGT Probe Port     Probe-IP   STS Trk# Sla_id
  ------------------- ------- --- ---- ----- --------------- -- --- -------
  2           2.2.2.2  Active   1 ICMP                       OK   3   10003

      Bucket List
      --------------------------------------------------------------------------
      test_itd_bucket_2

  Node  IP            Cfg-S   WGT Probe Port     Probe-IP   STS Trk# Sla_id
  ------------------- ------- --- ---- ----- --------------- -- --- -------
  3           3.3.3.3  Active   1 ICMP                       PF   4   10004

      Bucket List
      --------------------------------------------------------------------------
      test_itd_bucket_3


switch#
switch# conf t
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)# end

switch#
```

This example shows how to configure an ITD service to evenly distribute traffic across all available nodes (rather than to just one active node) after a node failure.

```
switch# show run services

!Command: show running-config services
!Time: Thu Sep 22 22:30:21 2016

version 7.0(3)I5(1)
feature itd


itd session device-group dg


itd device-group dg
  probe icmp
  node ip 1.1.1.1
  node ip 2.2.2.2
  node ip 3.3.3.3
```

```
itd test
  device-group dg
  ingress interface Eth1/1
failaction bucket distribute
  no shut



switch#

switch# show itd
Legend:
 ST(Status): ST-Standby,LF-Link Failed,PF-Probe Failed,PD-Peer Down,IA-Inactive

Name           LB Scheme  Status   Buckets
-------------- ---------- -------- -------
test           src-ip     ACTIVE   4

Exclude ACL
-------------------------------


Device Group                                       Probe  Port
-------------------------------------------------- ----- ------
dg                                                  ICMP

Pool                           Interface    Status Track_id
------------------------------ ------------ ------ ---------
test_itd_pool                  Eth1/1       UP       1


  Node  IP            Cfg-S   WGT Probe Port     Probe-IP   STS Trk# Sla_id
------------------- ------- --- ---- ----- --------------- -- --- -------
  1           1.1.1.1  Active   1 ICMP                      OK   2   10002

      Bucket List
      --------------------------------------------------------------------------
      test_itd_bucket_1, 4

  Node  IP            Cfg-S   WGT Probe Port     Probe-IP   STS Trk# Sla_id
  ------------------- ------- --- ---- ----- --------------- -- --- -------
  2           2.2.2.2  Active   1 ICMP                      OK   3   10003

      Bucket List
      --------------------------------------------------------------------------
      test_itd_bucket_2

  Node  IP            Cfg-S   WGT Probe Port     Probe-IP   STS Trk# Sla_id
  ------------------- ------- --- ---- ----- --------------- -- --- -------
  3           3.3.3.3  Active   1 ICMP                      PF   4   10004

      Bucket List
      --------------------------------------------------------------------------
      test_itd_bucket_3
switch#
```

This example shows how to create an ITD session to nondisruptively add nodes in the dg1 device group:

```
switch(config)# feature itd
switch(config)# itd device-group dg1
switch(config-device-group)#   probe icmp
switch(config-device-group)#   node ip 1.1.1.1
switch(config-dg-node)#   node ip 2.1.1.1
switch(config-dg-node)#   node ip 3.1.1.1
switch(config-dg-node)#
switch(config-dg-node)# itd test
switch(config-itd)#   device-group dg1
switch(config-itd)#   ingress interface Eth1/11
switch(config-itd)#   load-balance method dst ip
Note: Configure buckets equal or more than the total number of nodes.

switch(config-itd)#   access-list acl1
switch(config-itd)#   no shut
switch(config-itd)# show itd test

Legend:
 ST(Status): ST-Standby,LF-Link Failed,PF-Probe Failed,PD-Peer Down,IA-Inactive

Name           LB Scheme  Status   Buckets
-------------- ---------- -------- -------
test           dst-ip     ACTIVE   4

Exclude ACL
-------------------------------


Device Group                                       Probe  Port
-------------------------------------------------- ----- ------
dg1                                                 ICMP

Pool                           Interface    Status Track_id
------------------------------ ------------ ------ ---------
test_itd_pool                  Eth1/11      UP       2

ACL Name
------------------------------
acl1


  Node  IP            Cfg-S   WGT Probe Port     Probe-IP   STS Trk# Sla_id
  ------------------- ------- --- ---- ----- --------------- -- --- -------
  1           1.1.1.1  Active   1 ICMP                       OK   3   10003
Bucket List
      --------------------------------------------------------------------------
      test_itd_bucket_1, 4

  Node  IP            Cfg-S   WGT Probe Port     Probe-IP   STS Trk# Sla_id
  ------------------- ------- --- ---- ----- --------------- -- --- -------
  2           2.1.1.1  Active   1 ICMP                       OK   4   10004

      Bucket List
      --------------------------------------------------------------------------
      test_itd_bucket_2

  Node  IP            Cfg-S   WGT Probe Port     Probe-IP   STS Trk# Sla_id
  ------------------- ------- --- ---- ----- --------------- -- --- -------
  3           3.1.1.1  Active   1 ICMP                       OK   5   10005

      Bucket List
      --------------------------------------------------------------------------
      test_itd_bucket_3

switch(config-itd)# show run service
!Command: show running-config services
!Time: Tue Sep 20 20:36:04 2016
version 7.0(3)I5(1)
feature itd
```

```
itd device-group dg1
  probe icmp
  node ip 1.1.1.1
  node ip 2.1.1.1
  node ip 3.1.1.1
itd test
  device-group dg1
  ingress interface Eth1/11
  load-balance method dst ip
  access-list acl1
  no shut

switch(config-itd)# itd session device-group dg1
switch(config-session-device-group)# node ip 4.1.1.1
switch(config-session-dg-node)# commit
switch(config)# show itd test

Legend:
 ST(Status): ST-Standby,LF-Link Failed,PF-Probe Failed,PD-Peer Down,IA-Inactive

Name           LB Scheme  Status   Buckets
-------------- ---------- -------- -------
test           dst-ip     ACTIVE   4

Exclude ACL
-------------------------------
Device Group                                       Probe  Port
-------------------------------------------------- ----- ------
dg1                                                 ICMP

Pool                           Interface    Status Track_id
------------------------------ ------------ ------ ---------
test_itd_pool                  Eth1/11      UP       2

ACL Name
------------------------------
acl1


  Node  IP            Cfg-S   WGT Probe Port     Probe-IP   STS Trk# Sla_id
  ------------------- ------- --- ---- ----- --------------- -- --- -------
  1           1.1.1.1  Active   1 ICMP                       OK   3   10003

      Bucket List
      --------------------------------------------------------------------------
      test_itd_bucket_1
Node  IP            Cfg-S   WGT Probe Port     Probe-IP   STS Trk# Sla_id
  ------------------- ------- --- ---- ----- --------------- -- --- -------
  2           2.1.1.1  Active   1 ICMP                       OK   4   10004

      Bucket List
      --------------------------------------------------------------------------
      test_itd_bucket_2

  Node  IP            Cfg-S   WGT Probe Port     Probe-IP   STS Trk# Sla_id
  ------------------- ------- --- ---- ----- --------------- -- --- -------
  3           3.1.1.1  Active   1 ICMP                       OK   5   10005

      Bucket List
      --------------------------------------------------------------------------
      test_itd_bucket_3
  Node  IP            Cfg-S   WGT Probe Port     Probe-IP   STS Trk# Sla_id
  ------------------ ------- --- ---- ----- --------------- -- --- -------
  4           4.1.1.1  Active   1 ICMP                       OK   6   10006

      Bucket List
      -----------------------------------------------------------------------
test_itd_bucket_4


switch(config)# show run service

!Command: show running-config services
!Time: Tue Sep 20 20:37:14 2016

version 7.0(3)I5(1)
feature itd


itd device-group dg1
  probe icmp
  node ip 1.1.1.1
node ip 2.1.1.1
  node ip 3.1.1.1
  node ip 4.1.1.1


itd test
  device-group dg1
  ingress interface Eth1/11
  load-balance method dst ip
  access-list acl1
  no shut
```

This example shows how to create an ITD session to nondisruptively delete nodes in the dg1 device group:

```
switch(config)# feature itd
switch(config)#
switch(config)# itd device-group dg1
switch(config-device-group)#   probe icmp
switch(config-device-group)#   node ip 1.1.1.1
switch(config-dg-node)#   node ip 2.1.1.1
switch(config-dg-node)#   node ip 3.1.1.1
switch(config-dg-node)#   node ip 4.1.1.1
switch(config-dg-node)#
switch(config-dg-node)# itd test
switch(config-itd)#   device-group dg1
switch(config-itd)#   ingress interface Eth1/11
switch(config-itd)#   load-balance method dst ip
Note: Configure buckets equal or more than the total number of nodes.

switch(config-itd)#   access-list acl1
switch(config-itd)#   no shut

switch(config-itd)# show itd test

Legend:
 ST(Status): ST-Standby,LF-Link Failed,PF-Probe Failed,PD-Peer Down,IA-Inactive
Name           LB Scheme  Status   Buckets
-------------- ---------- -------- -------
test           dst-ip     ACTIVE   4

Exclude ACL
-------------------------------
```

```
Device Group                                       Probe  Port
-------------------------------------------------- ----- ------
dg1                                                 ICMP

Pool                           Interface    Status Track_id
------------------------------ ------------ ------ ---------
test_itd_pool                  Eth1/11      UP       2

ACL Name
------------------------------
acl1
Node  IP            Cfg-S   WGT Probe Port     Probe-IP   STS Trk# Sla_id
  ------------------- ------- --- ---- ----- --------------- -- --- -------
  1           1.1.1.1  Active   1 ICMP                       OK   3   10003

      Bucket List
      --------------------------------------------------------------------------
      test_itd_bucket_1

  Node  IP            Cfg-S   WGT Probe Port     Probe-IP   STS Trk# Sla_id
  ------------------- ------- --- ---- ----- --------------- -- --- -------
  2           2.1.1.1  Active   1 ICMP                       OK   4   10004

      Bucket List
      --------------------------------------------------------------------------
      test_itd_bucket_2

  Node  IP            Cfg-S   WGT Probe Port     Probe-IP   STS Trk# Sla_id
  ------------------- ------- --- ---- ----- --------------- -- --- -------
  3           3.1.1.1  Active   1 ICMP                       OK   5   10005

      Bucket List
      --------------------------------------------------------------------------
test_itd_bucket_3

  Node  IP            Cfg-S   WGT Probe Port     Probe-IP   STS Trk# Sla_id
  ------------------- ------- --- ---- ----- --------------- -- --- -------
  4           4.1.1.1  Active   1 ICMP                       OK   6   10006

      Bucket List
      --------------------------------------------------------------------------
      test_itd_bucket_4

switch(config-itd)# sh run service

!Command: show running-config services
!Time: Tue Sep 20 20:39:55 2016
version 7.0(3)I5(1)
feature itd


itd device-group dg1
  probe icmp
  node ip 1.1.1.1
  node ip 2.1.1.1
  node ip 3.1.1.1
  node ip 4.1.1.1


itd test
  device-group dg1
  ingress interface Eth1/11
  load-balance method dst ip
  access-list acl1
  no shut

switch(config-itd)# itd session device-group dg1
switch(config-session-device-group)# no node ip 4.1.1.1
switch(config-session-device-group)# commit
switch(config)# show itd test

