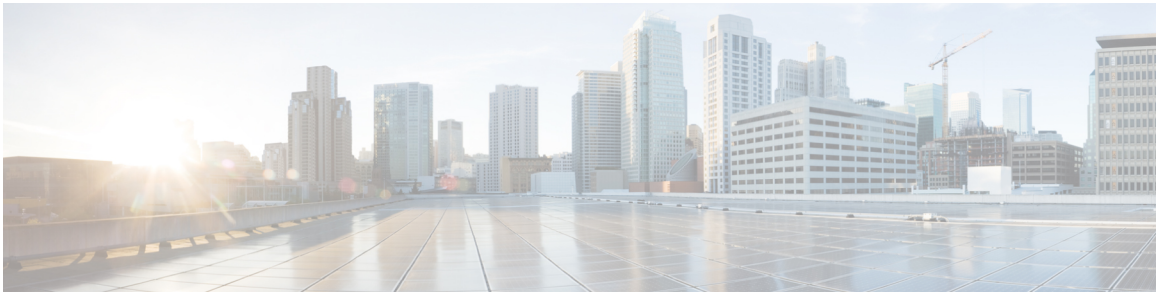
Cisco Nexus 7000 Series NX-OS Intelligent Traffic Director Configuration Guide

First Published: 2016-12-22

Last Modified: 2020-09-22

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



CONTENTS

The documentation set for this product strives to use bias-free language. For purposes of this documentation set, bias-free is defined as language that does not imply discrimination based on age, disability, gender, racial identity, ethnic identity, sexual orientation, socioeconomic status, and intersectionality. Exceptions may be present in the documentation due to language that is hardcoded in the user interfaces of the product software, language used based on standards documentation, or language that is used by a referenced third-party product.

Preface

This preface describes the audience, organization and conventions of the *Cisco Nexus 7000 Series NX-OS Intelligent Traffic Director Configuration Guide*. It also provides information on how to obtain related documentation.

Audience

This publication is for network administrators who configure and maintain Cisco Nexus devices.

Document Conventions

Note As part of our constant endeavor to remodel our documents to meet our customers' requirements, we have modified the manner in which we document configuration tasks. As a result of this, you may find a deviation in the style used to describe these tasks, with the newly included sections of the document following the new format.

Command descriptions use the following conventions:

Convention	Description
bold	Bold text indicates the commands and keywords that you enter literally as shown.
Italic	Italic text indicates arguments for which the user supplies the values.
[x]	Square brackets enclose an optional element (keyword or argument).
[x | y]	Square brackets enclosing keywords or arguments separated by a vertical bar indicate an optional choice.

Convention	Description
{x | y}	Braces enclosing keywords or arguments separated by a vertical bar indicate a required choice.
[x {y | z}]	Nested set of square brackets or braces indicate optional or required choices within optional or required elements. Braces and a vertical bar within square brackets indicate a required choice within an optional element.
`variable`	Indicates a variable for which you supply values, in context where italics cannot be used.
string	A nonquoted set of characters. Do not use quotation marks around the string or the string will include the quotation marks.

Examples use the following conventions:

Convention	Description
`screen font`	Terminal sessions and information the switch displays are in screen font.
`boldface screen font`	Information you must enter is in boldface screen font.
italic screen font	Arguments for which you supply values are in italic screen font.
< >	Nonprinting characters, such as passwords, are in angle brackets.
[ ]	Default responses to system prompts are in square brackets.
!, #	An exclamation point (!) or a pound sign (#) at the beginning of a line of code indicates a comment line.

This document uses the following conventions:

Note Means *reader take note*. Notes contain helpful suggestions or references to material not covered in the manual.

Caution Means *reader be careful*. In this situation, you might do something that could result in equipment damage or loss of data.

Related Documentation for Cisco Nexus 7000 Series NX-OS Software

The entire Cisco Nexus 7000 Series NX-OS documentation set is available at the following URL:

https://www.cisco.com/c/en/us/support/switches/nexus-7000-series-switches/series.html#~tab-documents

Release Notes

The release notes are available at the following URL:

http://www.cisco.com/en/US/products/ps9402/prod_release_notes_list.html

Configuration Guides

These guides are available at the following URL:

http://www.cisco.com/en/US/products/ps9402/products_installation_and_configuration_guides_list.html

The documents in this category include:

- *Cisco Nexus 7000 Series NX-OS Configuration Examples*
- *Cisco Nexus 7000 Series NX-OS FabricPath Configuration Guide*
- *Cisco Nexus 7000 Series NX-OS Fundamentals Configuration Guide*
- *Cisco Nexus 7000 Series NX-OS Interfaces Configuration Guide*
- *Cisco Nexus 7000 Series NX-OS IP SLAs Configuration Guide*
- *Cisco Nexus 7000 Series NX-OS Layer 2 Switching Configuration Guide*
- *Cisco Nexus 7000 Series NX-OS LISP Configuration Guide*
- *Cisco Nexus 7000 Series NX-OS MPLS Configuration Guide*
- *Cisco Nexus 7000 Series NX-OS Multicast Routing Configuration Guide*
- *Cisco Nexus 7000 Series NX-OS OTV Configuration Guide*
- *Cisco Nexus 7000 Series NX-OS Quality of Service Configuration Guide*
- *Cisco Nexus 7000 Series NX-OS SAN Switching Guide*
- *Cisco Nexus 7000 Series NX-OS Security Configuration Guide*
- *Cisco Nexus 7000 Series NX-OS System Management Configuration Guide*
- *Cisco Nexus 7000 Series NX-OS Unicast Routing Configuration Guide*
- *Cisco Nexus 7000 Series NX-OS Verified Scalability Guide*
- *Cisco Nexus 7000 Series NX-OS Virtual Device Context Configuration Guide*
- *Cisco Nexus 7000 Series NX-OS Virtual Device Context Quick Start*
- *Cisco Nexus 7000 Series NX-OS OTV Quick Start Guide*
- *Cisco NX-OS FCoE Configuration Guide for Cisco Nexus 7000 and Cisco MDS 9500*
- *Cisco Nexus 2000 Series Fabric Extender Software Configuration Guide*

Command References

These guides are available at the following URL:

http://www.cisco.com/en/US/products/ps9402/prod_command_reference_list.html

The documents in this category include:

- *Cisco Nexus 7000 Series NX-OS Command Reference Master Index*
- *Cisco Nexus 7000 Series NX-OS FabricPath Command Reference*
- *Cisco Nexus 7000 Series NX-OS Fundamentals Command Reference*
- *Cisco Nexus 7000 Series NX-OS High Availability Command Reference*
- *Cisco Nexus 7000 Series NX-OS Interfaces Command Reference*
- *Cisco Nexus 7000 Series NX-OS Layer 2 Switching Command Reference*
- *Cisco Nexus 7000 Series NX-OS LISP Command Reference*
- *Cisco Nexus 7000 Series NX-OS MPLS Configuration Guide*
- *Cisco Nexus 7000 Series NX-OS Multicast Routing Command Reference*
- *Cisco Nexus 7000 Series NX-OS OTV Command Reference*
- *Cisco Nexus 7000 Series NX-OS Quality of Service Command Reference*
- *Cisco Nexus 7000 Series NX-OS SAN Switching Command Reference*
- *Cisco Nexus 7000 Series NX-OS Security Command Reference*
- *Cisco Nexus 7000 Series NX-OS System Management Command Reference*
- *Cisco Nexus 7000 Series NX-OS Unicast Routing Command Reference*
- *Cisco Nexus 7000 Series NX-OS Virtual Device Context Command Reference*
- *Cisco NX-OS FCoE Command Reference for Cisco Nexus 7000 and Cisco MDS 9500*

Other Software Documents

You can locate these documents starting at the following landing page:

https://www.cisco.com/c/en/us/support/switches/nexus-7000-series-switches/series.html#~tab-documents

- *Cisco Nexus 7000 Series NX-OS MIB Quick Reference*
- *Cisco Nexus 7000 Series NX-OS Software Upgrade and Downgrade Guide*
- *Cisco Nexus 7000 Series NX-OS Troubleshooting Guide*
- *Cisco NX-OS Licensing Guide*
- *Cisco NX-OS System Messages Reference*
- *Cisco NX-OS Interface User Guide*

Documentation Feedback

To provide technical feedback on this document, or to report an error or omission, please send your comments to: .

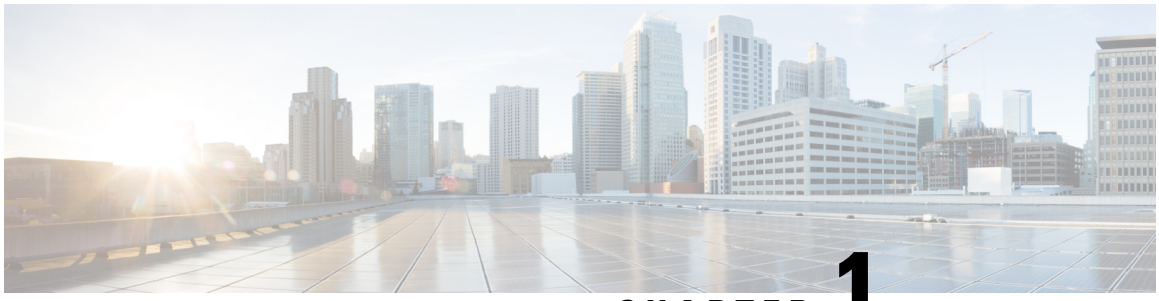
We appreciate your feedback.

Communications, Services, and Additional Information

- To receive timely, relevant information from Cisco, sign up at Cisco Profile Manager.
- To get the business impact you're looking for with the technologies that matter, visit Cisco Services.
- To submit a service request, visit Cisco Support.
- To discover and browse secure, validated enterprise-class apps, products, solutions and services, visit Cisco Marketplace.
- To obtain general networking, training, and certification titles, visit Cisco Press.
- To find warranty information for a specific product or product family, access Cisco Warranty Finder.

Cisco Bug Search Tool

Cisco Bug Search Tool (BST) is a web-based tool that acts as a gateway to the Cisco bug tracking system that maintains a comprehensive list of defects and vulnerabilities in Cisco products and software. BST provides you with detailed defect information about your products and software.



CHAPTER 1

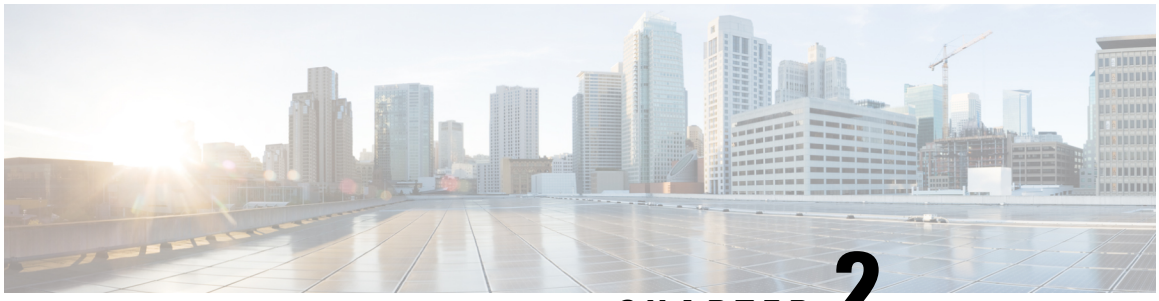
New and Changed Information

New and Changed Information

Your software release might not support all the features in this document. For the latest caveats and feature information, see the Bug Search Tool at https://tools.cisco.com/bugsearch/ and the release notes for your software release.

Feature	Description	Changed in Release	Where documented
Pre-fetch Optimization	The ITD failaction feature is enhanced to optimally pre-fetch the status of the service nodes before reassigning the failed nodes to active nodes.	8.3(1)	Configuring ITD, on page 3
VIP knob	The VIP knob feature allows you to configure a VIP for ITD device group, with route creation based on health of device group node health.	8.2(1)	Server Load-Balancing Deployment Mode , on page 8
HTTP Probe	HTTP probe is supported.	8.2(1)	Monitor Node, on page 13

New and Changed Information



CHAPTER 2

Configuring ITD

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

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*.

Finding Feature Information

Your software release might not support all the features documented in this module. For the latest caveats and feature information, see the Bug Search Tool at https://tools.cisco.com/bugsearch/ and the release notes for your software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the "New and Changed Information"chapter or the Feature History table in this chapter.

Information About ITD

Intelligent Traffic Director (ITD) is an intelligent, scalable clustering and load-balancing engine that addresses the performance gap between a multi-terabit switch and gigabit servers and appliances. The ITD architecture integrates Layer 2 and Layer 3 switching with Layer 4 to Layer 7 applications for scale and capacity expansion to serve high-bandwidth applications.

Note The documentation set for this product strives to use bias-free language. For the purposes of this documentation set, bias-free is defined as language that does not imply discrimination based on age, disability, gender, racial identity, ethnic identity, sexual orientation, socioeconomic status, and intersectionality. Exceptions may be present in the documentation due to language that is hardcoded in the user interfaces of the product software, language used based on RFP documentation, or language that is used by a referenced third-party product.

ITD provides adaptive load balancing to distribute traffic to an application cluster. With this feature on the Cisco Nexus 7000 Series switch, you can deploy servers and appliances from any vendor without a network or topology upgrade.