Legend:
 ST(Status): ST-Standby,LF-Link Failed,PF-Probe Failed,PD-Peer Down,IA-Inactive

Name           LB Scheme  Status   Buckets
-------------- ---------- -------- -------
test           dst-ip     ACTIVE   4

Exclude ACL
-------------------------------


Device Group                                       Probe  Port
-------------------------------------------------- ----- ------
dg1                                                 ICMP

Pool                           Interface    Status Track_id
------------------------------ ------------ ------ ---------
test_itd_pool                  Eth1/11      UP       2

ACL Name
------------------------------
acl1


  Node  IP            Cfg-S   WGT Probe Port     Probe-IP   STS Trk# Sla_id
  ------------------- ------- --- ---- ----- --------------- -- --- -------
  1           1.1.1.1  Active   1 ICMP                       OK   3   10003

      Bucket List
      --------------------------------------------------------------------------
      test_itd_bucket_1

  Node  IP            Cfg-S   WGT Probe Port     Probe-IP   STS Trk# Sla_id
  ------------------- ------- --- ---- ----- --------------- -- --- -------
  2           2.1.1.1  Active   1 ICMP                       OK   4   10004

      Bucket List
      --------------------------------------------------------------------------
      test_itd_bucket_2

  Node  IP            Cfg-S   WGT Probe Port     Probe-IP   STS Trk# Sla_id
  ------------------- ------- --- ---- ----- --------------- -- --- -------
3           3.1.1.1  Active   1 ICMP                       OK   5   10005

      Bucket List
      --------------------------------------------------------------------------
      test_itd_bucket_3, 4


switch(config)# show run service

!Command: show running-config services
!Time: Tue Sep 20 20:41:07 2016

version 7.0(3)I5(1)
feature itd
```

```
itd device-group dg1
  probe icmp
  node ip 1.1.1.1
  node ip 2.1.1.1
  node ip 3.1.1.1


itd test
  device-group dg1
  ingress interface Eth1/11
  load-balance method dst ip
  access-list acl1
  no shut

switch(config)# sh itd test

Legend:
 ST(Status): ST-Standby,LF-Link Failed,PF-Probe Failed,PD-Peer Down,IA-Inactive

Name           LB Scheme  Status   Buckets
-------------- ---------- -------- -------
test           src-ip     ACTIVE   n/a

Source Interface
----------------

Device Group                                       Probe  Port
-------------------------------------------------- ----- ------

Pool                           Interface    Status Track_id
------------------------------ ------------ ------ ---------
                               Eth1/3       UP       1

ACL Name                           Buckets
------------------------------------------------------------------------------
APP1                             8

  Device Group
  ----------------------------------------------------------------------------
  dg1


  Node  IP                  Cluster-id Cfg-S   WGT Probe Port     Probe-IP   STS Trk# Sla_id

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

  1                 1.1.1.3             Active   1 ICMP                       OK   3   10003


      Bucket List
      ---------------------------------------------------------------------------
      test_itd_bucket_2, 1

  Node  IP                  Cluster-id Cfg-S   WGT Probe Port     Probe-IP   STS Trk# Sla_id

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

  2                 1.1.1.4             Active   1 ICMP                       OK   4   10004


      Bucket List
      ---------------------------------------------------------------------------
      test_itd_bucket_3, 6
  Node  IP                 Cluster-id Cfg-S   WGT Probe Port     Probe-IP   STS Trk# Sla_id

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

  3                 1.1.1.5             Active   1 ICMP                      OK   5   10005


      Bucket List
      ---------------------------------------------------------------------------
      test_itd_bucket_4, 5

  Node  IP                 Cluster-id Cfg-S   WGT Probe Port     Probe-IP   STS Trk# Sla_id

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

  4                 1.1.1.2             Active   1 ICMP                      OK   2   10010


      Bucket List
      ---------------------------------------------------------------------------
      test_itd_bucket_8, 7
ACL Name                           Buckets
-------------------------------------------------------------------------------
APP2                             8

  Device Group
  -----------------------------------------------------------------------------
  dg2


  Node  IP                 Cluster-id Cfg-S   WGT Probe Port     Probe-IP   STS Trk# Sla_id

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

  1                 2.1.1.1             Active   1 ICMP                      OK   6   10006


      Bucket List
      ---------------------------------------------------------------------------
      test_itd_acl_1_bucket_1, 6

  Node  IP                 Cluster-id Cfg-S   WGT Probe Port     Probe-IP   STS Trk# Sla_id

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

  2                 2.1.1.2             Active   1 ICMP                      OK   7   10007


      Bucket List
      ---------------------------------------------------------------------------
      test_itd_acl_1_bucket_2, 7

  Node  IP                 Cluster-id Cfg-S   WGT Probe Port     Probe-IP   STS Trk# Sla_id

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

  3                 2.1.1.3             Active   1 ICMP                      OK   8   10008


      Bucket List
      ---------------------------------------------------------------------------
      test_itd_acl_1_bucket_3, 8

  Node  IP                 Cluster-id Cfg-S   WGT Probe Port     Probe-IP   STS Trk# Sla_id
  ------------------------ ---------- ------- --- ---- ----- -------------- -- --- -------
  4                2.1.1.4            Active   1 ICMP                      OK   9   10009

      Bucket List
      --------------------------------------------------------------------------
      test_itd_acl_1_bucket_4, 5

switch(config)# show run services

!Command: show running-config services
!Running configuration last done at: Sun Nov 15 12:09:30 2020
!Time: Sun Nov 15 12:15:10 2020

version 9.4(1) Bios:version N/A
feature itd

itd device-group dg1
  probe icmp frequency 1 timeout 1
  node ip 1.1.1.3
  node ip 1.1.1.4
  node ip 1.1.1.5
  node ip 1.1.1.2

itd device-group dg2
  probe icmp frequency 1 timeout 1
  node ip 2.1.1.1
  node ip 2.1.1.2
  node ip 2.1.1.3
  node ip 2.1.1.4

itd test
  ingress interface Eth1/3
  failaction node least-bucket
  load-balance method src ip
  access-list APP1 device-group dg1
  access-list APP2 device-group dg2
  no shut

switch(config)# itd session device-group dg1
switch(config-session-device-group)# node ip 1.1.1.5
switch(config-session-dg-node)# weight 2
switch(config-session-dg-node)# node ip 1.1.1.4
switch(config-session-dg-node)# weight 3
switch(config-session-dg-node)# node ip 1.1.1.6
switch(config-session-dg-node)# weight 2
switch(config-session-dg-node)# no node ip 1.1.1.2
switch(config-session-device-group)# commit
switch(config)# sh itd test

Legend:
 ST(Status): ST-Standby,LF-Link Failed,PF-Probe Failed,PD-Peer Down,IA-Inactive

Name           LB Scheme  Status   Buckets
-------------- ---------- -------- -------
test           src-ip     ACTIVE   n/a

Source Interface
----------------


Device Group                                       Probe  Port
```

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

Pool                           Interface    Status Track_id
------------------------------ ------------ ------ ---------
                               Eth1/3       UP       1

ACL Name                           Buckets
------------------------------------------------------------------------------
APP1                             8

  Device Group
  ----------------------------------------------------------------------------
  dg1

  Node  IP                 Cluster-id Cfg-S   WGT Probe Port     Probe-IP   STS Trk# Sla_id

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

  1                 1.1.1.3            Active   1 ICMP                       OK   3   10003

      Bucket List
      --------------------------------------------------------------------------
      test_itd_bucket_2

  Node  IP                 Cluster-id Cfg-S   WGT Probe Port     Probe-IP   STS Trk# Sla_id

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

  2                 1.1.1.4            Active   3 ICMP                       OK   4   10004

      Bucket List
      --------------------------------------------------------------------------
      test_itd_bucket_3, 6, 7

  Node  IP                 Cluster-id Cfg-S   WGT Probe Port     Probe-IP   STS Trk# Sla_id

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

  3                 1.1.1.5            Active   2 ICMP                       OK   5   10005

      Bucket List
      --------------------------------------------------------------------------
      test_itd_bucket_4, 5

  Node  IP                 Cluster-id Cfg-S   WGT Probe Port     Probe-IP   STS Trk# Sla_id

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

  4                 1.1.1.6            Active   2 ICMP                       PF  10   10011

      Bucket List
      --------------------------------------------------------------------------
      test_itd_bucket_8, 1
ACL Name                           Buckets
------------------------------------------------------------------------------
APP2                             8

  Device Group
  -----------------------------------------------------------------------------
  dg2


  Node  IP                 Cluster-id Cfg-S   WGT Probe Port     Probe-IP   STS Trk# Sla_id

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

  1                 2.1.1.1            Active   1 ICMP                     OK   6   10006


      Bucket List
      ---------------------------------------------------------------------------
      test_itd_acl_1_bucket_1, 6

  Node  IP                 Cluster-id Cfg-S   WGT Probe Port     Probe-IP   STS Trk# Sla_id

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

  2                 2.1.1.2            Active   1 ICMP                     OK   7   10007


      Bucket List
      ---------------------------------------------------------------------------
      test_itd_acl_1_bucket_2, 7

  Node  IP                 Cluster-id Cfg-S   WGT Probe Port     Probe-IP   STS Trk# Sla_id

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

  3                 2.1.1.3            Active   1 ICMP                     OK   8   10008


      Bucket List
      ---------------------------------------------------------------------------
      test_itd_acl_1_bucket_3, 8

  Node  IP                 Cluster-id Cfg-S   WGT Probe Port     Probe-IP   STS Trk# Sla_id

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

  4                 2.1.1.4            Active   1 ICMP                     OK   9   10009


      Bucket List
      ---------------------------------------------------------------------------
      test_itd_acl_1_bucket_4, 5

switch(config)# sh run services

!Command: show running-config services
!Running configuration last done at: Sun Nov 15 12:17:19 2020
!Time: Sun Nov 15 12:18:16 2020

version 9.4(1) Bios:version N/A
feature itd

itd device-group dg1
  probe icmp frequency 1 timeout 1
  node ip 1.1.1.3
    weight 1
  node ip 1.1.1.4
    weight 3
  node ip 1.1.1.5
```

```
    weight 2
  node ip 1.1.1.6
    weight 2

itd device-group dg2
  probe icmp frequency 1 timeout 1
  node ip 2.1.1.1
  node ip 2.1.1.2
  node ip 2.1.1.3
  node ip 2.1.1.4

itd test
  ingress interface Eth1/3
  failaction node least-bucket
  load-balance method src ip
  access-list APP1 device-group dg1
  access-list APP2 device-group dg2
  no shut
```