ITD Feature Overview

Intelligent Traffic Director offers simplicity, flexibility, and scalability. This makes it easier for customers to deploy a traffic distribution solution in a wide variety of use cases without the use of any external hardware. Here are a few common deployment scenarios:

- Firewall cluster optimization
- Predictable redundancy and scaling of security services such as Intrusion Prevention System, Intrusion Detection System and more.
- High-scale DNS solutions for enterprise and service providers
- Scaling specialized web services such as SSL Accelerators, HTTP compression, and others
- Using the data plane of the network to distribute high bandwidth applications

The following example use cases are supported by the Cisco ITD feature:

- Load-balance traffic to 256 servers of 10Gbps each.
- Load-balance to a cluster of Firewalls. ITD is much superior than policy-based routing (PBR).
- Scale up NG IPS and WAF by load-balancing to standalone devices.
- Scale the WAAS / WAE solution.
- Scale the VDS-TC (video-caching) solution.
- Replace ECMP/Port-channel to avoid re-hashing. ITD is resilient.

Benefits of ITD

ITD on the Cisco NX-OS switch enables the following:

High Scalability

- Hardware based multi-terabit scaling for Layer 3 and 4 services and applications load balancing and traffic redirect
- High performance, line-rate 1, 10, 40, and 100 Gigabit Ethernet (GE) traffic distribution connectivity

Operational Simplicity

- Transparent connectivity for appliance and server clustering
- Optimized for fast and simple provisioning

Investment Protection

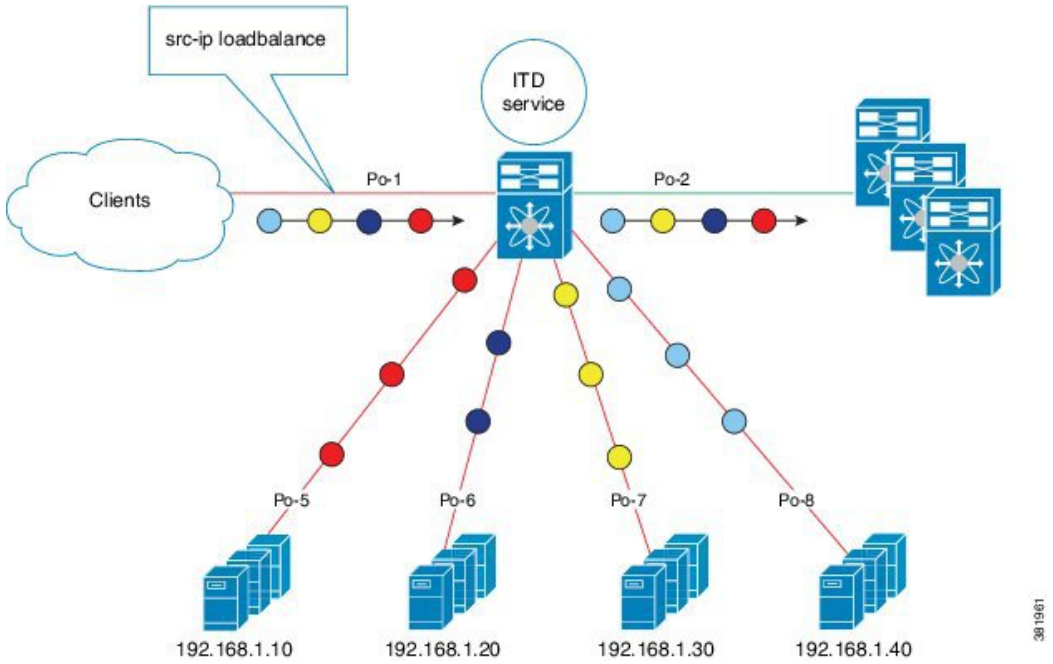
- Supported on all Cisco Nexus 5000, 6000, 7000, and 9000 switching platforms. No new hardware is required.
- End device agnostic. It supports all servers and service appliances.

Deployment Modes

One-Arm Deployment Mode

You can connect servers to the Cisco NX-OS device 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 in a server into the network with no changes to the existing topology or network.

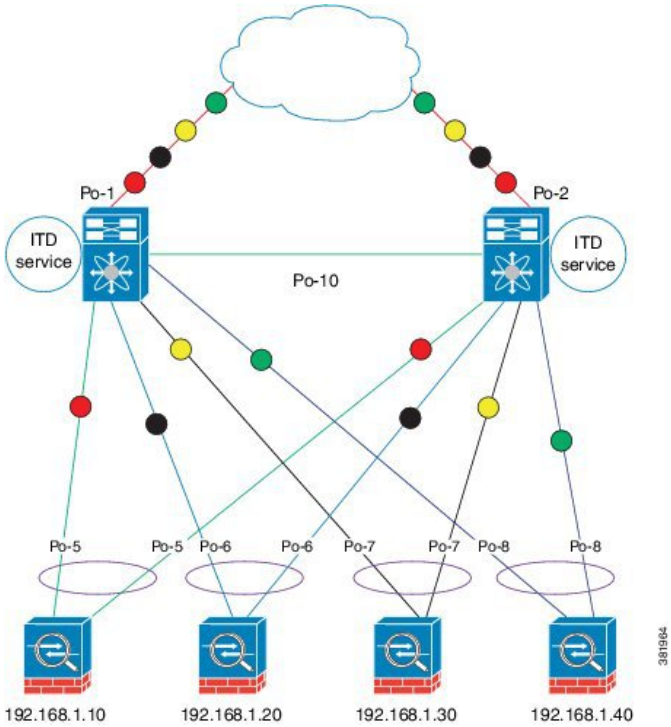
Figure 1: One-Arm Deployment Mode



One-Arm Deployment Mode with VPC

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

Figure 2: One-Arm Deployment Mode with VPC



Sandwich Deployment Mode

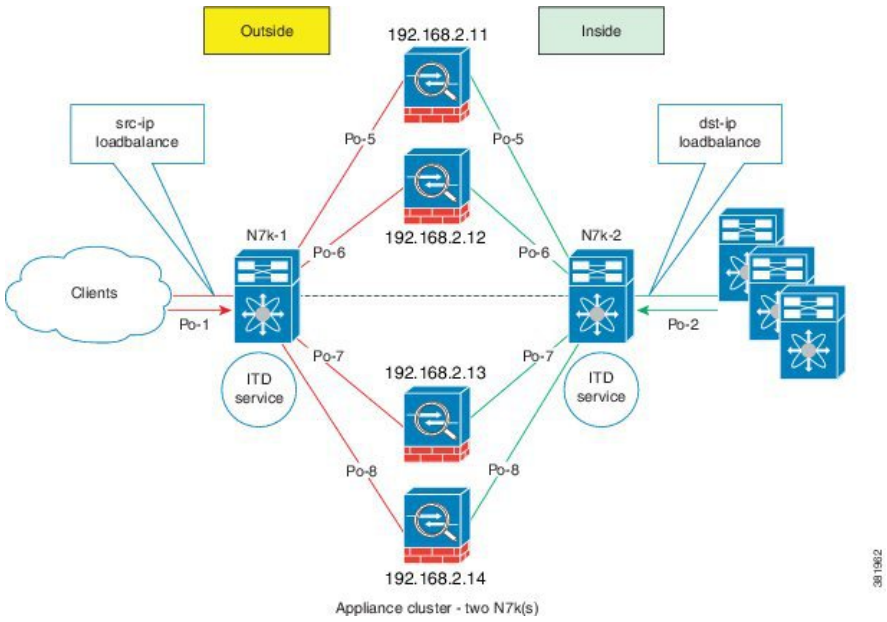
The sandwich deployment mode uses two Cisco NX-OS 7000 Series switches to provide stateful handling of traffic.

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

The key features are:

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

Figure 3: Sandwich Deployment Mode

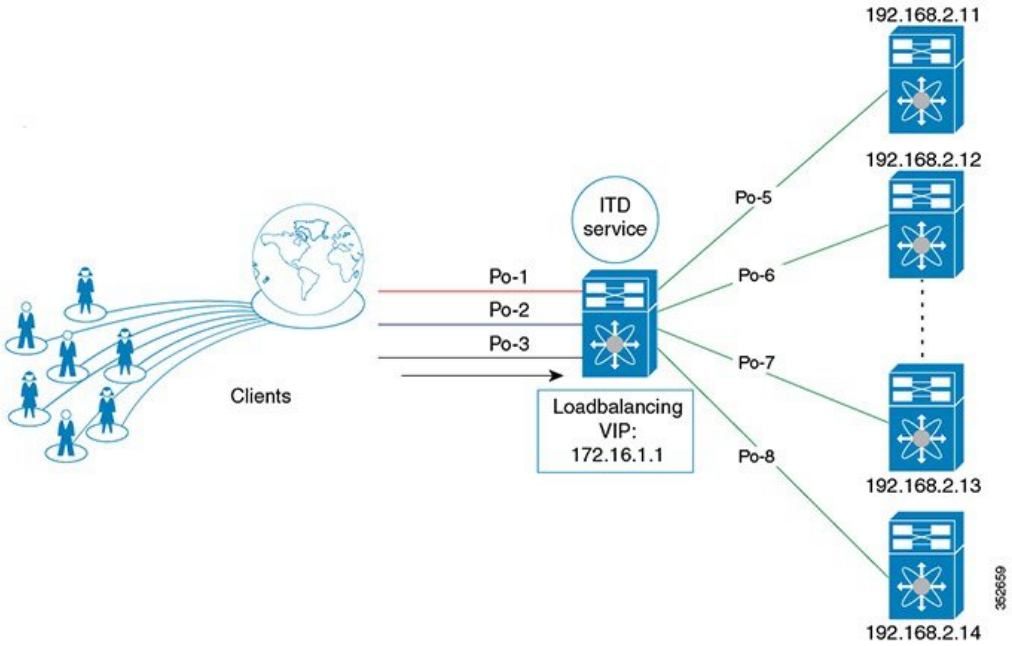


Server Load-Balancing Deployment Mode

The ITD service can be configured to host a virtual IP (VIP) on a Cisco NX-OS 7000 Series switch. Internet traffic destined for the VIP will be load balanced to the active nodes. Unlike traditional server load balancers, source NAT is not needed as the ITD service is not a stateful load balancer.

Note You need to configure ITD service similarly on each Cisco NX-OS 7000 Series switch. The ITD service configuration needs to be done manually on each switch.

Figure 4: ITD Load Distribution with VIP



Attention Configure a single VIP address for an ITD service serving a group of nodes (or device group).

Destination NAT

Network Address Translation (NAT) is a commonly deployed feature in load balancing, firewall, and service appliances. Destination NAT is one of the types of NAT that is used in load balancing.

Benefits of Destination NAT

The following are the benefits of using NAT in ITD deployments:

- Not all the servers in the server pool is required to host the virtual IP address.
- The client, which is not required to be aware of the Server IP, always sends the traffic to the virtual IP address.
- The load balancer detects server failures, and redirects the traffic to the appropriate server, without the client being aware of the status of the primary server.
- NAT provides security by hiding the real server IP from the client.
- NAT provides increased flexibility in moving the real servers across different server pools.

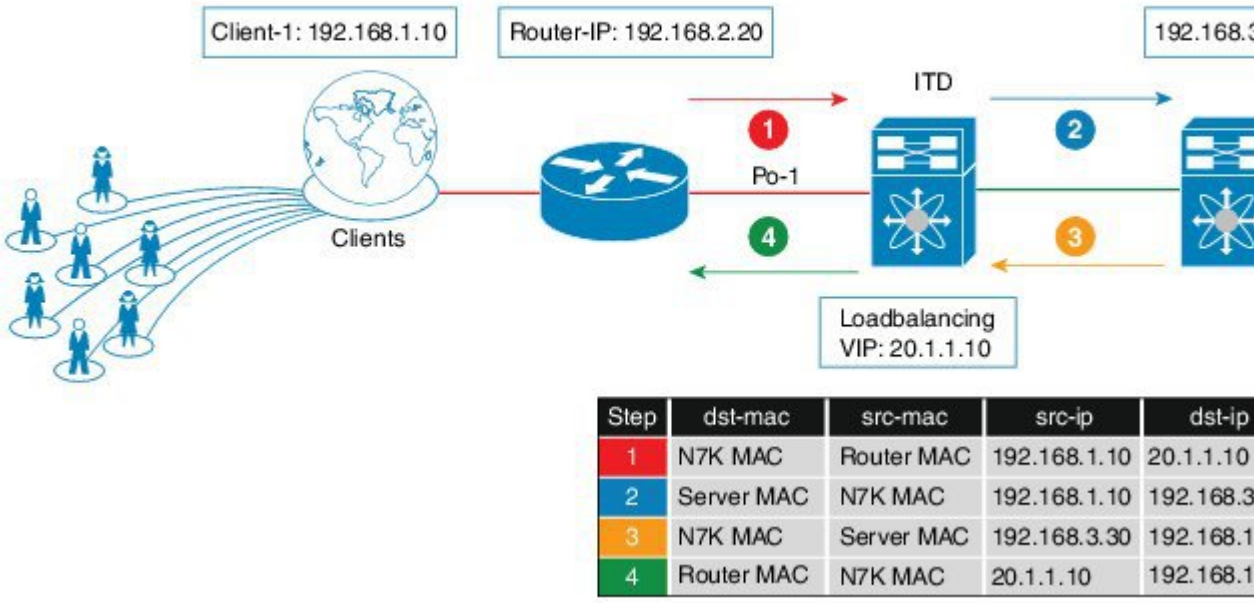
Among the different types of NAT, Destination NAT is deployed commonly in load balancing because of the following advantages it provides:

- The traffic from source or client to the virtual IP address is rewritten and redirected to server.

- The traffic from the source or client to the destination or server, which is the forward path, is handled as follows: the traffic from the source or client to virtual IP address is translated and redirected as the traffic from source to the destination or server.
- The traffic from the destination to the source or client, which is the reverse path, is re-translated with the virtual IP address as the source IP address. That is, the traffic from the server or source to the client or destination is translated as client or source to client or destination.

The following figure illustrates the NAT with Virtual IP Address:

Figure 5: NAT with Virtual IP Address



Step	dst-mac	src-mac	src-ip	dst-ip
1	N7K MAC	Router MAC	192.168.1.10	20.1.1.10
2	Server MAC	N7K MAC	192.168.1.10	192.168.3
3	N7K MAC	Server MAC	192.168.3.30	192.168.1
4	Router MAC	N7K MAC	20.1.1.10	192.168.1

Device Groups

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

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

Multiple Device-Groups within an ITD Service

The feature, by enabling the existence of multiple device-groups per service on the same interface, allows the ITD to scale.

The traffic from one ingress interface is distributed based on both VIPs and device-groups.

An ITD service generates a single route-map that has next hops point to nodes from different device-groups.

Optimized Node Insertion or Removal

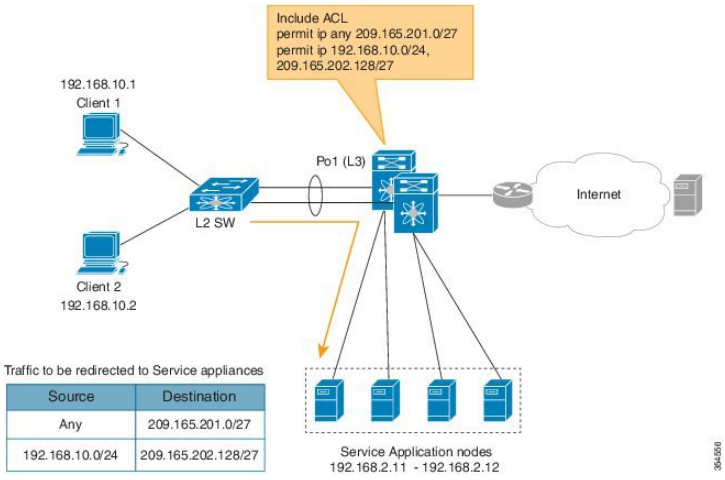
This feature enables users to dynamically add or remove nodes with minimal disruption to existing traffic. ITD now maintains an intermittent state of nodes when the nodes are deleted or added in a service that is active. In addition, ITD automatically re-programs the buckets when the user adds or deletes the node with minimum disruption to service. This feature is supported:

- at Device-group level
- in Virtual IP Address (VIP), and also without VIP
- in multiple VIP device-group feature

Include ACL

The Include ACL feature allows traffic selection for ITD load-balancing by defining the IP addresses to be allowed through ITD. The ACL configured under this feature defines permit ACEs to match traffic for load-balancing. Any unmatched addresses in the ACL will bypass ITD. The Include ACL and Exclude ACL features can be used in conjunction for granular traffic selection within ITD. Both these ACLs can only have permit ACEs but not deny ACEs. In circumstances where service appliances cater only to specific internet traffic, ITD selects the traffic to load-balance or redirect the traffic while the rest of the traffic is routed normally through the RIB.

Figure 6: Include ACL



The Include ACL feature is used to achieve traffic selection and traffic filtering within the ITD. The VIP feature can only match destination fields, whereas the Include ACL feature matches both source and destination fields.

VRF Support

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

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

Load Balancing

The ITD feature enables you to configure specific load-balancing options by using the **loadbalance** command.

The optional keywords for the **loadbalance** command are as follows:

- **buckets**—Specifies the number of buckets to create. Buckets must be configured in powers of two. One or more buckets are mapped to a node in the cluster. If you configure more buckets than the number of nodes, the buckets are applied in round robin fashion across all the nodes.
- **mask-position**— Specifies the mask position of the load balancing. This keyword is useful when a packet classification has to be made based on specific octets or bits of an IP addresses. By default the system uses the last octet's starting most significant bits (MSBs).

  If you prefer to use nondefault bits/octets, you can use the **mask-position** keyword to provide the starting point at which bits the traffic classification is to be made. For example, you can start at the 8th bit for the second octet and the 16th bit for the third octet of an IP address.
- **src** or **dst ip**— Specifies load balancing based on source or destination IP address.
- **src ip** or **src ip-l4port**— Specifies load balancing based on source IP address, or source IP address and source L4 port.
- **dst ip** or **dst ip-l4port**— Specifies load balancing based on destination IP address, or destination IP address and destination L4 port.

Hot Standby

ITD supports N+1 redundancy where M nodes can act as standby nodes for N active nodes.

When an active node fails, ITD looks for an operational standby node and selects the first available 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 newly active node. The service does not impose any fixed mapping of standby nodes to active nodes.

When the failed node becomes operational again, it is reinstated as an active node and traffic from the acting standby node is redirected back to the original node and the 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 standby node.

A node can be configured as a standby at the node-level or device-group-level. A node-level standby receives traffic only if its associated active node fails. A device-group-level standby receives traffic if any of the active nodes fail.

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. The **ingress interface** command enables you to configure multiple ingress interfaces.

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

Configuring 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 IPv4 traffic and an IPv6 ITD policy is applied to redirect IPv6 traffic.

Note Make sure the ingress interface is not configured in more than one IPv4 ITD service and/or more than one IPv6 ITD service. The system does not automatically check this.

System Health Monitoring

ITD supports health monitoring functionality to do the following:

- Monitor the ITD channel and peer ITD service.
- Monitor the state of the interface connected to each node.
- Monitor the health of the node through the configured probe.
- Monitor the state of ingress interface(s).

With health monitoring, the following critical errors are detected and remedied:

- ITD service is shut/no shut or deleted.
- iSCM process crash.
- iSCM process restart.
- Switch reboot.
- Supervisor switchover.
- In-service software upgrade (ISSU).
- ITD service node failure.
- ITD service node port or interface down.
- Ingress interface down.

Monitor Node

The ITD health monitoring module periodically monitors nodes to detect any failure and to handle failure scenarios.

ICMP, TCP, UDP, DNS and HTTP probes are supported to probe each node periodically for health monitoring. A probe can be configured at the device-group level or at node-level. A probe configured at the device-group level is sent to each node member of the device-group. A probe configured at a node-level is sent only to the node it is associated with. If a node-specific probe is configured, only that probe is sent to the node. For all the nodes that do not have node-specific probe configuration, the device-group level probe (if configured) is sent.

Note HTTPS probe is not supported on ITD.

IPv4 Control Probe for IPv6 Data Nodes

For an IPv6 node (in an IPv6 device-group), if the node is a dual-homed node (that is, it supports IPv4 and IPv6 network interfaces), an IPv4 probe can be configured to monitor the health. Since IPv6 probes are not supported, this provides a way to monitor health of IPv6 data nodes using a IPv4 probe.

Note IPv6 probes are not supported.

Health of an Interface Connected to a Node

ITD leverages the IP service level agreement (IP SLA) feature to periodically probe each node. The probes are sent at a one second frequency and sent simultaneously to all nodes. You can configure the probe as part of the cluster group configuration. A probe is declared to have failed after retrying three times.

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.
- Identifies the node as a candidate node for traffic handling, if the standby node is operational.
- 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.

Monitor Peer ITD Service

For sandwich mode cluster deployments, the ITD service runs on each Cisco NX-OS 7000 series switch. The health of the ITD channel is crucial to ensure flow coherent traffic passing through cluster nodes in both directions.

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

Note Since only a single instance of the ITD service is running on the switch in one-arm mode deployment, monitoring of the peer ITD is not applicable.

ITD channel failure handling

If the heartbeat signal is missed three times in a row, then the ITD channel is considered to be down.

While the ITD channel is down, traffic continues to flow through cluster nodes. However, since the ITD service on each switch is not able to exchange information about its view of the cluster group, this condition requires immediate attention. A down ITD channel can lead to traffic loss in the event of a node failure.

Failaction Reassignment

Failaction for ITD enables traffic on the failed nodes to be reassigned to the first available active node. Once the failed node comes back, it automatically resumes serving the connections. The **failaction** command enables this feature.

When the node is down, the traffic 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, traffic is reassigned to the next available active node. Once the failed node becomes active again, traffic is diverted back to the new node and resumes serving connections.

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

From Cisco NX-OS Release 8.4(1), the ITD failaction feature is enhanced to optimally pre-fetch the failaction node per bucket (i.e., the status of the service nodes before reassigning the failed node's buckets to the next available active nodes). This functionality is automatically added when a user upgrades to Cisco NX-OS Release 8.4(1). You do not need to configure any commands to enable this functionality.

From Cisco NX-OS Release 8.3(1), the ITD failaction feature is enhanced to optimally pre-fetch the status of the service nodes before reassigning the failed node's buckets to the next available active nodes. This functionality is automatically added when a user upgrades to Cisco NX-OS Release 8.3(1). You do not need to configure any commands to enable this functionality.

When a node is down, instead of assigning it to the next available active node, the pre-fetch optimization internally fetches the status of all the available nodes and then reassigns the failed node to an active node. Similarly, in a least bucket node fail reassignment, the pre-fetch optimization fetches the status of all the nodes before reassigning the failed node buckets to the least bucketed node.

Prior to Cisco NX-OS Release 8.3(1), when the node is down, the 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 bucket is reassigned to the next available active node. The reassignment is done in a round robin fashion across nodes. The delay in reassigning the failed node bucket to an active node impacts the network performance.

Currently, the pre-fetch optimization is enabled for the failaction node least-bucket and failaction node per-bucket options.

The following example shows the failaction assignment functionality before and after pre-fetch optimization.

Without pre-fetch optimization:

Let us consider an example of 4 Nodes with 256 buckets, and each node has 64 buckets.

- Suppose Node 1 and Node 3 fails. ITD will process node failure notification one at a time.
- ITD processes Node 1 failure first and reassigns Node 1's 32 buckets to Node 2, Node 3, and Node 4. 64 buckets are reassigned. Node 2 (32+22), Node 3 (32+21), and Node 4(21).
- ITD receives Node 3 failure notification. It has to move Node 3 (32 + 21) buckets to Node 2 and Node 4. So this time, total 53 buckets need to be reassigned.
- Node 1 failure (64 buckets are moved) + Node 3 failure (85 buckets are moved) = **total 149 buckets are reassigned**.

With pre-fetch optimization:

Let us consider an example of 4 Nodes with 256 buckets, and each node has 64 buckets.

- Suppose Node 1 and Node 3 fails. ITD will check the status of all the nodes before reassigning the buckets.
- Now, it moves Node 1's 32 buckets to Node 2 and Node 4. And moves Node 3's buckets to Node 2 and Node 4.
- Node 1 failure (64 buckets are moved) + Node 3 failure (64 buckets are moved) = **total 128 buckets are reassigned**.

Failaction Reassignment Without a Standby Node

When the node is down, the traffic 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 bucket is reassigned to the next available active node. Once the failed node comes back and becomes active, the traffic is diverted back to the new node and starts serving the connections.

If all the nodes are down, the packets get routed automatically.

- When the node goes down (probe failed), the traffic is reassigned to the first available active node.
- When the node comes up (probe success) from the failed state, it starts handling the connections.
- If all the nodes are down, the packets get routed automatically.

Failaction Reassignment with a Standby Node

When the node is down and if the standby is active, the traffic serves the connections and there is no change in the bucket assignment. When both the active and standby nodes are down, the traffic 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 bucket is reassigned to the next available active node. Once the failed node comes back up and becomes active, the traffic is diverted back to the new node and begins serving connections.

- When the node goes down (probe failed) and when there is a working standby node, traffic is directed to the first available standby node.
- When all nodes are down including the standby node, the traffic is reassigned to the first available active node.
- When the node comes up (probe success) from failed state, the node that came up starts handling the connections.
- If all the nodes are down, the packets are routed automatically.

No Failaction Reassignment

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

- Scenario 1: Probe configured; and:
  - with standby configured; or
  - without standby configured.

- Scenario 2: No probe configured.

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 and a standby is configured, the standby node takes over the connections.
- If the node fails and there is no standby configuration, the traffic gets routed and does not get reassigned, as failaction is not configured. Once the node recovers, the recovered node starts handling 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.

Prerequisites for ITD

ITD has the following prerequisites:

- You must enable the ITD feature with the **feature itd** command.
- The following commands must be configured prior to entering the **feature itd** command:
  - **feature pbr**
  - **feature sla sender**
  - **feature sla responder**
  - **ip sla responder**

Guidelines and Limitations for ITD

ITD has the following configuration guidelines and limitations:

- From Cisco NX-OS Release 8.4(3), statistics for an ITD service that has include ACL is supported.
- From Cisco NX-OS Release 8.4(2), the ACLs created by ITD are not displayed in the show ip/ipv6 access-list command output. You need to use show ip/ipv6 access-list dynamic command to get the ITD ACL list.
- Ensure that all node IPs in use by ITD services with the destination NAT feature enabled are reachable when the service is initially brought up. Also services with destination NAT enabled are required to be shut before reloading the switch. Service shut followed by a no-shut is recommended if nodes are unreachable during service enablement or if the service is enabled across reloads.
- Ensure that all node IPs in use by ITD services with the destination NAT feature enabled are Layer-2 adjacent.
- ITD services with destination NAT feature is not supported with fail-action mechanisms of fail-action distribute and fail-action node-per-bucket.