This example shows how to nondisruptively add an ACE to an include ACL:

```
switch(config)#
switch(config-acl)# ip access-list acl1
switch(config-acl)# 1010 permit tcp any 10.220.0.0/16
switch(config-acl)# 1020 permit tcp any 20.1.1.0/24

switch(config)# show ip access-lists acl1

IP access list acl1
        1010 permit tcp any 10.220.0.0/16
        1020 permit tcp any 20.1.1.0/24

switch(config)# itd device-group dg1
switch(config-device-group)#   probe icmp
switch(config-device-group)#   node ip 1.1.1.1
switch(config-dg-node)#   node ip 2.1.1.1
switch(config-dg-node)#   node ip 3.1.1.1
switch(config-dg-node)#   node ip 4.1.1.1

switch(config-dg-node)# itd test
switch(config-itd)#   device-group dg1
switch(config-itd)#   ingress interface Eth1/11
switch(config-itd)#   load-balance method dst ip
Note: Configure buckets equal or more than the total number of nodes.

switch(config-itd)#   access-list acl1
switch(config-itd)#   no shut

switch(config)# show run service

!Command: show running-config services
!Time: Tue Sep 20 20:44:17 2016

version 7.0(3)I5(1)
feature itd


itd device-group dg1
  probe icmp
  node ip 1.1.1.1
  node ip 2.1.1.1
  node ip 3.1.1.1
  node ip 4.1.1.1
```

```
itd test
  device-group dg1
ingress interface Eth1/11
  load-balance method dst ip
  access-list acl1
  no shut

switch(config-itd)# ip access-list acl1
switch(config-acl)# 1030 permit tcp any 30.1.1.0/24
switch(config-acl)# exit
switch(config)# itd session access-list acl1 refresh
switch(config)# sh ip access-lists | grep n 4 itd_
IP access list test_itd_bucket_1
        1010 permit tcp any 10.220.0.0 0.0.63.255
        1020 permit tcp any 20.1.1.0 0.0.0.63
        1030 permit tcp any 30.1.1.0/26
IP access list test_itd_bucket_2
        1010 permit tcp any 10.220.64.0 0.0.63.255
        1020 permit tcp any 20.1.1.64 0.0.0.63
        1030 permit tcp any 30.1.1.64/26
IP access list test_itd_bucket_3
        1010 permit tcp any 10.220.128.0 0.0.63.255
        1020 permit tcp any 20.1.1.128 0.0.0.63
1030 permit tcp any 30.1.1.128/26
IP access list test_itd_bucket_4
        1010 permit tcp any 10.220.192.0 0.0.63.255
        1020 permit tcp any 20.1.1.192 0.0.0.63
        1030 permit tcp any 30.1.1.192/26
switch(config)# sh run rpm
interface Ethernet1/11
  ip policy route-map test_itd_pool
```

This example confirms that the access list was generated properly and has the expected ip match condition. Starting from Cisco Nexus Release 9.3(3)F, you can find ACLs in the system using show **ip access-list dynamic** command.

```
Nexus# show ip access-lists CiscoService_itd_vip_1_bucket_1 dynamic

IP access list CiscoService_itd_vip_1_bucket_1
        10 permit ip 1.1.1.0 255.255.255.31 192.168.255.1/32
513E-A-15-C9336C-FX-2-1# show ip access-lists CiscoService_itd_vip_1_bucket_2 dynamic

IP access list CiscoService_itd_vip_1_bucket_2
        10 permit ip 1.1.1.32 255.255.255.31 192.168.255.1/32
513E-A-15-C9336C-FX-2-1# show ip access-lists CiscoService_itd_vip_1_bucket_3 dynamic

IP access list CiscoService_itd_vip_1_bucket_3
        10 permit ip 1.1.1.64 255.255.255.31 192.168.255.1/32
513E-A-15-C9336C-FX-2-1# show ip access-lists CiscoService_itd_vip_1_bucket_4 dynamic

IP access list CiscoService_itd_vip_1_bucket_4
        10 permit ip 1.1.1.96 255.255.255.31 192.168.255.1/32
513E-A-15-C9336C-FX-2-1# show ip access-lists CiscoService_itd_vip_1_bucket_5 dynamic

IP access list CiscoService_itd_vip_1_bucket_5
        10 permit ip 1.1.1.128 255.255.255.31 192.168.255.1/32
513E-A-15-C9336C-FX-2-1# show ip access-lists CiscoService_itd_vip_1_bucket_6 dynamic

IP access list CiscoService_itd_vip_1_bucket_6
        10 permit ip 1.1.1.160 255.255.255.31 192.168.255.1/32
513E-A-15-C9336C-FX-2-1# show ip access-lists CiscoService_itd_vip_1_bucket_7 dynamic

IP access list CiscoService_itd_vip_1_bucket_7
        10 permit ip 1.1.1.192 255.255.255.31 192.168.255.1/32
```

```
513E-A-15-C9336C-FX-2-1# show ip access-lists CiscoService_itd_vip_1_bucket_8 dynamic

IP access list CiscoService_itd_vip_1_bucket_8
        10 permit ip 1.1.1.224 255.255.255.31 192.168.255.1/32
```

This example shows how to nondisruptively delete an ACE from an include ACL:

```
switch(config)# feature itd

switch(config-acl)# ip access-list acl1
switch(config-acl)# 1010 permit tcp any 10.220.0.0/16
switch(config-acl)# 1020 permit tcp any 20.1.1.0/24
switch(config-acl)# 1030 permit tcp any 30.1.1.0/24

switch(config)# itd device-group dg1
switch(config-device-group)#   probe icmp
switch(config-device-group)#   node ip 1.1.1.1
switch(config-dg-node)#   node ip 2.1.1.1
switch(config-dg-node)#   node ip 3.1.1.1
switch(config-dg-node)#   node ip 4.1.1.1
switch(config-dg-node)#
switch(config-dg-node)# itd test
switch(config-itd)#   device-group dg1
switch(config-itd)#   ingress interface Eth1/11
switch(config-itd)#   load-balance method dst ip
Note: Configure buckets equal or more than the total number of nodes.

switch(config-itd)#   access-list acl1
switch(config-itd)#   no shut


switch(config-acl)# sh itd test

Legend:
 ST(Status): ST-Standby,LF-Link Failed,PF-Probe Failed,PD-Peer Down,IA-Inactive

Name           LB Scheme  Status   Buckets
-------------- ---------- -------- -------
test           dst-ip     ACTIVE   4

Exclude ACL
-------------------------------
Device Group                                       Probe  Port
-------------------------------------------------- ----- ------
dg1                                                 ICMP

Pool                           Interface    Status Track_id
------------------------------ ------------ ------ ---------
test_itd_pool                  Eth1/11      UP       2

ACL Name
------------------------------
acl1


  Node  IP            Cfg-S   WGT Probe Port     Probe-IP   STS Trk# Sla_id
  ------------------- ------- --- ---- ----- --------------- -- --- -------
  1          1.1.1.1  Active   1 ICMP                       OK   3   10003

      Bucket List
      --------------------------------------------------------------------------
      test_itd_bucket_1
Node  IP            Cfg-S   WGT Probe Port     Probe-IP   STS Trk# Sla_id
  ------------------- ------- --- ---- ----- --------------- -- --- -------
  2          2.1.1.1  Active   1 ICMP                       OK   4   10004
```

```
      Bucket List
      --------------------------------------------------------------------------
      test_itd_bucket_2

  Node  IP            Cfg-S   WGT Probe Port     Probe-IP   STS Trk# Sla_id
  ------------------- ------- --- ---- ----- --------------- -- --- -------
  3           3.1.1.1  Active   1 ICMP                          OK   5   10005

      Bucket List
      --------------------------------------------------------------------------
      test_itd_bucket_3

  Node  IP            Cfg-S   WGT Probe Port     Probe-IP   STS Trk# Sla_id
  ------------------- ------- --- ---- ----- --------------- -- --- -------
  4           4.1.1.1  Active   1 ICMP                          OK   6   10006

      Bucket List
      --------------------------------------------------------------------------
test_itd_bucket_4

switch(config)# show itd test

Legend:
 ST(Status): ST-Standby,LF-Link Failed,PF-Probe Failed,PD-Peer Down,IA-Inactive
Name           LB Scheme  Status   Buckets
-------------- ---------- -------- -------
test           dst-ip     ACTIVE   4

Exclude ACL
-------------------------------


Device Group                                       Probe  Port
-------------------------------------------------- ----- ------
dg1                                                 ICMP

Pool                           Interface    Status Track_id
------------------------------ ------------ ------ ---------
test_itd_pool                  Eth1/11      UP       2

ACL Name
------------------------------
acl1
Node  IP            Cfg-S   WGT Probe Port     Probe-IP   STS Trk# Sla_id
  ------------------- ------- --- ---- ----- --------------- -- --- -------
  1           1.1.1.1  Active   1 ICMP                          OK   3   10003

      Bucket List
      --------------------------------------------------------------------------
      test_itd_bucket_1

  Node  IP            Cfg-S   WGT Probe Port     Probe-IP   STS Trk# Sla_id
  ------------------- ------- --- ---- ----- --------------- -- --- -------
  2           2.1.1.1  Active   1 ICMP                          OK   4   10004

      Bucket List
      --------------------------------------------------------------------------
      test_itd_bucket_2

  Node  IP            Cfg-S   WGT Probe Port     Probe-IP   STS Trk# Sla_id
  ------------------- ------- --- ---- ----- --------------- -- --- -------
  3           3.1.1.1  Active   1 ICMP                          OK   5   10005
```

```
      Bucket List
      --------------------------------------------------------------------------
test_itd_bucket_3

  Node  IP            Cfg-S   WGT Probe Port     Probe-IP   STS Trk# Sla_id
  ------------------- ------- --- ---- ----- --------------- -- --- -------
  4           4.1.1.1  Active   1 ICMP                       OK   6   10006

      Bucket List
      --------------------------------------------------------------------------
      test_itd_bucket_4

switch(config)# sh run rpm
```

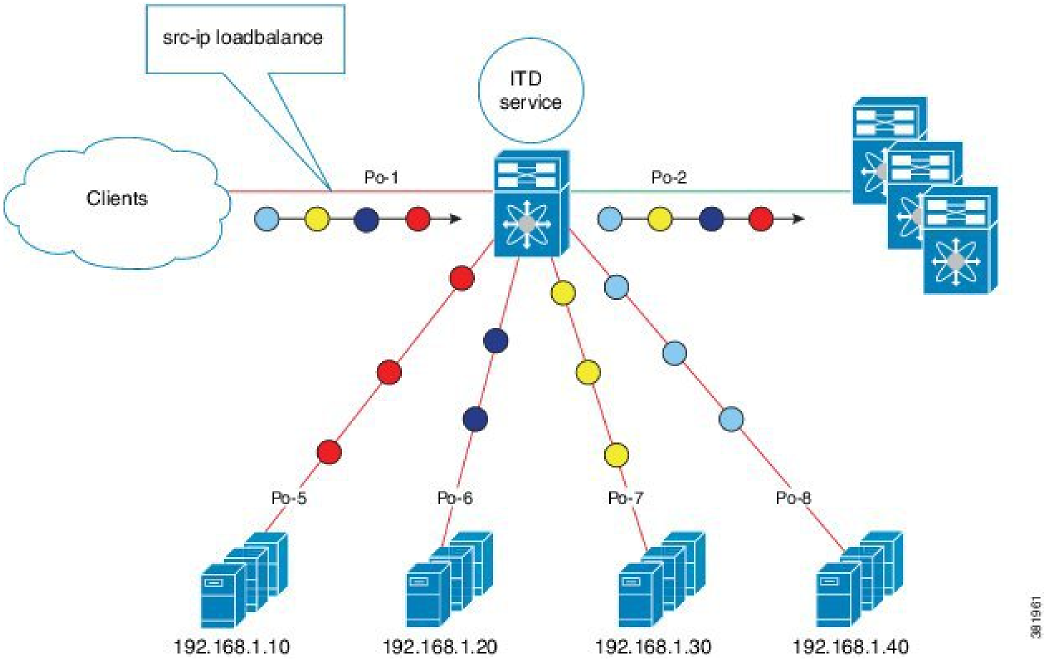
This example shows how to configure ITD node level standby with bucket distribute:

```
itd device-group dg
probe icmp
node ip 10.10.10.2
standby ip 13.13.13.2
node ip 11.11.11.2
standby ip 12.12.12.2
node ip 12.12.12.2
standby ip 11.11.11.2
node ip 13.13.13.2
standby ip 10.10.10.2
itd test
device-group dg
virtual ip 20.20.20.20.255.255.255.255 tcp 80 advertise enable
ingress interface Eth1/9
failaction bucket distribute
load-balance buckets 16
no shut
```