- ITD sessions are not supported on device-groups used by services with NAT destination feature enabled.
- ITD NAT is not supported on Cisco Nexus 7000 and Cisco Nexus 7700 Series switches.
- A combination of ITD Standby, Hot Standby, and Failaction mechanism is not supported in a single device-group.
- Hot Standby is not supported with the bucket distribute failaction method.
- When ITD service is enabled, access-lists, route-maps, tracks, and IP SLA are auto-configured. Ensure that you do not modify or remove these configurations. Modifying these configurations disrupts ITD functionality.
- Virtual IP type and the ITD device group nodes type should be either IPv4 or IPv6, but not both.
- Configure the same protocol for both ITD Virtual IP (VIP) and load balance method. If VIP is configured with no protocol, VIP assumes the value as ANY, and load balance method cannot be TCP or UDP. A mismatch in the protocol method can cause the following error:

  ```
  ERROR: VIP Protocol and LB Protocol does not match
  ```

- A total number of 300 ACEs (including deny/permit/remark) are supported for an ACL. However, a maximum of 256 permit ACEs are supported in one ACL.
- From Cisco NX-OS Release 8.4(2), a total number of 2000 ACEs are supported for multiple Include ACLs.
- You can configure upto 8 ACLs in a ITD service.
- In a multi-dimensional ITD configuration (Virtual IPs or include ACL with number of ACEs or number node per bucket), the number of TCAM entries per ITD service cannot exceed more than 2000.
- You can configure either VIP or Include ACL on a single ITD service, but not both.
- IPv6 probes are not supported for a device group with IPv6 nodes, however IPv4 probes can be configured to monitor an IPv6 data node if the node is dual-homed (that is, it has both IPv6 and IPv4 networks interfaces).
- Configuration rollback is only supported when the ITD service is in shut mode in both target and source configurations.
- SNMP is not supported for ITD.
- Beginning from Cisco 7000 Series Switches Release 8.2(1), ITD is supported on M3 Line Cards.
- ITD does not support FEX, either with ingress or egress traffic.

The Optimized Node Insertion/Removal feature is supported:

- Without standby nodes and backup nodes
- Not supported with weights
- Not supported with NAT (Cisco NX-OS 7000 Series switch)
- Not supported with the Include ACL feature configured
- Not supported with Node level probes.

The following are ITD guidelines and restrictions for IPv6:

- IPv6 with IPv4 probe is supported on F3 (on Nexus 7000 Series and Nexus 7700) and F2E (Nexus 7700) modules only.
- IPv6 probe for the IPv6 standby node is not supported.
- IPv6 probe for the IPv6 hot-standby node is supported.
- Beginning with Cisco NX-OS Release 8.3(1), the ITD feature is supported for IPv6 services.
- Beginning with Cisco NX-OS Release 8.3(1), IPv6 device group probes are supported.
- IPv6 services for ITD is not supported on F2E Line Cards.
- ITD service groups and modules does not support IPv6 NAT destination.
- Beginning with Cisco NX-OS Release 8.2(1), IPv6 is supported on M3 modules.
- ITDv6 supports only the failaction reassign and failaction least-bucket.
- In Cisco NX-OS Release 8.3(1), device probes for TCPv6 and ICMPv3 are supported.
- Per node-probe level is not supported.

The following are ITD guidelines and restrictions for IPv4:

- In the Cisco NX-OS Release 7.3(0)D1(1), the Include ACL feature is supported for IPv4 only.
- The following fail-action methods are supported on IPv4:
  - **reassign**
  - **least-bucket**
  - **per-bucket**
  - **bucket-distribute**

Default Settings for ITD

This table lists the default settings for ITD parameters.

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

Note The default probe values will change based on the server capability and number of applications configured. For example, if the server is busy, users can configure probe timeout to be longer and reduce the frequency.

Configuring ITD

The server can be connected to the switch through a routed interface or port-channel, or via a switchport port with SVI configured.

Enabling ITD

Before you begin

Before you configure the **feature itd** command you must enter the **feature pbr** and **feature ipsla** commands.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **feature itd**	Enables the ITD feature.

Configuring a Device Group

Before you begin

Enable the ITD feature.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **itd device-group** *name*	Creates an ITD device group and enters into device group configuration mode.
Step 3	switch(config-device-group)# **node ip** *ipv4-address*	Specifies the nodes for ITD. Repeat this step to specify all nodes. To configure IPv6 nodes, use the **node ipv6** *ipv6-address* . **Note** An ITD device group can have either IPv4 or IPv6 nodes, but not both.
Step 4	switch(config-dg-node)# [**mode hot-standby**] [**standby** *ipv4-address*] [**weight** *value*] [**probe** {**icmp**| **tcp port** *port-number*| **udp port** *port-number*| **dns** {*hostname*| *target-address*} | **http get** *filename*} [**frequency** *seconds*]	Specifies the device group nodes for ITD. Repeat this step to specify all nodes. The **weight** *value* keyword specifies the proportionate weight for the node for weighted traffic distribution.

	Command or Action	Purpose
	[[**retry-down-count**| **retry-up-count**] *number*] [**timeout** *seconds*]	The **mode hot-standby** specifies that this is node is to be designated as standby node for the device-group. A node-level standby can be associated for each node. The **standby** value specifies the standby node information for this active node. A node-level probe can be configured to monitor health of the node. The **Probe** value specifies probe parameters to use for monitoring health of this active node. **Note** IPv6 probes are not supported.
Step 5	switch(config-device-group)# **probe** {**icmp** | **tcp port** *port-number* | **udp port** *port-number* | **dns** {*hostname* |*target-address*} | **http get** *filename* } [**frequency** *seconds*] [[**retry-down-count** | **retry-up-count**] *number*] [**timeout** *seconds*]	Configures the cluster group service probe. You can specify the following protocols as the probe for the ITD service: • ICMP • TCP • UDP • DNS • HTTP The keywords are as follows: • **retry-down-count**—Specifies the consecutive number of times the probe must have failed prior to the node being marked DOWN. • **retry-up-count**—Specifies the consecutive number of times the probe must have succeeded prior to the node being marked UP. • **timeout**—Specifies the number of seconds to wait for the probe response. • **frequency**—Specifies the time interval in seconds between successive probes sent to the node. **Note** IPv6 probes are not supported.

Configuring an ITD Service

Before you begin

- Enable the ITD feature.
- Configure the device-group to be added to the ITD service.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **itd** *service-name*	Configures an ITD service and enters into ITD configuration mode.
Step 3	switch(config-itd)# **device-group** *device-group-name*	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	switch(config-itd)# **virtual ip** *ipv4-address ipv4-network-mask* **device-group** *device-group-name* [**advertise** {**enable** |**disable**} [**active**]]	Allows you to configure a VIP for ITD device group with route creation based on health of device group node.
Step 5	switch(config-itd)# **ingress interface** *interface*	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 6	switch(config-itd)# **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*}	Configures the load-balancing options for the ITD service. The keywords are as follows: • **buckets**—Specifies the number of buckets to create. Buckets must be configured in powers of two. • **mask-position**— Specifies the mask position of the load balance. • **method**—Specifies the source IP address or destination IP address, or source IP address and source port, or the destination IP address and destination port based load-balancing.

	Command or Action	Purpose
Step 7	switch(config-itd)# **virtual ip** *ipv4-address ipv4-network-mask* [**tcp** | **udp** {*port-number* | **any**}] [**advertise** {**enable** | **disable**}]	Configures the virtual IPv4 address of the ITD service. Configure a single VIP address for an ITD service serving a group of nodes (or device group). **Note** To configure an IPv6 virtual address, use the **virtual ipv6** *ipv6-address ipv6-network-mask* | *ipv6-prefix/length*} [**ip** | **tcp** {*port-number* | **any**} | **udp** {*port-number* | **any**}] [**advertise** {**enable** | **disable**}] The **advertise enable** keywords specify that the virtual IP route is advertised to neighboring devices. The **tcp**, **udp**, and **ip** keywords specify that the virtual IP address will accept flows from the specified protocol.
Step 8	switch(config-itd)# **failaction node per-bucket**	When a particular node is failed, the least bucketed node is identified and the buckets are distributed across the rest of the active nodes starting from the least bucketed node.
Step 9	switch(config-itd)# **failaction node reassign**	Enables traffic to be reassigned, following a node failure. The traffic to the failed node gets reassigned to the first available active node.
Step 10	switch(config-itd)# **vrf** *vrf-name*	Specifies the VRF for the ITD service.
Step 11	switch(config-itd)# **no shutdown**	Enables the ITD service.
Step 12	switch(config-itd)# **exclude access-list** *acl-name*	Excludes traffic from redirection. The **acl-name** specifies the matching traffic that should be excluded from ITD redirection.

Configuring Destination NAT

Configuring Virtual IP Address any with NAT Destination

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`switch# configure terminal`	
Step 2	**itd** *service-name* **Example:** `switch (config) # itd service1`	Configures an ITD service and to enter into ITD configuration mode.
Step 3	**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	**virtual ip** *ipv4-address ipv4-network-mask* **Example:** `switch(config-itd)# virtual ip 172.16.1.10 255.255.255.255`	Configures the virtual IPv4 address of an ITD service.
Step 5	**nat destination** **Example:** `switch(config-itd)# nat destination`	Configures destination NAT.
Step 6	**ingress interface interface next-hop** *ip-address* **Example:** `switch(config-itd)# ingress interface ethernet 3/1 next-hop 203.0.113.254`	Adds an ingress interface or multiple interfaces to an ITD service and configures the next hop IP address which is the IP address of the interface connected directly to the configuring ingress interface.
Step 7	**no shutdown** **Example:** `switch(config-itd)# no shutdown`	Enables the ITD service.

Configuring Virtual IP Address with Port with NAT Destination

Before you begin

Enable the ITD feature.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.
Step 2	**itd** *service-name* **Example:** `switch (config) # itd service1`	Configures an ITD service and to enter into ITD configuration mode.

	Command or Action	Purpose
Step 3	**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	**virtual ip** *ipv4-address ipv4-network-mask* **80***80* **Example:** `switch(config-itd)# virtual ip 172.16.1.10 255.255.255.255`	Configures the virtual IPv4 address with TCP port on an ITD service.
Step 5	**nat destination** **Example:** `switch(config-itd)# nat destination`	Configures destination NAT.
Step 6	**ingress interface interface next-hop** *ip-address* **Example:** `switch(config-itd)# ingress interface ethernet  3/1 next-hop 192.168.1.70`	Adds an ingress interface or multiple interfaces to an ITD service and configures the next hop IP address which is the IP address of the interface connected directly to the configuring ingress interface.
Step 7	**no shutdown** **Example:** `switch(config-itd)# no shutdown`	Enables the ITD service.

Configuring Multiple Virtual IP with NAT Destination and Port Translation

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.
Step 2	**itd device-group** *name* **Example:** `switch(config)# itd device-group dg`	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 3	**node ip** *ipv4-address* **Example:** `switch(config-device-group)# node ip 192.168.1.20`	Creates an IPv4 cluster node for Intelligent Traffic Director.
Step 4	**exit** **Example:**	Exits the ITD device group configuration mode and enters the global configuration mode.

	Command or Action	Purpose
	`switch# exit`	
Step 5	**itd** *service-name* **Example:** `switch (config) # itd service1`	Configures an ITD service and to enter into ITD configuration mode.
Step 6	**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 7	**virtual ip** *ipv4-address ipv4-network-mask* **Example:** `switch(config-itd)# virtual ip 172.16.1.10 255.255.255.255`	Configures the virtual IPv4 address of an ITD service.
Step 8	**virtual ip** *ipv4-address ipv4-network-mask* **Example:** `switch(config-itd)# virtual ip 172.16.1.20 255.255.255.255`	Configures the virtual IPv4 address of an ITD service.
Step 9	**nat destination** **Example:** `switch(config-itd)# nat destination`	Configures destination NAT.
Step 10	**ingress interface** *interface slot / port* **Example:** `switch(config-itd)# ingress interface ethernet  3/1`	Adds an ingress interface to an ITD service.

Configuring Optimized Node Insertion or Removal

Configuring Optimized Node Insertion

Configuring an ITD Service

Before you begin

- To configure the include ACL feature, you need to configure the loadbalance command.

Procedure

Step 1 Enter global configuration mode:

switch# **configure terminal**

Step 2 Create an ITD device group and enters into device group configuration mode:

switch(config)# **itd device-group** *name*

Step 3 Specify the nodes for ITD.

- Repeat this step thrice to specify three nodes using the following IP addresses one for each repetition:
  - 10.2.1.10
  - 10.2.1.20
  - 10.2.1.30
- To configure IPv6 nodes, use the **node ipv6** *ipv6-address*.

switch(config-device-group)# **node ip** *ipv4-address*

Step 4 Configure an ITD service and enters into ITD configuration mode:

switch(config-device-group) #**itd service-name**

Step 5 Add 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.

switch(config-itd)# **device-group** *device-group-name*

Step 6 Add an ingress interface an ITD service:

switch(config-itd)# **ingress interface** *interface slot/port*

Step 7 Enable the ITD device:

switch(config-itd)# **no shutdown**

Creating an ITD Session to Insert Nodes

Procedure

Step 1 Enter global configuration mode:

switch# **configure terminal**

Step 2 Create an ITD session:

switch# **itd session device-group webservers**

Step 3 Specify the nodes for ITD. Repeat this step to specify all nodes:

switch(config-device-group)# **node ip**

Step 4 Use the **commit** command to synchronize the configuration with the peer switch and to apply the configuration locally. Configurations are stored in the buffer until the **commit** command is issued.

switch(config-device-group)#**commit**

Example for Optimized Node Insertion

The following is the node distribution on some of the scenarios of configuring optimized insertion:

if there are 3 nodes in the devices group, then the default buckets are distributed as such:

Node1 = bucket 1 and 4

Node2 = bucket 2

Node3 = bucket 3

When a fourth bucket is added the fourth bucket is redistributed to the newly added node (Node4) resulting in the following distribution :

if there are 3 nodes in the devices group, then the default buckets are distributed as such:

Node1 = bucket 1

Node2 = bucket 2

Node3 = bucket 3

Node4 = bucket 4

If another node is added, new buckets are required. This will always be the next power of 2 in number. Thus by adding a 5th node 8 buckets are created by default:

Here is the new distribution:

Node 1 = bucket 1 and 6

Node 2 = bucket 2 and 7

Node 3 = bucket 3 and 8

Node4 = bucket 4

Node5 = bucket 5

Configuration Example: Configuring Optimized Node Insertion

This example shows a running configuration:

```
configure terminal
 itd device-group webservers
  node ip 10.2.1.10
  node ip 10.2.1.20
  node ip 10.2.1.30
  itd http_service
   device-group webservers
   ingress interface Ethernet 3/1
   no shutdown
   exit
 itd session device-group webservers
  node ip 10.2.1.40
  commit
```

Configuring Optimized Node Removal

Creating an ITD Session to Remove Nodes

Before you begin

Configure ITD Services. Refer the configuration in the previous task for ITD service *http_service* which has 4 nodes in device group *webservers*. Use the below steps to remove a service without impacting the service to the other nodes.

Procedure

Step 1 Enter global configuration mode:

switch# **configure terminal**

Step 2 Create an ITD session:

switch(config)#**itd session device-group** *name*

Step 3 Specify the nodes those are to be deleted, which already is part of the configured device-group:

switch(config)# **no node ip** *ipv4-address*

Step 4 Specify the node for ITD:

switch(config-device-group)# **node ip** *ipv4-address*

Step 5 Use the **commit** command to synchronize the configuration with the peer switch and to apply the configuration locally. Configurations are stored in the buffer until the **commit** command is issued.

switch(config-device-group)# **commit**

Example for Optimized Node Removal

When deleting a node, the bucket(s) associated to it is redistributed to the nodes with the least buckets assign starting with the first node in the device group.

Node1 = bucket 1

Node2 = bucket 2

Node3 = bucket 3

Node4 = bucket 4

Now, if Node2 is removed, the bucket distribution will be the following:

if there are 3 nodes in the devices group, then the default buckets are distributed as such:

Node1 = bucket 1 and 2

Node2 (deleted)

Node3 = bucket 3

Node4 = bucket 4

Configuration Example: Configuring Optimized Node Removal

This example shows a running configuration:

```
configure terminal
 itd device-group webservers
  node ip 10.2.1.10
  node ip 10.2.1.20
  node ip 10.2.1.30
  itd http_service
  device-group webservers
  ingress interface Ethernet 3/1
  no shutdown
  exit
 itd session device-group webservers
  no node ip 10.2.1.20
```

Configuring Optimized Node Replacement

Creating an ITD Session to Replace Nodes

Before you begin

Configure ITD Services. Refer the configuration in the previous task for ITD service *http_service* which has 4 nodes in device group *webservers*. Use the steps below to replace a service without impacting the service to the other nodes.

Procedure

Step 1 Enter global configuration mode:

switch# **configure terminal**

Step 2 Create an ITD session:

switch(config)# **itd session device-group** *name*

Step 3 Specify the node that is to be removed:

switch(config-device-group)# **no node ip** *ipv4-address*

Step 4 Specify the node that is to be added.

switch(config-device-group)# **node ip** *ipv4-address*

Step 5 Use the **commit** command to synchronize the configuration with the peer switch and to apply the configuration locally. Configurations are stored in the buffer until the **commit** command is issued.

switch(config-device-group)# **commit**

Example for Optimized Node Replacement

When deleting a node, the bucket(s) associated to it is redistributed to the nodes with the least buckets assign starting with the first node in the device group.

Node1 = bucket 1

Node2 = bucket 2

Node3 = bucket 3

Node4 = bucket 4

Now, if Node2 is removed, the bucket distribution will be the following:

if there are 3 nodes in the devices group, then the default buckets are distributed as such:

Node1 = bucket 1 and 2

Node2 (deleted)

Node3 = bucket 3

Node4 = bucket 4

Configuration Example: Configuring Optimized Node Replacement

This example shows a running configuration:

```
configure terminal
 itd device-group webservers
  node ip 10.2.1.10
  node ip 10.2.1.20
  node ip 10.2.1.30
  itd http_service
  device-group webservers
  ingress interface Ethernet 3/1
  no shutdown
  exit
 itd session device-group webservers
  no node ip 10.2.1.30
  node ip 10.2.1.50
 commit
```

Configuring a Device Group

Before you begin

Enable the ITD feature.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **itd device-group** *name*	Creates an ITD device group and enters into device group configuration mode.
Step 3	switch(config-device-group)# **node ip** *ipv4-address*	Specifies the nodes for ITD. Repeat this step to specify all nodes. To configure IPv6 nodes, use the **node ipv6** *ipv6-address* . **Note** An ITD device group can have either IPv4 or IPv6 nodes, but not both.
Step 4	switch(config_dg_node)# [**mode hot-standby**] [**standby** *ipv4-address*] [**weight** *value*] [**probe** {**icmp**| **tcp port** *port-number*| **udp port** *port-number*| **dns** {*hostname*| *target-address*}} [**frequency** *seconds*] [[**retry-down-count**| **retry-up-count**] *number*] [**timeout** *seconds*]	Specifies the device group nodes for ITD. Repeat this step to specify all nodes. The **weight** *value* keyword specifies the proportionate weight for the node for weighted traffic distribution. The **mode hot-standby** specifies that this is node is to be designated as standby node for the device-group. A node-level standby can be associated for each node. The **standby** value specifies the standby node information for this active node. A node-level probe can be configured to monitor health of the node. The **Probe** value specifies probe parameters to use for monitoring health of this active node. **Note** IPv6 probes are not supported.
Step 5	switch(config-device-group)# **probe** {**icmp** | **tcp port** *port-number* | **udp port** *port-number* | **dns** {*hostname* |*target-address*} } [**frequency** *seconds*] [[**retry-down-count** | **retry-up-count**] *number*] [**timeout** *seconds*]	Configures the cluster group service probe. You can specify the following protocols as the probe for the ITD service: • ICMP • TCP • UDP • DNS The keywords are as follows: • **retry-down-count**—Specifies the consecutive number of times the probe

	Command or Action	Purpose
		must have failed prior to the node being marked DOWN. • **retry-up-count**—Specifies the consecutive number of times the probe must have succeeded prior to the node being marked UP. • **timeout**—Specifies the number of seconds to wait for the probe response. • **frequency**—Specifies the time interval in seconds between successive probes sent to the node. **Note** IPv6 probes are not supported.

Verifying the ITD Configuration

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

Command	Purpose
show itd [*itd-name*] [**brief**]	Displays the status and configuration for all or specified ITD instances. • Use the *itd-name* argument to display the status and configuration for the specific instance. • Use the **brief** keyword to display summary status and configuration information.
show itd [*itd-name* | **all**] {**src** | **dst**} *ip-address*] **statistics** [**brief**]	Displays the statistics for ITD instances. • Use the *itd-name* argument to display statistics for the specific instance. • Use the **brief** keyword to display summary information. **Note** Before using the **show itd statistics** command, you need to enable ITD statistics by using the **itd statistics** command.
show running-config services	Displays the configured ITD device-group and services.
show itd session device-group	Lists all the sessions configured.

Command	Purpose
show itd session device-group*device-group-name*	Lists the ITD session matching the name of the device-group.

These examples show how to verify the ITD configuration:

```
switch# show itd

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

Exclude ACL
------------------------------
 exclude-smtp-traffic

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


Pool                           Interface    Status Track_id
------------------------------ ------------ ------ ---------
WEB_itd_pool                   Po-1         UP     3

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

  Node  IP                 Config-State Weight Status     Track_id  Sla_id
  ------------------------ ------------ ------ ---------- --------- ---------
  1     10.10.10.11        Active       1      OK           1         10001

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


  Node  IP                 Config-State Weight Status     Track_id  Sla_id
  ------------------------ ------------ ------ ---------- --------- ---------
  2     10.10.10.12        Active       1      OK           2         10002

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

switch# show itd brief

Name           Probe LB Scheme  Interface  Status   Buckets
-------------- ----- ---------- ---------- -------- --------
WEB            ICMP  src-ip     Eth3/3     ACTIVE   2

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

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

```
  Node  IP                   Config-State Weight Status     Track_id  Sla_id
  ----------------------- ------------ ------ ---------- --------- ---------
  1     10.10.10.11        Active       1      OK          1         10001
  2     10.10.10.12        Active       1      OK          2         10002

switch(config)# show itd statistics

Service        Device Group             VIP/mask               #Packets
-------------------------------------------------------------------------------
test            dev              9.9.9.10 / 255.255.255.0       114611  (100.00%)

Traffic Bucket        Assigned to      Mode       Original Node   #Packets
-------------------------------------------------------------------------------
test_itd_vip_0_acl_0  10.10.10.9       Redirect    10.10.10.9      57106  (49.83%)

Traffic Bucket        Assigned to      Mode       Original Node   #Packets
-------------------------------------------------------------------------------
test_itd_vip_0_acl_1  12.12.12.9       Redirect    12.12.12.9      57505  (50.17%)
switch (config)# show running-config services

version 6.2(10)
feature itd

itd device-group WEB-SERVERS
probe icmp
node ip 10.10.10.11
node ip 10.10.10.12

itd WEB
device-group WEB-SERVERS
virtual ip 10.10.10.100 255.255.255.255
ingress interface po-1
no shut
```

Configuring Include ACL

Before you begin

Enable the ITD feature.

Enable the ITD service.

To configure the include ACL feature, you need to configure the loadbalance command.

Procedure

Step 1 Enter global configuration mode:

switch# **configure terminal**

Step 2 Defines an IP access list by name:

switch(config-if)# **ip access-list** *access-list-name*

Step 3 Set the conditions for a named IP access list and configure permit ACEs to select traffic for ITD:

switch(config-acl)# **permit ip any** *destination-address address-mask*

Step 4 Set the conditions for a named IP access list and configure permit ACEs to select traffic for ITD:

- Note: This example shows two ACEs one for selecting any traffic to the destination network 209.165.202.0/27 and the traffic from source network 192.168.10.0/24 to the destination.

switch(config-acl)# **permit ip any** *source-address address-maskdestination-address address-mask*

Step 5 Exits the ACL configuration mode:

switch(config-acl)# **exit**

Step 6 Add an existing device group to the ITD service. The *device-group-name* argument specifies the name of the device group. You can enter up to 32 alphanumeric characters:

- Use a comma (",") to separate multiple interfaces.
- Use a hyphen ("-") to separate a range of interfaces.

switch(config)# **device-group** *device-group-name*

Step 7 Add 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.

switch(config-itd)# **ingress interface** *interface*

Step 8 Configure the load-balancing options for the ITD service:

- Method keyword-Specifies the source IP address or destination IP address based load/traffic distribution.

switch(config-itd)# **load-balance** *method src ip*

Step 9 Apply the specified ACL in the ITD service or interface:

- Method keyword—Specifies the source IP address or destination IP address based load/traffic distribution.

switch(config-itd)# **access-list** *acl-name*

Step 10 Associate the ACL to the device-group. You can repeat this step to associate all ACLs to the respective device groups.

- Method keyword—Specifies the source IP address or destination IP address based load/traffic distribution.

switch(config-itd)# **access-list** *acl-name* **device-group** *device-group-name*

From Cisco NX-OS Release 8.4(1), you can configure multiple ACLs under an ITD service. You have an option to associate each ACL with its own device-group.

Configuring Include ACL

This example shows a running configuration:

```
configure terminal
 ip access-list includeACL
  permit ip any 209.165.201.0 255.255.255.224
  permit ip any 192.168.10.0 255.255.255.0 209.165.201.0 255.255.255.224
```

```
  exit
   device-group dg1
    ingress interface Ethernet 3/1
    load-balance method src ip
    access-list includeACL2
```

Example: Configuring Multiple ACLs under an ITD Service And Associating With The Device Group

This example shows how to configure multiple ACLs under an ITD service and associate each ACL with its own device group.