Configuration Example: One-Arm Deployment Mode

The configuration below uses the topology in the following figure:

Figure 6: One-Arm Deployment Mode



Step 1: Define the device group.

```
switch(config)# itd device-group DG
switch(config-device-group)# node ip 210.10.10.11
switch(config-device-group)# node ip 210.10.10.12
switch(config-device-group)# node ip 210.10.10.13
switch(config-device-group)# node ip 210.10.10.14
switch(config-device-group)# probe icmp
```

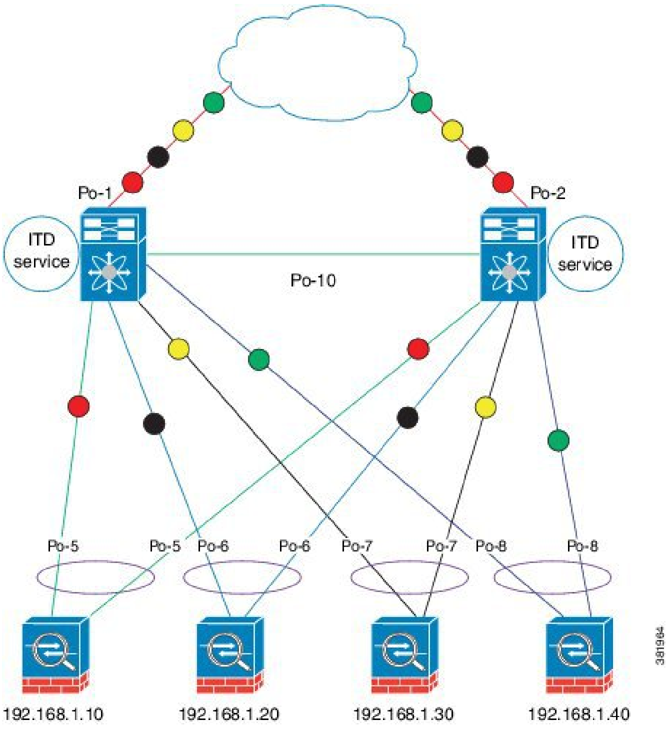
Step 2: Define the ITD service.

```
switch(config)# itd HTTP
switch(config-itd)# ingress interface port-channel 1
switch(config-itd)# device-group DG
switch(config-itd)# no shutdown
```

Configuration Example: One-Arm Deployment Mode with vPC

The configuration below uses the topology in the following figure:

Figure 7: One-Arm Deployment Mode with VPC



Device 1

Step 1: Define the device group.

```
switch(config)# itd device-group DG
switch(config-device-group)# node ip 210.10.10.11
switch(config-device-group)# node ip 210.10.10.12
switch(config-device-group)# node ip 210.10.10.13
switch(config-device-group)# node ip 210.10.10.14
switch(config-device-group)# probe icmp
```

Step 2: Define the ITD service.

```
switch(config)# itd HTTP
switch(config-itd)# ingress interface port-channel 1
switch(config-itd)# device-group DG
switch(config-itd)# no shutdown
```

Device 2

Step 1: Define the device group.

```
switch(config)# itd device-group DG
switch(config-device-group)# node ip 210.10.10.11
switch(config-device-group)# node ip 210.10.10.12
switch(config-device-group)# node ip 210.10.10.13
switch(config-device-group)# node ip 210.10.10.14
switch(config-device-group)# probe icmp
```

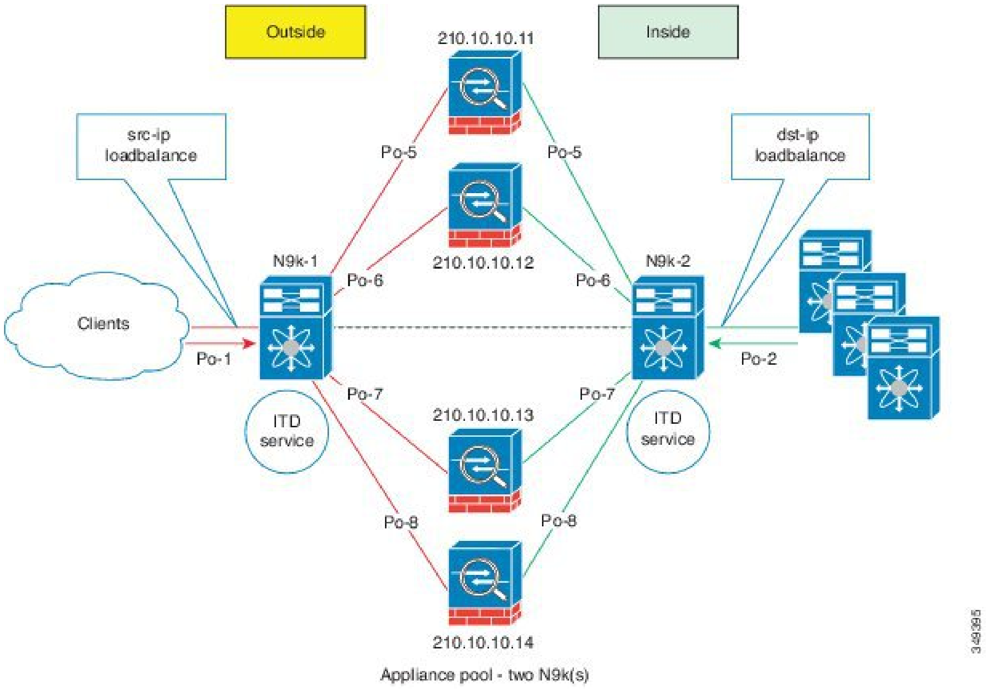
Step 2: Define the ITD service.

```
switch(config)# itd HTTP
switch(config-itd)# ingress interface port-channel 2
switch(config-itd)# device-group DG
switch(config-itd)# no shutdown
```

Configuration Example: Sandwich Deployment Mode

The configuration below uses the topology in the following figure:

Figure 8: Sandwich Deployment Mode



Device 1

Step 1: Define the device group.

```
switch(config)# itd device-group DG
switch(config-device-group)# node ip 210.10.10.11
switch(config-device-group)# node ip 210.10.10.12
switch(config-device-group)# node ip 210.10.10.13
switch(config-device-group)# node ip 210.10.10.14
switch(config-device-group)# probe icmp
```

Step 2: Define the ITD service.

```
switch(config)# itd HTTP
switch(config-itd)# ingress interface port-channel 1
switch(config-itd)# device-group DG
switch(config-itd)# load-balance method src ip
switch(config-itd)# no shutdown
```

Device 2

Step 1: Define the device group.

```
switch(config)# itd device-group DG
switch(config-device-group)# node ip 220.10.10.11
switch(config-device-group)# node ip 220.10.10.12
switch(config-device-group)# node ip 220.10.10.13
switch(config-device-group)# node ip 220.10.10.14
switch(config-device-group)# probe icmp
```

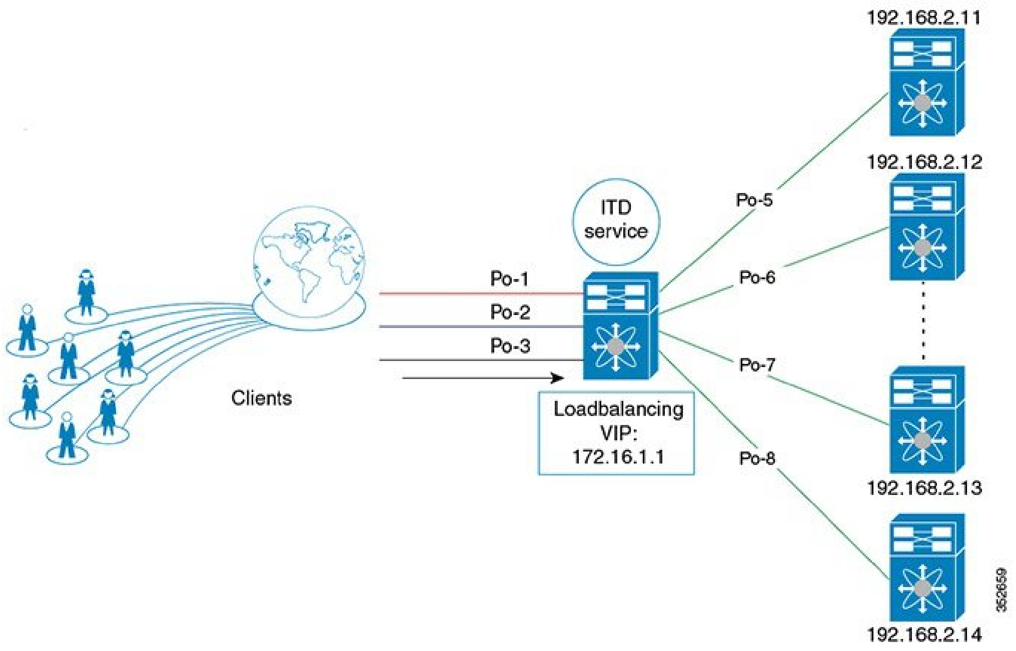
Step 2: Define the ITD service.

```
switch(config)# itd HTTP
switch(config-itd)# ingress interface port-channel 2
switch(config-itd)# device-group DG
switch(config-itd)# load-balance method dst ip
switch(config-itd)# no shutdown
```

Configuration Example: Server Load-Balancing Deployment Mode

The configuration below uses the topology in the following figure:

Figure 9: ITD Load Distribution with VIP



Step 1: Define the device group.

```
switch(config)# itd device-group DG
switch(config-device-group)# node ip 192.168.2.11
switch(config-device-group)# node ip 192.168.2.12
switch(config-device-group)# node ip 192.168.2.13
switch(config-device-group)# node ip 192.168.2.14
switch(config-device-group)# probe icmp
```

Step 2: Define the ITD service.

```
switch(config)# itd HTTP
switch(config-itd)# ingress interface port-channel 1
switch(config-itd)# ingress interface port-channel 2
switch(config-itd)# ingress interface port-channel 3
switch(config-itd)# device-group DG
Switch(config-itd)# virtual ip 172.16.1.1 255.255.255.255
switch(config-itd)# no shutdown
```

Configuration Example: ITD as WCCP Replacement (Web-Proxy Deployment Mode)

A proxy server acts as an intermediary for requests from clients seeking resources from other servers. A web-proxy server specifically operates as an intermediary between a local network and the Internet. Typically, a web-proxy server needs the network device to redirect Internet-bound web traffic toward it (forward flow); however, subsequent packet forwarding only requires the network device to forward the packet regularly.