```
ip access-list test1
  10 permit tcp 20.1.1.0/24 any
  20 permit tcp 21.1.1.0/24 any
ip access-list test2
  10 permit tcp 30.1.1.0/24 any
  20 permit tcp 31.1.1.0/24 any
  30 permit tcp 32.1.1.0/24 any
ip access-list test3
  10 permit ip 40.1.1.0/24 any
  11 permit ip 41.1.1.0/24 any
  12 permit ip 42.1.1.0/24 any
  13 permit ip 43.1.1.0/24 any
feature itd


itd device-group DG1
  probe icmp frequency 2 timeout 2
  node ip 10.1.1.1
  node ip 11.1.1.1
  node ip 12.1.1.1
  node ip 13.1.1.1


itd device-group DG2
  probe icmp frequency 1 timeout 1
  node ip 14.1.1.1
  node ip 15.1.1.1


itd device-group DG3
  probe icmp frequency 1 timeout 1
  node ip 16.1.1.1
  node ip 17.1.1.1
  node ip 18.1.1.1

itd SERVICE
  ingress interface Eth13/24
  load-balance method src ip
  access-list test1 device-group DG1
  access-list test2 device-group DG2
  access-list test3 device-group DG3
  no shut
```

This example shows how to configure multiple ACLs under an ITD service for IPv6.

```
itd device-group DG1-IPV6
probe icmp
node ipv6 101:1::101:1
node ipv6 102:1::102:1
```

```
node ipv6 103:1::103:1
node ipv6 104:1::104:1

itd device-group DG2-IPV6
probe icmp
node ipv6 203:1::103:1
node ipv6 204:1::104:1

itd SERVICE-IPV6
ingress interface Eth1/2
load-balance method src ip
failaction node per-bucket
access-list ipv6 test1-ipv6 device-group DG1-IPV6
access-list ipv6 test2-ipv6 device-group DG2-IPV6
no shut
```

Verifying the Include ACL

To display the ITD configuration and to verify Include ACL feature, perform one of the following tasks:

Command	Purpose
show itd [*itd-name*] [**brief**]	Displays the status and configuration for all or specified ITD instances. • Use the *itd-name* argument to display the status and configuration for the specific instance. • Use the **brief** keyword to display summary status and configuration information.
show running-config services	Displays the configured ITD device-group and services.
show ip access-lists *name*	Displays the specified IP ACL configuration.
show{ **ip** | **ipv6**} **access-list dynamic**	Displays the IP/IPv6 ACLs created by ITD.

These examples show how to verify the ITD configuration:

```
switch# show itd

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

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

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


Device Group                                        Probe  Port
--------------------------------------------------- ----- ------
WEB-SERVERS                                               ICMP
```

```
Pool                           Interface    Status Track_id
------------------------------ ------------ ------ ---------
WEB_itd_pool                   Po-1       UP       4

ACL Name/SeqNo                   IP/Netmask/Prefix             Protocol  Port
-------------------------------- -------------------------------- ---- -------
acl2/10                          192.168.1.30/24                     TCP  0

  Node  IP            Cfg-S   WGT Probe Port     Probe-IP   STS Trk# Sla_id
  ------------------- ------- --- ---- ----- --------------- -- --- -------
  1     192.168.1.10    Active   1  ICMP                      OK   5   10005

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

  Node  IP            Cfg-S   WGT Probe Port     Probe-IP   STS Trk# Sla_id
  ------------------- ------- --- ---- ----- --------------- -- --- -------
  2     192.168.1.20     Active   1 ICMP                      OK   6   10006

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

ACL Name/SeqNo                   IP/Netmask/Prefix             Protocol  Port
-------------------------------- -------------------------------- ---- -------
acl2/20                          192.168.1.40/24                     TCP  0

  Node  IP            Cfg-S   WGT Probe Port     Probe-IP   STS Trk# Sla_id
  ------------------- ------- --- ---- ----- --------------- -- --- -------
  1     192.168.1.10   Active   1 ICMP                       OK   5   10005

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

  Node  IP            Cfg-S   WGT Probe Port     Probe-IP   STS Trk# Sla_id
  ------------------- ------- --- ---- ----- --------------- -- --- -------
  2    192.168.1.20   Active   1 ICMP                      OK   6   10006

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

These examples show how to verify the Include ACL feature:

```
switch (config)# show running-config services

!Command: show running-config services
!Time: Wed Feb 10 15:31:53 2016

version 7.3(1)D1(1)

feature itd


itd device-group WEB-SERVERS
  probe icmp
  node ip 192.168.1.10
  node ip 192.168.1.20


itd WEB
  device-group WEB-SERVERS
```

```
  ingress interface Po-1
  failaction node reassign
  load-balance method src ip
  access-list acl2
  no shut
```

These examples show how to verify the ACL lists

```
switch(config-itd)# show ip access-lists IncludeACL


10 permit ip any 209.165.201.0 255.255.255.224
20 permit ip 192.168.10.0 255.255.255.0 209.165.202.128 255.255.255.224
```

Configuring Multiple Device-Groups within an ITD Service

Creating Multiple Device Groups

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.
Step 2	**feature itd** *name* **Example:** `switch(config)# feature itd`	Enables the ITD feature.
Step 3	**itd 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	**probe icmp** **Example:** `switch(config-device-group)# probe icmp`	Configures the cluster group service probe for Intelligent Traffic Director.
Step 5	**node ip** *ipv4-address* **Example:** `switch(config-device-group)# node ip 192.168.1.10`	Creates an IPv4 cluster node for Intelligent Traffic Director.

	Command or Action	Purpose
Step 6	**node ip** *ipv4-address* **Example:** `switch(config-device-group)# node ip 192.168.1.20`	Creates an IPv4 cluster node for Intelligent Traffic Director.
Step 7	**exit** **Example:** `switch# exit`	Exits the ITD device group configuration mode and enters the global configuration mode.
Step 8	**itd device-group** *name* **Example:** `switch(config)# itd device-group dg_server1`	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 9	**probe icmp** **Example:** `switch(config-device-group)# probe icmp`	Configures the cluster group service probe for Intelligent Traffic Director.
Step 10	**node ip** *ipv4-address* **Example:** `switch(config-device-group)# node ip 192.168.1.30`	Creates an IPv4 cluster node for Intelligent Traffic Director.
Step 11	**node ip** *ipv4-address* **Example:** `switch(config-device-group)# node ip 192.168.2.40`	Creates an IPv4 cluster node for Intelligent Traffic Director.
Step 12	**exit** **Example:** `switch# exit`	Exits the ITD device group configuration mode and enters the global configuration mode.
Step 13	**itd device-group** *name* **Example:** `switch(config)# itd device-group dg_server2`	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 14	**probe icmp** **Example:** `switch(config-device-group)# probe icmp`	Configures the cluster group service probe for Intelligent Traffic Director.
Step 15	**node ip** *ipv4-address* **Example:** `switch(config-device-group)# node ip 192.168.1.50`	Creates an IPv4 cluster node for Intelligent Traffic Director.

	Command or Action	Purpose
Step 16	**node ip** *ipv4-address* **Example:** `switch(config-device-group)# node ip 192.168.1.60`	Creates an IPv4 cluster node for Intelligent Traffic Director.
Step 17	**exit** **Example:** `switch# exit`	Exits the ITD device group configuration mode and enters the global configuration mode.

Associating Multiple Device Group Within a Service

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.
Step 2	**itd** *service-name* **Example:** `switch (config) # itd multi-dg`	Configures an ITD service and to enter into ITD configuration mode.
Step 3	**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	**virtual ip** *ipv4-address ipv4-network-mask* **tcp** *port-number* **device-group** *device-group-name* **Example:** `switch(config-itd)# virtual ip 172.16.1.10 255.255.255.255 tcp 23 device-group dg1_servers`	Configures the virtual IPv4 address of an ITD service.
Step 5	**virtual ip** *ipv4-address ipv4-network-mask* **tcp** *port-number* **device-group** *device-group-name* **Example:** `switch(config-itd)# virtual ip 172.16.1.20 255.255.255.255 tcp 23 device-group dg2_servers`	Configures the virtual IPv4 address of an ITD service.
Step 6	**ingress interface interface** *name number* **Example:**	Adds an ingress interface or multiple interfaces to an ITD service and configures the next hop IP address which is the IP address of the

	Command or Action	Purpose
	`switch(config-itd)# ingress interface ethernet 3/1`	interface connected directly to the configuring ingress interface.
Step 7	**no shutdown** **Example:** `switch(config-itd)# no shutdown`	Enables the ITD service.

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)# probe icmp
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
```

This example shows how to configure a virtual IPv4 address:

```
switch(config)# feature itd
switch(config)# itd test
switch(config-itd)# device-group dg
switch(config-itd)# ingress interface Po-1
switch(config-itd)# virtual ip  172.16.1.10 255.255.255.255 advertise enable tcp any
```

This example shows how to configure a virtual IPv6 address:

```
switch(config)# feature itd
switch(config)# itd test
switch(config-itd)# device-group dg
switch(config-itd)# ingress interface Po-1
switch(config-itd)# virtual ipv6 ffff:eeee::cccc:eeee dddd:efef::fefe:dddd tcp 10 advertise
 enable
```

This example shows how to configure device-group-level standby node. Node 192.168.2.15 is configured as standby for the entire device group. If any of the active nodes fail, the traffic going to the failed node will be redirected to 192.168.2.15:

```
switch(config)# feature itd
switch(config)# itd device-group dg
switch(config-device-group)# probe icmp
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)# node ip 192.168.2.15
switch(config-dg-node)# mode hot standby
switch(config-dg-node)# exit
```

This example shows how to configure node-level standby node. Node 192.168.2.15 is configured as standby for node 192.168.2.11 only. Only when node 192.168.2.11 fails, the traffic going to node 192.168.2.11 is redirected to 192.168.2.15:

```
switch(config)# feature itd
switch(config)# itd device-group dg
```

```
switch(config-device-group)# probe icmp
switch(config-device-group)# node ip 192.168.2.11
switch(config-dg-node)# standby ip 192.168.2.15
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-dg-node)# exit
```

This example shows how to configure weight for proportionate distribution of traffic. 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 192.168.2.11
switch(config-dg-node)# weight 3
switch(config-device-group)# node ip 192.168.2.12
switch(config-dg-node)# weight 3
switch(config-device-group)# node ip 192.168.2.13
switch(config-device-group)# node ip 192.168.2.14
switch(config-dg-node)# exit
```

This example shows how to configure a node-level probe. Node 192.168.2.14 is configured with TCP probe and ICMP probe is configured for device-group. TCP probe gets sent to node 192.168.2.14 and ICMP probe gets sent to nodes 192.168.2.11, 192.168.2.12 and 192.168.2.13:

```
switch(config)# feature itd
switch(config)# itd device-group dg
switch(config-device-group)# probe icmp
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-dg-node)# probe tcp port 80
switch(config-dg-node)# exit
```

This example shows how to configure probe for standby mode. Node 192.168.2.15 is configured as standby for node 192.168.2.11 only. While ICMP probe is configured for device-group, TCP probe is configured for standby node 192.168.2.15. ICMP probe gets sent to nodes 192.168.2.11, 192.168.2.12, 192.168.2.13 and 192.168.2.14. TCP probe gets sent to node 192.168.2.15:

```
switch(config)# feature itd
switch(config)# itd device-group dg
switch(config-dg-node)# probe icmp
switch(config-device-group)# node ip 192.168.2.11
switch(config-device-group)# standby ip 192.168.2.15
switch(config-dg-node-standby)# probe tcp port 80
switch(config-dg-node)# 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-dg-node)# exit
```

This example shows how to configure IPv4 probe for IPv6 node.dg-v6 is an IPv6 device group and IPv6 probes are not supported. Assuming node 210::10:10:14 is dual-homed (i.e. it supports both IPv6 and IPv4 network interfaces and IPv4 node address is 210. 10.10.1), an IPv4 probe can be configured to monitor the health of the node. The below example shows TCP probe configured to be sent to IPv4 address 192.168.2.11 for monitoring health of IPv6 data node 210::10:10:14:

```
switch(config)# feature itd
switch(config)# itd device-group dg-v6
switch(config-device-group)#  node ipv6 210::10:10:11
switch(config-device-group)#  node ipv6 210::10:10:12
```

```
switch(config-device-group)# node ipv6 210::10:10:13
switch(config-device-group)# node ipv6 210::10:10:14
switch(config-dg-node)# probe tcp port 80 ip 192.168.2.11
switch(config-dg-node)# exit
```

This example shows how to configure failaction node per-bucket for a service with ACL:

```
switch(config)# feature itd
switch(config)# itd test
switch(config-itd)# device-group dg_v6
switch(config-itd)# ingress interface vlan66
switch(config-itd)# failaction node per-bucket
switch(config-itd)# access-list ipv6 acl_v6
switch(config-itd)# no shut
```

This example shows how to configure exclude ACL for ITD service. In the below example, an exclude ACL 'exclude-SMTP-traffic' is configured to exclude SMTP traffic from ITD redirection.:

```
switch(config)# feature itd
switch(config)# itd test
switch(config-device-group)# device-group dg
switch(config-itd)# ingress interface Po-1
switch(config-itd)# vrf RED
switch(config-itd)# exclude access-list exclude-SMTP-traffic
switch(config-idt)# no shut
```

This example shows how to configure VRF for ITD service:

```
switch(config)# feature itd
switch(config)# itd test
switch(config-itd)# device-group dg
switch(config-itd)# ingress interface Po-1
switch(config-itd)# vrf RED
switch(config-idt)# no shut
```