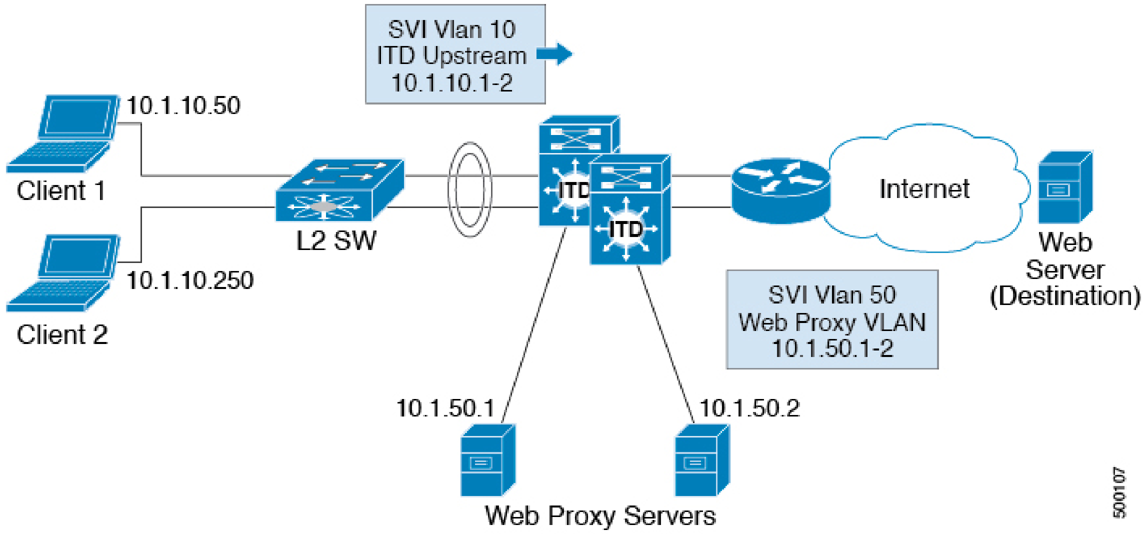
In a web-proxy deployment with ITD, the switch matches the Internet-bound web traffic and load balances it toward the proxy servers. The proxy servers work in an autonomous mode (independent of WCCP and as Active-Active) and handle the traffic that gets redirected to them. The node health probing done through ITD serves the purpose of tracking the state of the nodes and removing or adding them back appropriately based on their availability. Standby servers can also be configured at the group level or node level for redundancy.

ITD redirection is normally only required in the forward direction in the client-facing VLAN. Subsequently, the packets are routed or forwarded without any ITD redirection or distribution. ITD with such web-proxy deployments only need one ITD service, which is configured for the forward direction. However, reverse traffic redirection is required, with traffic selection based on the source Layer 4 ports. Flow symmetry also needs to be maintained by reversing the LB parameter.

With ITD for web-proxy deployments, ITD probes are used to check the availability of the web-proxy server, which is critical because traffic sent toward a failed proxy server is lost.

The configuration below uses the topology in the following figure:

Figure 10: Web-Proxy Deployment Mode



In this example, destination port 80/443 (ingress VLAN 10) to the Internet will be distributed to web-proxy servers 10.1.50.1 and 10.1.50.2. Traffic on VLAN 10 destined to private networks (10.0.0.0/8, 192.168.0.0/16, 172.16.0.0/12) will not be sent to the proxy.

Step 0: Configure an access-list

```
ip access-list ACL1
  10 permit ip any any tcp 80
  20 permit ip any any tcp 443
```

Step 1: Configure the ITD device group web-proxy servers and point to the server IP addresses.

```
itd device-group Web_Proxy_Servers
  probe icmp
  node ip 10.1.50.1
  node ip 10.1.50.2
```

Step 2: Configure an exclude ACL to exclude all traffic destined to private IP addresses.

```
ip access-list itd_exclude_ACL
  10 permit ip any 10.0.0.0/8
  20 permit ip any 192.168.0.0/16
  30 permit ip any 172.16.0.0/12
```

Step 3: Apply the exclude ACL.

```
Itd Web_proxy_SERVICE
  device-group Web_Proxy_Servers
  exclude access-list itd_exclude_ACL
  access-list ACL1
  ingress interface Vlan 10
  failaction node reassign
  load-balance method src ip
  no shutdown
```

If return traffic redirection is also required for any reason, the following additional configuration steps are needed.

Note Only port filtering is possible using the Layer 4 range operator. Also, the exclude ACL supports only permit entries.

Step 4: Configure the return exclude ACL to exclude all but ports 80 and 443.

```
ip access-list itd_exclude_return
  10 permit tcp any range 0 79 any
  20 permit tcp any range 81 442 any
  30 permit tcp any range 444 65535 any
```

Step 5: Configure the return ITD service for the return traffic and apply the exclude ACL.

```
Itd Web_proxy_SERVICE
  device-group Web_Proxy_Servers
  exclude access-list itd_exclude_return
  ingress interface Vlan 20   <- Internet-facing ingress interface on the Nexus switch
  failaction node reassign
  load-balance method dst ip  <- Flow symmetry between forward/return flow achieved by
flipping the LB parameter
  no shutdown
```

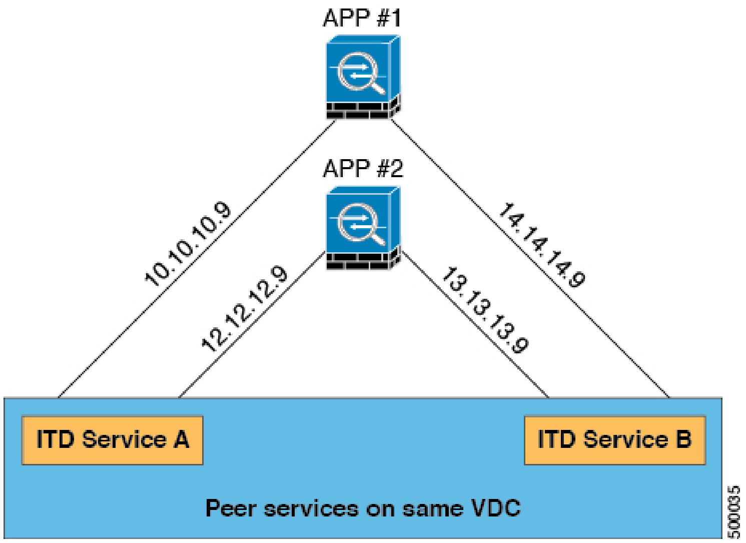
Configuration Example: Peer Synchronization for Sandwich Mode

Whenever the link to a sandwiched appliance on an ITD peer service goes down, the service sends a notification to its peer indicating that the link to the node is down. The peer service then brings the link down so that no traffic traverses that link.

Without peer synchronization, if the link connected to appliance APP #1 on ITD service A goes down in the following topology and ITD service B is not notified, service B will continue to send traffic to APP #1, and the traffic will be dropped.

The configuration below uses this topology:

Figure 11: Peer Synchronization for Sandwich Mode



Device 1

Step 1: Define the device group.

```
switch(config)# itd device-group dev-A
switch(config-device-group)# node ip 10.10.10.9 ---> Link to app #1
switch(config-device-group)# node ip 12.12.12.9 ---> Link to app #2
switch(config-device-group)# probe icmp
```

Step 2: Define the ITD service with peer synchronization enabled.

```
switch(config)# itd service-A
switch(config-itd)# device-group dev-A
switch(config-itd)# ingress interface ethernet 7/4
switch(config-itd)# peer local service service-B
switch(config-itd)# no shutdown

switch(config-itd)# show itd
Name           Probe LB Scheme  Status   Buckets
-------------- ----- ---------- -------- -------
Service-A      ICMP  src-ip     ACTIVE   2

Device Group                                         VRF-Name
-------------------------------------------------- -------------
Dev-A


Route Map                      Interface    Status Track_id
------------------------------ ------------ ------ ---------
Service-A_itd_pool             Eth7/45      UP       3

  Node  IP                  Config-State Weight Status     Track_id  Sla_id
  ------------------------- ------------ ------ ---------- --------- ---------
  1     10.10.10.9          Active       1      Peer Down    1         10001

        IP Access List
        -----------------------------------------------------------------------
        Service-A_itd_bucket_0


  Node  IP                  Config-State Weight Status     Track_id  Sla_id
  ------------------------- ------------ ------ ---------- --------- ---------
  2     12.12.12.9          Active       1      OK           2         10002

        IP Access List
        -----------------------------------------------------------------------
        Service-A_itd_bucket_1
```

Device 2

Step 1: Define the device group.

```
switch(config)# itd device-group dev-B
switch(config-device-group)# node ip 14.14.14.9 ---> Link to app #1
switch(config-device-group)# node ip 13.13.13.9 ---> Link to app #2
switch(config-device-group)# probe icmp
```

Step 2: Define the ITD service with peer synchronization enabled.

```
switch(config)# itd service-B
switch(config-itd)# device-group dev-B
switch(config-itd)# ingress interface ethernet 7/45
switch(config-itd)# peer local service service-A
switch(config-itd)# no shutdown

switch(config-itd)# show itd
Name           Probe LB Scheme  Status   Buckets
-------------- ----- ---------- -------- -------
Service-B      ICMP  src-ip     ACTIVE   2

Device Group                                         VRF-Name
-------------------------------------------------- -------------
Dev-B


Route Map                      Interface    Status Track_id
------------------------------ ------------ ------ ---------
Service-B_itd_pool             Eth7/45      UP       3

  Node  IP                  Config-State Weight Status       Track_id  Sla_id
  ------------------------- ------------ ------ ----------   --------- ---------
  1     14.14.14.9          Active       1      Probe Failed    3        10003

        IP Access List
        -----------------------------------------------------------------------
        Service-B_itd_bucket_0


  Node  IP                  Config-State Weight Status     Track_id  Sla_id
  ------------------------- ------------ ------ ---------- --------- ---------
  2     13.13.13.9          Active       1      OK           4         10004

        IP Access List
        -----------------------------------------------------------------------
        Service-B_itd_bucket_1
```

Configuration Example: Firewall on a Stick

ITD Services

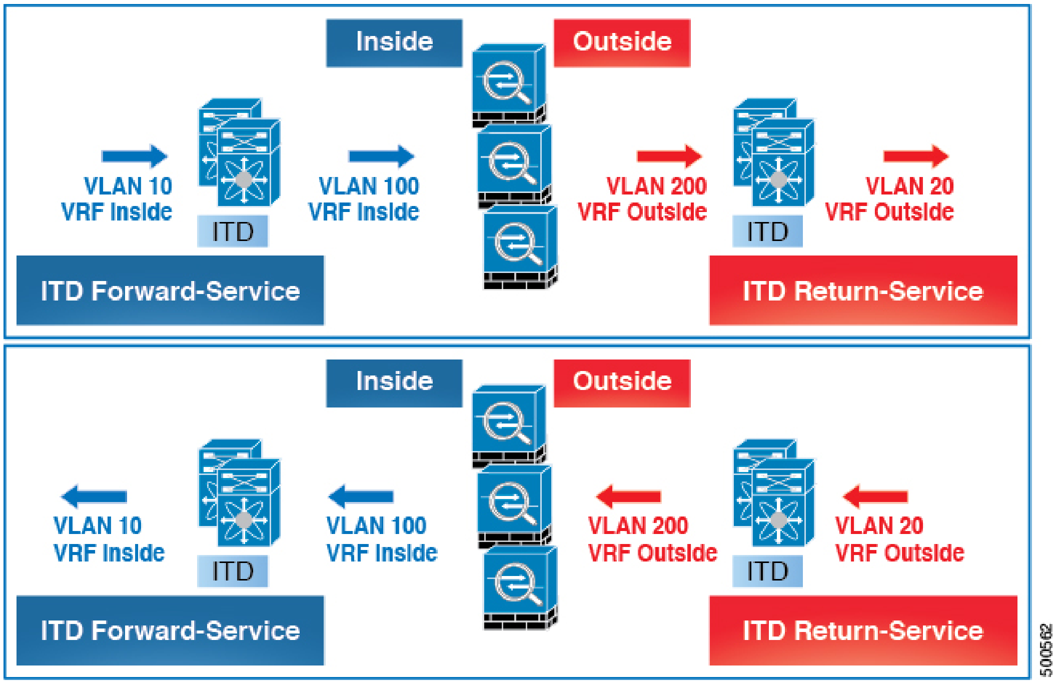
An ITD service configuration defines the ITD traffic distribution for a particular direction of the traffic flow. If both directions of a flow need to be redirected, two ITD services need to be configured, one for the forward traffic flow and one for the return traffic flow. Because an ASA has different inside and outside interface IP addresses, two different device groups also need to be configured to point to the corresponding inside and outside IP addresses.

ASA VLANs

The ITD forward and return services are attached to the inside and outside VLAN SVIs on the Nexus switch. Because a security application such as a firewall needs to examine all traffic, no traffic filtering is configured on the services. As a result, any traffic that hits the SVI is redirected to the corresponding ASA interfaces.

If the ASA interfaces are configured on the same VLANs as that of the switch, the traffic going to the switch from the firewall is redirected to the ASA due to the presence of an ITD service on another VLAN on the switch. Therefore, a pair of separate VLANs is required to prevent traffic looping between the firewalls and the Nexus switch.

Figure 12: ITD ASA Deployment



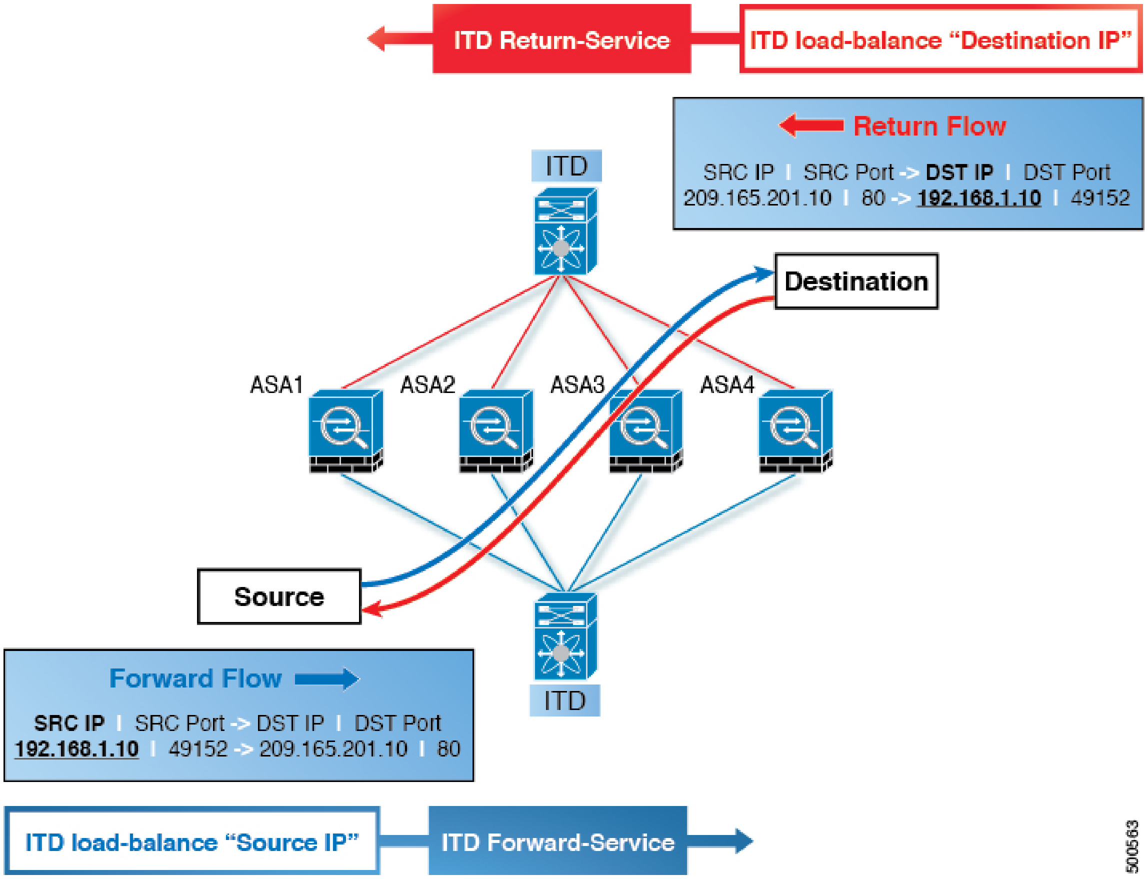
This diagram shows VLANs 10 and 20 as the inside and outside interfaces toward the source and destination on the network. VLANs 100 and 200 are used toward the ASAs to ensure loop-free traffic.

Flow Symmetry

Firewalls typically inspect traffic flows in both the forward and return directions. Due to the stateful nature of the inspection, it is generally required that flow symmetry be maintained during normal operation of firewalls that are not clustered. Even for clustered firewalls, the asymmetry of traffic flows results in the increased redirection of flows over cluster control links. The increase of asymmetric flows adds unnecessary overhead to the firewalls and adversely impedes performance.

Flow symmetry can be achieved using the inherent IP persistence and deterministic nature of the ITD algorithms. A typical ITD configuration for firewalls uses one ITD service for the forward flow and one ITD service for the return flow. Configuring these two ITD services in such a way that the value of the load-balance parameter remains the same for both services ensures that flow symmetry is maintained.

Figure 13: Flow Symmetry in ITD ASA Deployment

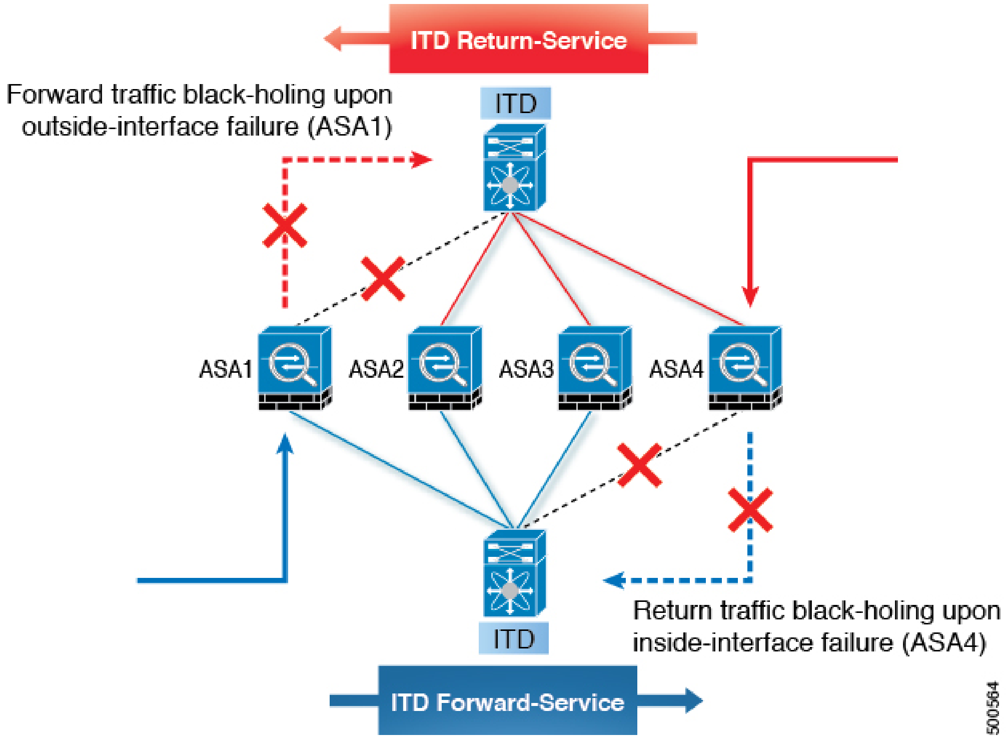


This diagram shows how the source IP address of the forward flow and the destination IP address of the reverse flow remain constant. Choosing the appropriate parameter for the each ITD service ensures flow symmetry due to ITD IP persistence.

Link Failures

When the ASA inside or outside interface fails, the traffic coming into the other side of that ASA can be lost because the egress interface for traffic is down. The ITD peer switch node state synchronization feature can resolve this issue by removing the remote side of the ASA from ITD and synchronizing the node states across the switches.