This example shows how to enable statistics collection for ITD service:

Note You must enable statistics collection for 'show itd statistics' to show the packet counters.

```
switch(config)# itd statistics test
```

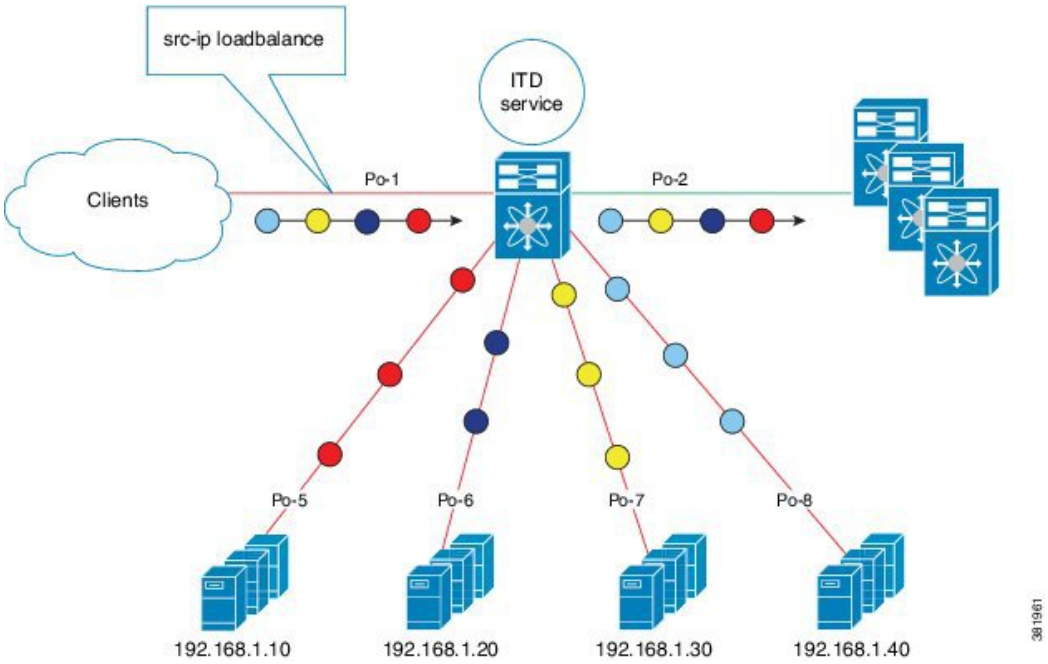
This example shows how to disable statistics collection for ITD service:

```
switch(config)# no itd statistics test
```

Configuration Example: One-Arm Deployment Mode

The configuration below uses the topology in the following figure:

Figure 7: One-Arm Deployment Mode



Step 1: Define device group

```
switch(config)# itd device-group DG
switch(config-device-group)# probe icmp
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
```

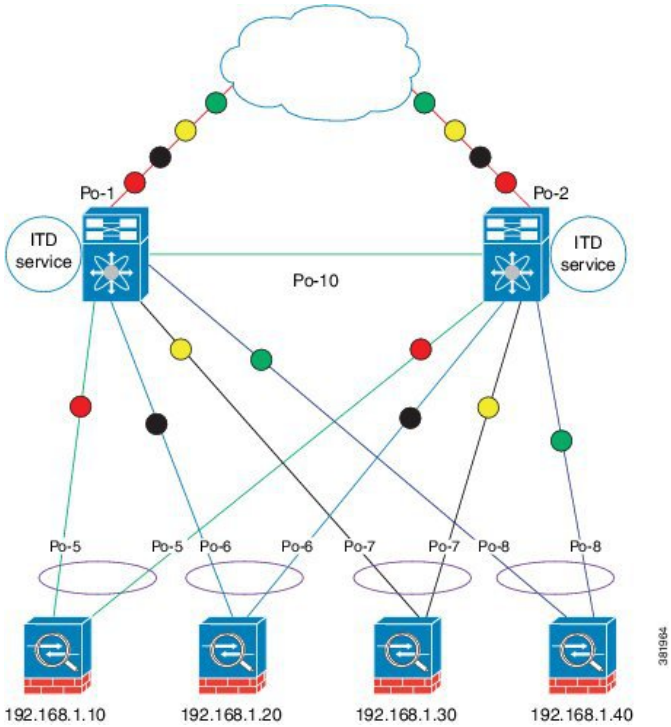
Step 2: Define ITD service

```
switch(config)# itd Service1
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 8: One-Arm Deployment Mode with VPC



Device 1

Step 1: Define device group

```
N7k-1(config)# itd device-group DG
N7k-1s(config-device-group)# probe icmp
N7k-1(config-device-group)# node ip 192.168.2.11
N7k-1(config-device-group)# node ip 192.168.2.12
N7k-1(config-device-group)# node ip 192.168.2.13
N7k-1(config-device-group)# node ip 192.168.2.14
```

Step 2: Define ITD service

```
N7k-1(config)# itd Service1
N7k-1(config-itd)# ingress interface port-channel 1
N7k-1(config-itd)# device-group DG
N7k-1(config-itd)# no shutdown
```

Device 2

Step 1: Define device group

```
N7k-2(config)# itd device-group DG
N7k-2(config-device-group)# probe icmp
N7k-2(config-device-group)# node ip 192.168.2.11
N7k-2(config-device-group)# node ip 192.168.2.12
N7k-2(config-device-group)# node ip 192.168.2.13
N7k-2(config-device-group)# node ip 192.168.2.14
```

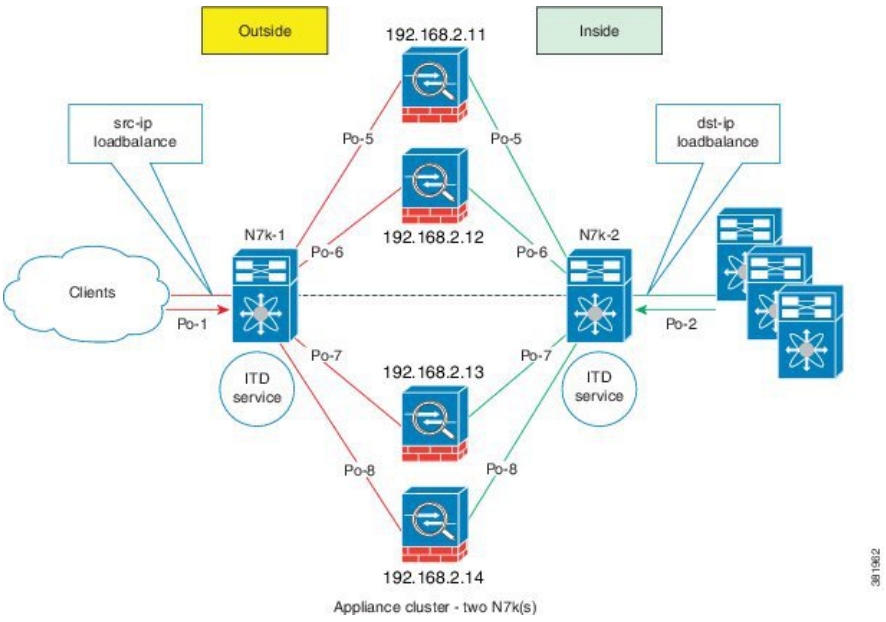
Step 2: Define ITD service

```
N7k-2(config)# itd Service1
N7k-2(config-itd)# ingress interface port-channel 2
N7k-2(config-itd)# device-group DG
N7k-2(config-itd)# no shutdown
```

Configuration Example: Sandwich Deployment Mode

The configuration below uses the topology in the following figure:

Figure 9: Sandwich Deployment Mode



Device 1

Step 1: Define device group

```
N7k-1(config)# itd device-group DG
N7k-1s(config-device-group)# probe icmp
N7k-1(config-device-group)# node ip 192.168.2.11
N7k-1(config-device-group)# node ip 192.168.2.12
N7k-1(config-device-group)# node ip 192.168.2.13
N7k-1(config-device-group)# node ip 192.168.2.14
```

Step 2: Define ITD service

```
N7k-1(config)# itd HTTP
N7k-1(config-itd)# ingress interface port-channel 1
N7k-1(config-itd)# device-group DG
N7k-1(config-itd)# load-balance method src ip
N7k-1(config-itd)# no shutdown
```

Device 2

Step 1: Define device group

```
N7k-2(config)# itd device-group DG
N7k-2(config-device-group)# probe icmp
N7k-2(config-device-group)# node ip 192.168.2.11
N7k-2(config-device-group)# node ip 192.168.2.12
N7k-2(config-device-group)# node ip 192.168.2.13
N7k-2(config-device-group)# node ip 192.168.2.14
```

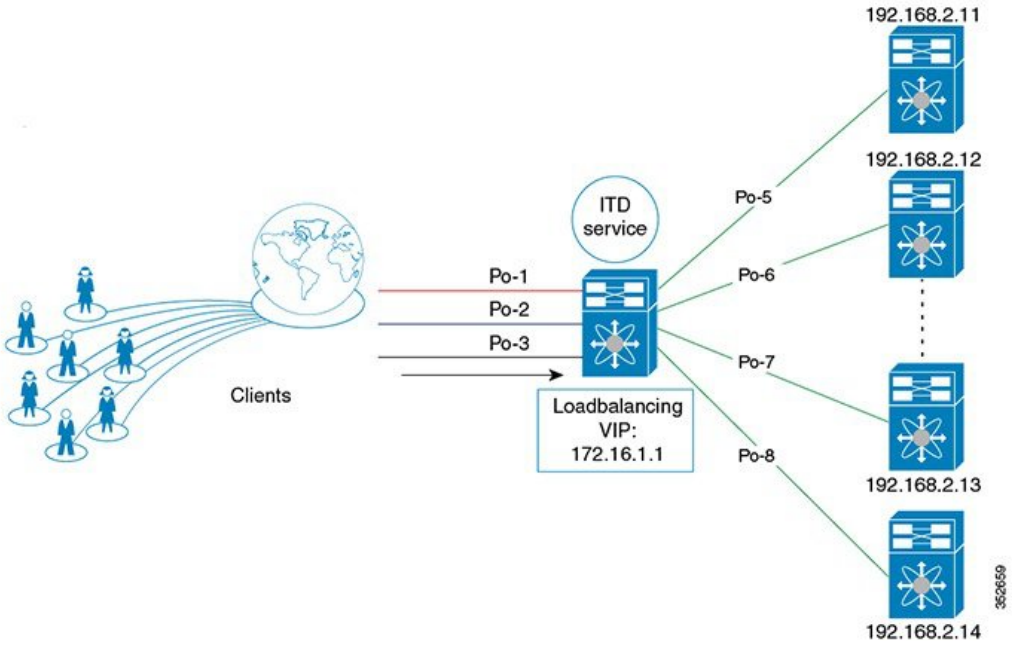
Step 2: Define ITD service

```
N7k-2(config)# itd HTTP
N7k-2(config-itd)# ingress interface port-channel 2
N7k-2(config-itd)# device-group DG
N7k-2(config-itd)# load-balance method dst ip
N7k-2(config-itd)# no shutdown
```

Configuration Example: Server Load-Balancing Deployment Mode

The configuration below uses the topology in the following figure:

Figure 10: ITD Load Distribution with VIP



Step 1: Define device group

```
switch(config)# itd device-group DG
switch(config-device-group)# probe icmp
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
```

Step 2: Define ITD service