Figure 14: ASA Failure Scenario

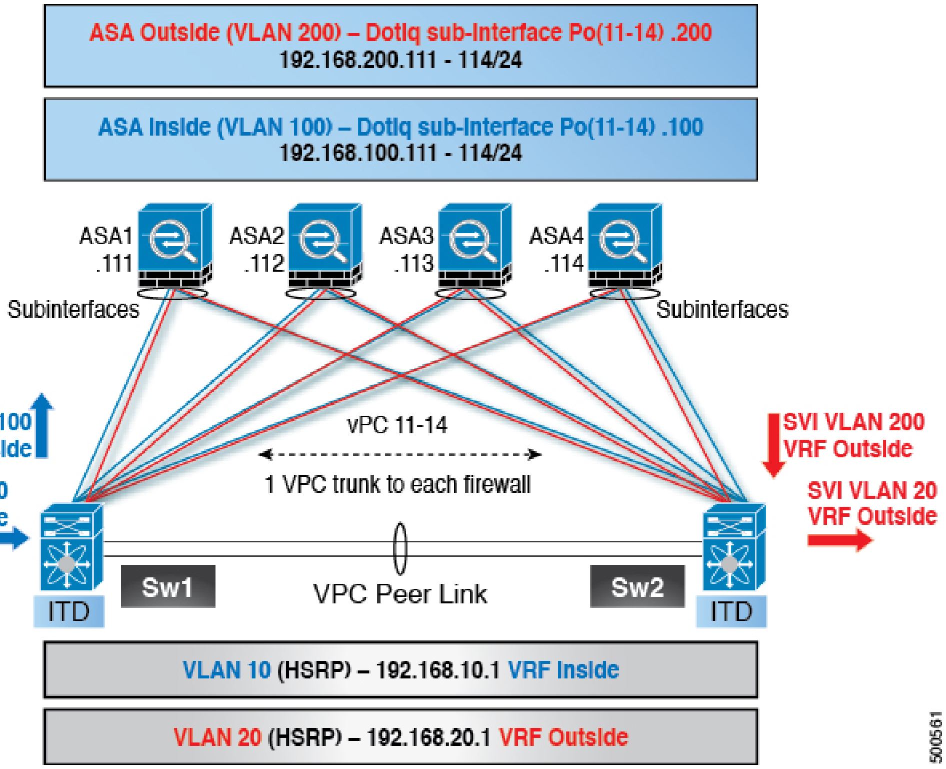

ITD Return-Service
Forward traffic black-holing upon outside-interface failure (ASA1)
ITD
ASA1
ASA2
ASA3
ASA4
ITD
Return traffic black-holing upon inside-interface failure (ASA4)
ITD Forward-Service
500564

The ITD peer switch node state synchronization feature is supported only in a dual-switch non-vPC (or single switch) topology. ASA clustering also solves this problem because clustering ensures that the ASA is fully brought down in the case of such failures. The firewall-on-a-stick implementation (single link or vPC) does not address this issue because the ASA inside and outside interfaces belong to the same physical (or virtual) interface.

Configuration Example

In a firewall on a stick deployment, vPC port-channel (or single port) trunks are typically used to connect the ASAs to the switches. In this configuration, the inside and outside interfaces are dot1q subinterfaces (VLAN 100 and 200), and the switches have two VLANs or SVIs each in the inside and outside contexts without physical port separation between them.

Figure 15: Firewall on a Stick (with vPC) Deployment



Step 1: Configure the switch.

Note This example shows a partial configuration of switch Sw1. The configuration needs to be extended appropriately toward all the ASAs similarly. Other features are assumed to be configured already.

```
interface vlan 10
  description Inside_Vlan_to_Network
  vrf member INSIDE
  ip address 192.168.10.10/24
  hsrp 10
    ip address 192.168.10.1

interface vlan 20
  description Outside_Vlan_to_Network
  vrf member OUTSIDE
  ip address 192.168.20.10/24
  hsrp 20
    ip address 192.168.20.1

interface vlan 100
  description Inside_Vlan_to_ASA
  vrf member INSIDE
  ip address 192.168.100.10/24
  hsrp 100
    ip address 192.168.100.1

interface vlan 200
  description Outside_Vlan_to_ASA
  vrf member OUTSIDE
  ip address 192.168.200.10/24
  hsrp 200
    ip address 192.168.200.1

interface port-channel 11
  description VPC_TO_ASA1
  switchport mode trunk
  switchport trunk allowed vlan 100,200
  vpc 11
  no shutdown

interface ethernet 4/25
  description Link_To_ITD-ASA-1
  switchport
  switchport mode trunk
  switchport trunk allowed vlan 100,200
  channel-group 11 mode active
  no shutdown

interface port-channel 41
  description Downstream_vPC_to_network
  switchport mode trunk
  switchport trunk allowed vlan 10,20
  vpc 41
  no shutdown

interface ethernet 5/1-4
  description Downstream_vPC_member
  switchport
  switchport mode trunk
  switchport trunk allowed vlan 10,20
  channel-group 41
  no shutdown


itd device-group FW_INSIDE
    #Config Firewall Inside interfaces as nodes
  node ip 192.168.100.111
  node ip 192.168.100.112
  node ip 192.168.100.113
  node ip 192.168.100.114
probe icmp frequency 5 timeout 5 retry-count 1

itd device-group FW_OUTSIDE
    #Config Firewall Outside interfaces as nodes
  node ip 192.168.200.111
  node ip 192.168.200.112
  node ip 192.168.200.113
  node ip 192.168.200.114
probe icmp frequency 5 timeout 5 retry-count 1

itd INSIDE
  vrf INSIDE
    #applies ITD service to VRF 'INSIDE'
  device-group FW_INSIDE
    #FW inside interfaces attached to service.
  ingress interface vlan 10
    #applies ITD route map to vlan 1101 interface
  failaction node reassign
    #To use the next available Active FW if an FW goes offline
  load-balance method src ip buckets 16
    #distributes traffic into 16 buckets
    #load balances traffic based on Source IP.
    #OUTSIDE service uses Dest IP.
  no shut

itd OUTSIDE
  vrf OUTSIDE
    #applies ITD service to VRF 'OUTSIDE'
  device-group FW_OUTSIDE
  ingress interface vlan 20
  failaction node reassign
  load-balance method dst ip buckets 16
    #load balances traffic based on Dest IP.
    #INSIDE service uses Src IP.
  no shut
```

Step 2: Configure ASA.

```
interface port-channel 11
  nameif aggregate
  security-level 100
  no ip address

interface port-channel 11.100
  description INSIDE
  vlan 100
  nameif inside
  security-level 100
  ip address 192.168.100.111 255.255.255.0

interface port-channel 11.200
  description OUTSIDE
  vlan 200
  nameif outside
  security-level 100
  ip address 192.168.200.111 255.255.255.0

same-security-traffic permit inter-interface

interface TenGigabitEthernet 0/6
  description CONNECTED_TO_SWITCH-A-VPC
  channel-group 11 mode active
  no nameif
  no security-level

interface TenGigabitEthernet 0/7
  description CONNECTED_TO_SWITCH-B-VPC
  channel-group 11 mode active
  no nameif
  no security-level
```

The following points apply to this example topology:

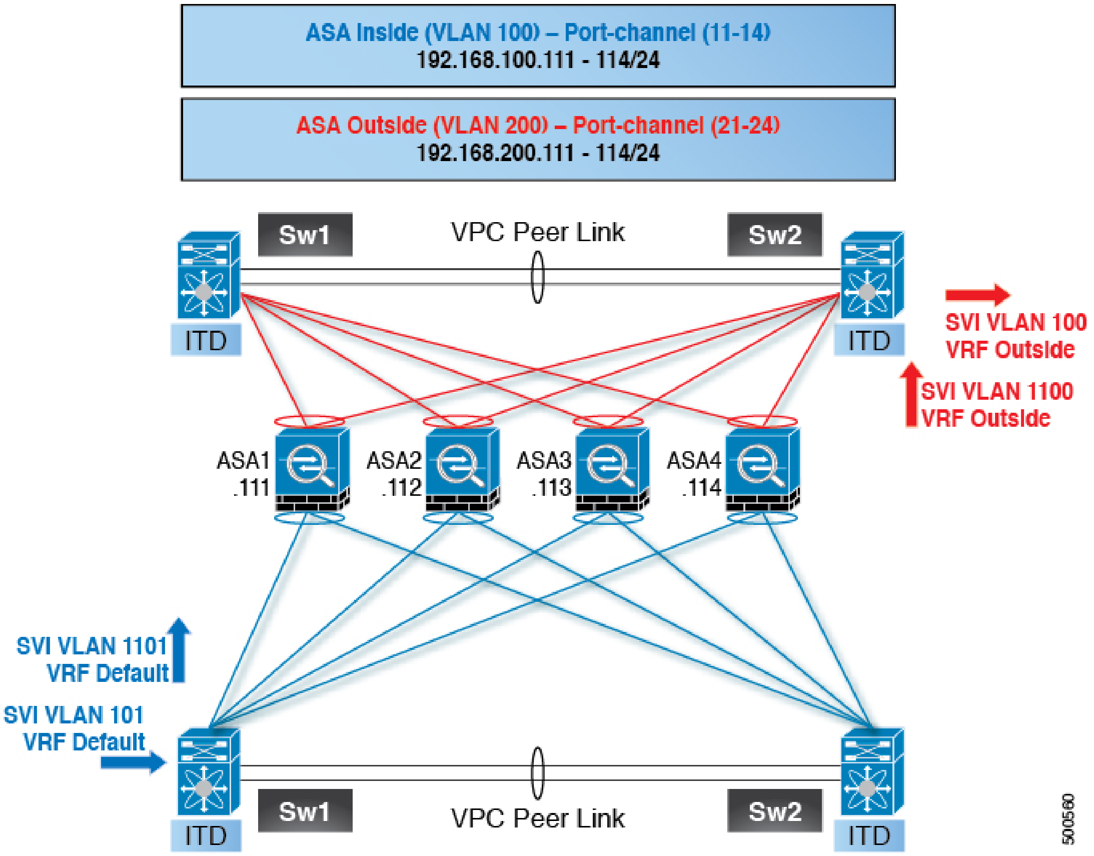
- VLANs 10, 20, 100, and 200 and their SVIs are mapped to appropriate VRFs.
- This example uses an ITD load-balancing configuration to achieve flow symmetry.
- In a vPC scenario, as long as one member of the vPC is up, there is no change to ITD. The ITD redirection on the switch with a failed vPC leg will traverse the peer switch through the peer link as in a typical vPC deployment.
- In this topology, traffic is not lost upon physical link failure because the inside and outside interfaces are tied to the same physical or virtual interface on the ASA (dot1q subinterfaces).

- To support routing protocol neighbors over a vPC, the **layer3 peer-router** command needs to be configured within the vPC domain.
- VRFs are needed because Layer 3 interfaces are used to connect to both inside and outside firewall interfaces. VRFs are put in place to prevent traffic from being (inter-VLAN) routed around the firewall in certain cases.
- Traffic is directed toward ASAs using policy-based routing, so routes are not needed.

Configuration Example: Firewall in Dual-Switch Sandwich Mode with vPCs

For sandwich mode with vPCs, the inside and outside ASA interfaces are each assigned to separate port-channel bundles. As a result of the vPCs, a single link failure does not impede the traffic flow, and ITD will continue to forward through the peer switch's link toward the ASA.

Figure 16: Dual-Switch Sandwich Mode with vPCs



Step 1: Configure the two switches.

```
switch #1:
interface vlan 10
  description INSIDE_VLAN
  ip address 192.168.10.10/24

interface vlan 100
  description FW_INSIDE_VLAN
  ip address 192.168.100.10/24

interface port-channel 11
  description To_ASA-1_INSIDE
  switchport mode access
  switchport access vlan 100
  vpc 11

interface ethernet 4/1
  description To_ASA-1_INSIDE
  switchport mode access
  switchport access vlan 100
  channel-group 11 mode active


switch #2:
interface vlan 20
  description OUTSIDE_VLAN
  ip address 192.168.20.10/24

interface vlan 200
  description FW_OUTSIDE_VLAN
  ip address 192.168.200.10/24

interface port-channel 21
  description To_ASA-1_OUTSIDE
  switchport mode access
  switchport access vlan 200
  vpc 11

interface ethernet 4/25
  description To_ASA-1_OUTSIDE
  switchport mode access
  switchport access vlan 200
  channel-group 21 mode active
```

Step 2: Configure ASA.

```
interface port-channel 11
  description INSIDE
  vlan 100
  nameif inside
  security-level 100
  ip address 192.168.100.111 255.255.255.0

interface port-channel 21
  description OUTSIDE
  vlan 100
  nameif outside
  security-level 100
  ip address 192.168.200.111 255.255.255.0

same-security-traffic permit inter-interface

interface TenGigabitEthernet 0/6
  description CONNECTED_TO_SWITCH-A-VPC
  channel-group 11 mode active
  no nameif
  no security-level

interface TenGigabitEthernet 0/7
  description CONNECTED_TO_SWITCH-B-VPC
  channel-group 11 mode active
  no nameif
  no security-level

interface TenGigabitEthernet 0/8
  description CONNECTED_TO_SWITCH-A-VPC
  channel-group 21 mode active
  no nameif
  no security-level

interface TenGigabitEthernet 0/9
  description CONNECTED_TO_SWITCH-B-VPC
  channel-group 21 mode active
  no nameif
  no security-level
```

The following points apply to this example topology:

- This example uses an ITD load-balancing configuration to achieve flow symmetry.
- In a vPC scenario, as long as one member of the vPC is up, there is no change to ITD. The ITD redirection on the switch with a failed vPC leg will traverse the peer switch through the peer link as in a typical vPC deployment.
- In this topology, traffic loss can occur if one of the port channels on the ASA (or a single physical link in a non-vPC topology) fails.
- To support routing protocol neighbors over a vPC, the **layer3 peer-router** command needs to be configured within the vPC domain.
- Traffic is directed toward ASAs using policy-based routing, so routes are not needed.

Configuration Example: Firewall in Layer 3 Clustering

An ASA cluster consists of multiple ASAs acting as a single unit. Grouping multiple ASAs together as a single logical device provides the convenience of a single device (management and integration into a network) while achieving increased throughput and redundancy of multiple devices.

ITD can load balance to individual mode Layer 3 ASA clusters. ITD is complementary to clustering in that ITD provides the predictability of knowing which flows are handled by each firewall. Instead of relying on OSPF ECMP and port-channel hashing algorithms, you can use ITD buckets to determine these flows.

With Layer 3 clusters, the flow owner can be predetermined based on the bucket allocation. Without ITD and Layer 3 clustering, the initial choice of owner is typically unpredictable. With ITD, the owner can be predetermined.

ASA clustering also uses a backup flow owner. For every flow traversing any particular firewall in the cluster, another firewall stores the state of that flow and the ASA that owns the flow. If the real active flow owner fails, ITD failaction reassign will cause all flows (the bucket) from the failed owner ASA to shift to the next active node listed in the device group. If the new firewall to receive this traffic is not the backup owner for the flows it receives, it should receive the flow state information from the backup owner and process the traffic seamlessly.

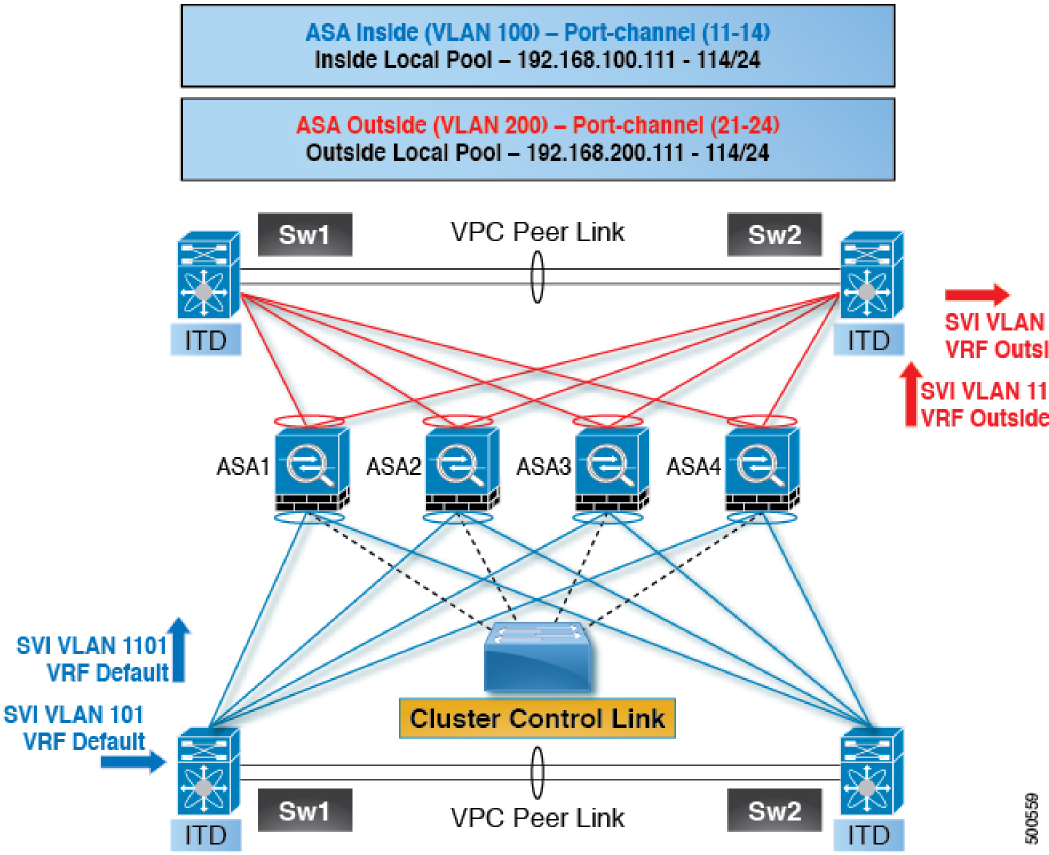
A potential drawback to using ASA clustering with ITD is that backup flows and other cluster table operations consume memory and CPU resources that non-clustered firewalls do not. Therefore, firewall performance might improve when using non-clustered firewalls.

The following table shows a summary comparison of the impact to the cluster control link (CCL) that occurs with ECMP versus ITD when the ASA device status changes.

Table 3: ECMP versus ITD - CCL Impact Summary Comparison

ASA Status	ITD	ECMP
Steady State	Minimal traffic on the CCL and expected traffic types. Exact same load distribution irrespective of the type of line card and switch.	Minimal traffic on the CCL if the same line card type and switch model is used everywhere. If differing hardware is used, a higher level of asymmetry might occur, causing traffic on the CCL network. Each hardware has a different hash function. Two switches (for example, in a vPC) might send the same flow to different ASA devices, causing CCL traffic.
Single ASA Failure	No additional traffic on the CCL. ITD offers IP stickiness and resilient hashing.	All flows are rehashed, and additional traffic redirection occurs on the CCL. Traffic to all ASA devices in the cluster might be affected.
Single ASA Recovery	Traffic redirection can occur on the CCL between two ASA devices in the cluster: the recovered ASA that receives a bucket and the ASA that previously serviced that bucket.	Additional traffic redirection can occur on the CCL. Traffic to all ASA devices in the cluster might be affected.
ASA Addition	Minimal additional traffic on the CCL.	All flows are rehashed, and additional traffic redirection occurs on the CCL. Traffic to all ASA devices in the cluster might be affected.

Figure 17: ASA Cluster with Dual-Switch Sandwich with vPC



Step 1: Configure the two switches.

Note The introduction of clustering does not change the ITD configuration. The ITD configuration depends on the type of topology. In this example, the configuration is the same as in the dual-switch sandwich with vPC topology.

```
switch #1:
interface vlan 10
  description INSIDE_VLAN
  ip address 192.168.10.10/24

interface vlan 100
  description FW_INSIDE_VLAN
  ip address 192.168.100.10/24

interface port-channel 11
  description To_ASA-1_INSIDE
  switchport mode access
  switchport access vlan 100
  vpc 11

interface ethernet 4/1
  description To_ASA-1_INSIDE
  switchport mode access
  switchport access vlan 100
  channel-group 11 mode active


switch #2:
interface vlan 20
  description OUTSIDE_VLAN
  ip address 192.168.20.10/24

interface vlan 200
  description FW_OUTSIDE_VLAN
  ip address 192.168.200.10/24

interface port-channel 21
  description To_ASA-1_OUTSIDE
  switchport mode access
  switchport access vlan 200
  vpc 11

interface ethernet 4/25
  description To_ASA-1_OUTSIDE
  switchport mode access
  switchport access vlan 200
  channel-group 21 mode active
```

Step 2: Configure ASA.

```
cluster group ASA-CLUSTER-L3
  local-unit ASA1
  cluster-interface port-channel 31
  ip address 192.168.250.100 255.255.255.0
  piority 1
  health-check holdtime 1.5
  clacp system-mac auto system-priority 1
  enable

mac-address pool MAC-INSIDE aaaa.0101.0001 - aaaa.0101.0008
mac-address pool MAC-OUTSIDE aaaa.0100.0001 - aaaa.0100.0008
ip local pool IP-OUTSIDE 192.168.200.111-192.168.200.114
ip local pool IP-INSIDE 192.168.100.111-192.168.100.114

interface port-channel 11
  description INSIDE
  lacp max-bundle 8
  mac-address cluster-pool MAC-INSIDE
  nameif inside
  security-level 100
  ip address 192.168.100.11 255.255.255.0 cluster-pool IP-INSIDE

interface port-channel 21
  description OUTSIDE
  lacp max-bundle 8
  mac-address cluster-pool MAC-OUTSIDE
  nameif outside
  security-level 100
  ip address 192.168.200.11 255.255.255.0 cluster-pool IP-OUTSIDE

interface port-channel 31
  description Clustering Interface
  lacp max-bundle 8

interface TenGigabitEthernet 0/6
  channel-group 11 mode active
  no nameif
  no security-level
  no ip address

interface TenGigabitEthernet 0/7
  channel-group 11 mode active
  no nameif
  no security-level
  no ip address

interface TenGigabitEthernet 0/8
  channel-group 21 mode active
  no nameif
  no security-level
  no ip address

interface TenGigabitEthernet 0/9
  channel-group 21 mode active
  no nameif
  no security-level
  no ip address

interface TenGigabitEthernet 1/0
  channel-group 31 mode on
  no nameif
  no security-level
  no ip address

interface TenGigabitEthernet 1/1
  channel-group 31 mode on
  no nameif
  no security-level
  no ip address
```

In this example, port channels 11 and 21 are used for the inside and outside interfaces. Port channel 31 is the clustering interface. Individual interfaces are normal routed interfaces, each with its own IP address taken from a pool of IP addresses. The main cluster IP address is a fixed address for the cluster that always belongs to the current primary unit. Similarly, a MAC address POOL is also configured and used under the corresponding inside or outside port channel.

Related Documents