```
switch(config)# itd Service2
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
```

Related Documents for ITD

Related Topic	Document Title
Intelligent Traffic Director commands	*Cisco Nexus 7000 Series NX-OS Intelligent Traffic Director Command Reference*

Standards for ITD

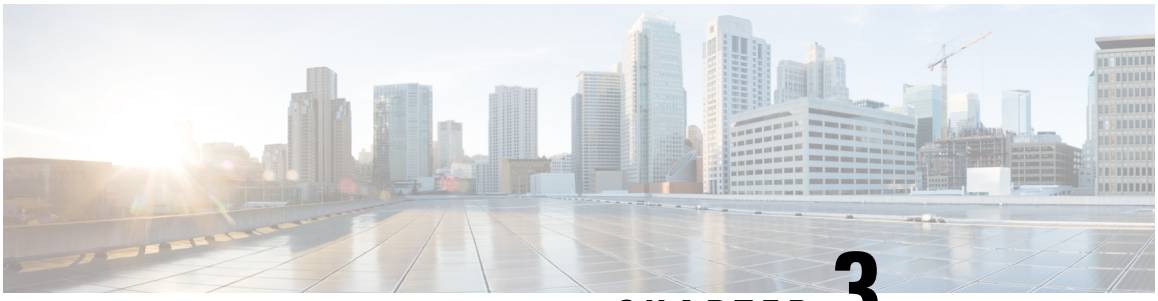
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.

Feature History for ITD

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

Feature Name	Release	Feature Information
ITD	8.4(2)	ITD multi ACL is enhanced to support upto 2000 ACEs.
Multiple access lists and multiple device group per ITD service	8.4(1)	This feature was introduced.
Pre-fetch Optimization	8.4(1)	The ITD failaction feature is enhanced to optimally pre-fetch the failaction node per bucket.
Pre-fetch Optimization	8.3(1)	The ITD failaction feature is enhanced to optimally pre-fetch the status of the service nodes before reassigning the failed nodes to active nodes.
Include ACL	7.3(0)D1(1)	This feature was introduced.
Optimized Node Insertion/Removal	7.3(0)D1(1)	This feature was introduced.

Feature Name	Release	Feature Information
Destination NAT	7.2(1)D1(1)	This feature was introduced.
Multiple Device-Groups within an ITD Service	7.2(1)D1(1)	This feature was introduced.
ITD	7.2(0)D1(1)	Added the following enhancements: • Node-level probe. • IPv4 control probe for IPv6 data node. • Exclude ACL to exclude traffic from redirection.
ITD	6.2(10)	Added the following enhancements: • Weighted load-balancing. • Node-level standby. • Layer 4 port load-balancing. • Sandwich mode node-state synchronization across two VDCs on the same device. • DNS probe. • Start/stop/clear ITD statistics collection. • VRF support for the ITD service and probes.
Intelligent Traffic Director (ITD)	6.2(8)	This feature was introduced.



CHAPTER 3

Deployment and Best Practices

Design and Deployment Considerations

This section describes the design and deployment considerations in ITD.

Number of ITD Services

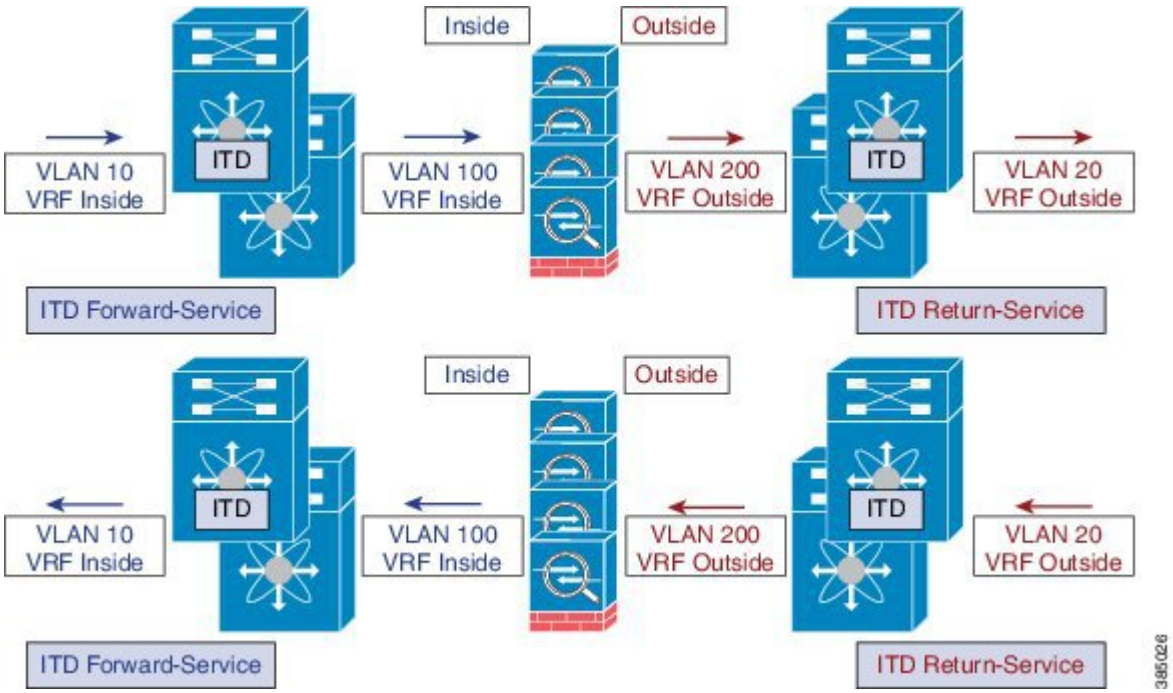
An ITD Service configuration defines the ITD traffic distribution for a particular direction of the traffic flow. If both directions of a flow are required to be redirected, two ITD services should be configured: one for the forward, and another for the return traffic flow. Because an ASA has different Inside and Outside interface IP addresses, two different device-groups should be configured that point to the corresponding Inside and Outside IP addresses.

Additional ASA VLANs

The ITD Forward and Return services are attached to the inside and outside VLAN SVIs on the Nexus switch. To enable a security application, such as a firewall, requires that all traffic is examined, 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 ASA interfaces are configured on the same VLANs as that of the switch, the traffic going back to switch from the firewall is redirected back to the ASA due to the presence of an ITD service on other VLAN on the switch. So, a pair of separate VLANs should be used to prevent traffic looping between the firewalls and the Nexus Switches.

Figure 11: Logical View of the ITD-ASA Deployment

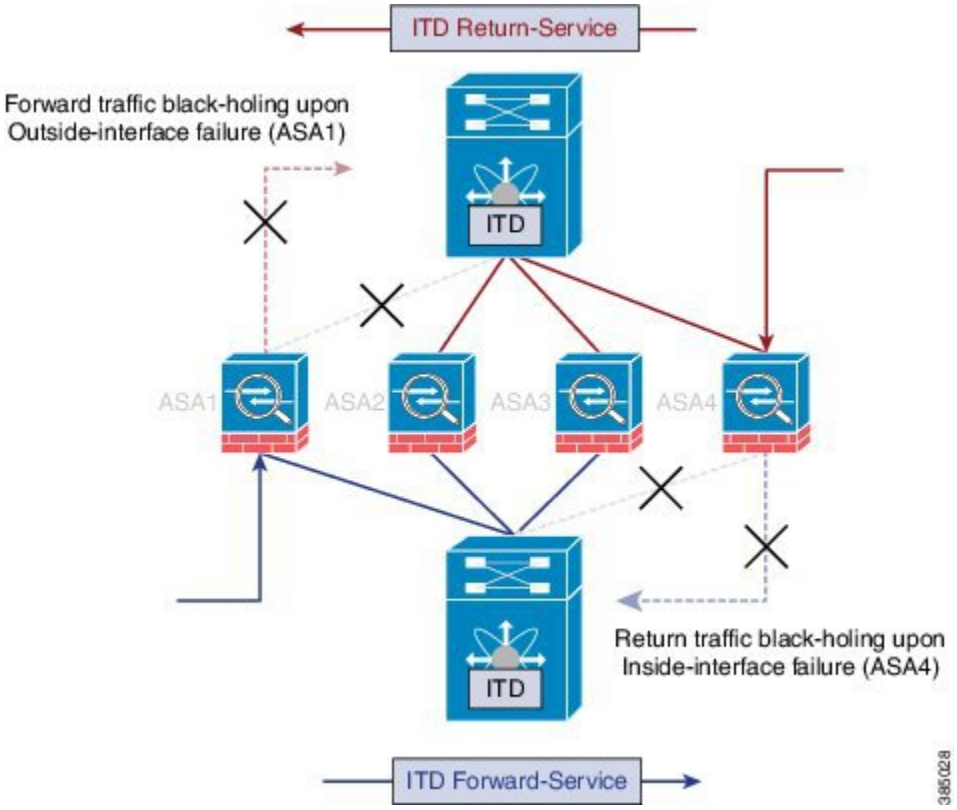


The above example shows the VLANs 10 and 20 being the inside and outside interfaces towards the source and destination on the network, and VLANs 100 and 200 being used towards the ASAs to enable loop-free traffic.

Link-Failure Scenario

When one of the interfaces of the ASA, either inside or outside, fails, then traffic coming into the other side of that ASA is blackholed as the egress interface for traffic is down. The ITD Peer-VDC Node-State Sync feature resolves this issue by removing the remote side of the ASA from ITD by synchronizing the node-states across the VDCs.

Figure 12: ASA Failure Scenario Without Peer-VDC Synchronization



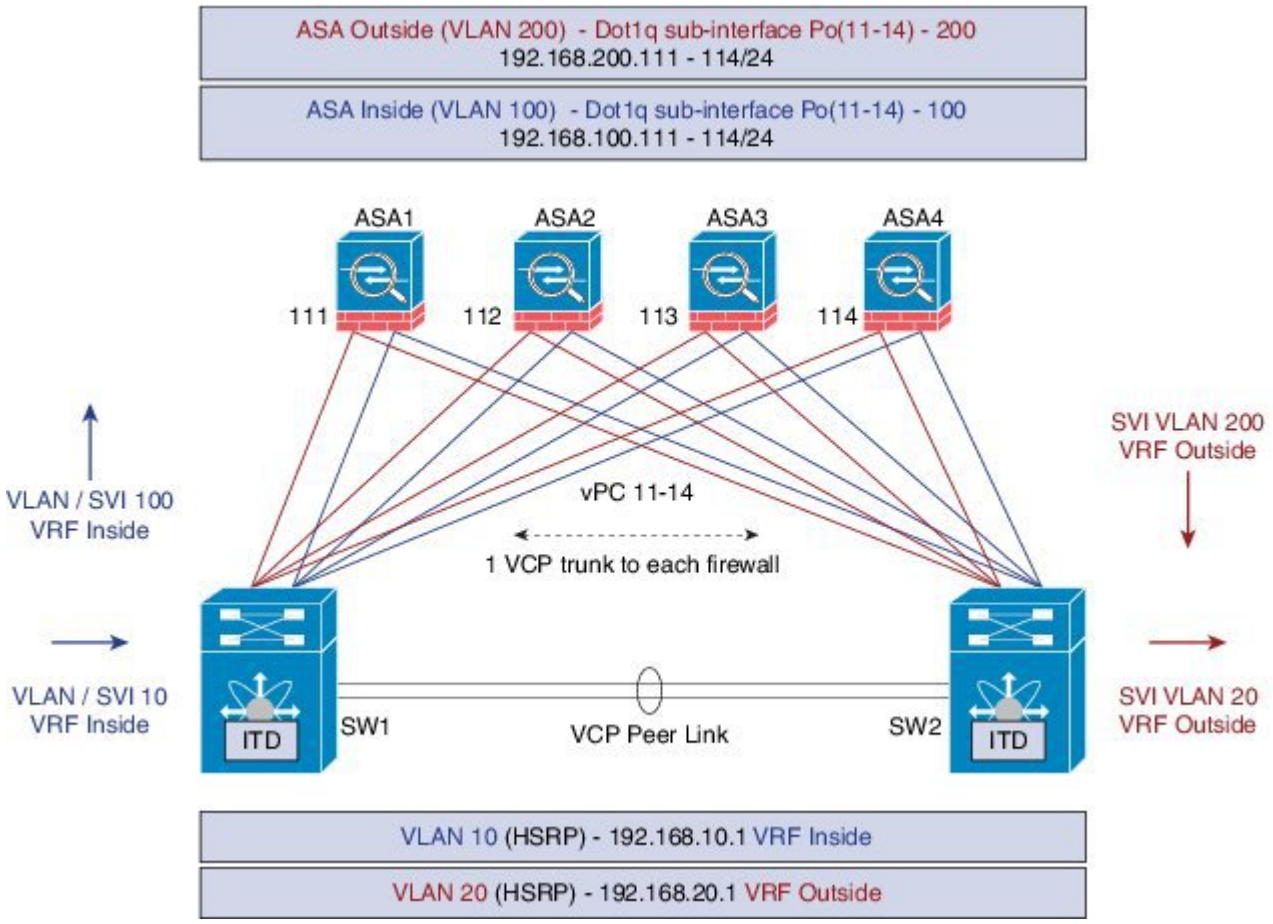
The ITD Peer-VDC Node-State-Sync feature currently supported only in the Dual VDC non-vPC single switch topology. ASA-Clustering also solves this problem as the clustering ensures the ASA is fully brought down in case of such failures. The Firewall-on-a-stick implementation, either single link or vPC, does not experience this issue as the inside and outside interfaces on the ASA belong to the same physical or virtual interface.

Deployment of ITD ASA

Configuration Example: Firewall on a Stick

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

Figure 13: Firewall on a Stick with vPC



The following is the sample configuration snippets of the Nexus 7000. The example shows partial configurations from a switch (sw1). The configuration must be to be extended appropriately towards 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 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 192.168.20.1

interface vlan100
description Inside_Vlan_to_ASA
vrf member INSIDE
ip address 192.168.100.10/24
hsrp 100
ip 192.168.100.1
```

```
interface vlan200
description Outside_Vlan_to_ASA
vrf member OUTSIDE
ip address 192.168.200.10/24
hsrp 200
ip 192.168.200.1

.............................

interface Port-Channel111
 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-Channel41
 description Downstream_vPC_to_Network
 switchport mode trunk
 switchport trunk allowed vlan 10,20
 vpc 41
 no shutdown

interface Port-Channel 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 192.168.100.111
   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 INSIDE
 vrf INSIDE
```

```
  #applies ITD service to VRF "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 a 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 Dst-IP
no shutdown

itd OUTSIDE
 vrf OUTSIDE
  #applies ITD service to VRF "OUTSIDE"
device-group FW_OUTSIDE
ingress interface Vlan 10
failaction node reassign
load-balance method dst ip buckets 16
#distributes traffic into 16 buckets
#load balances traffic based on Destination-IP.
#OUTSIDE service uses Dst-IP
no shutdown
```

The following is the configuration snippets of ASA. The ASA side of the configuration is show below from one ASA(ASA-1). Similar configuration must be extended to all the other ASAs.

```
interface Port-Channel11
 nameif aggregate
 security-level 100
 no ip address
!
interface Port-Channel11.100
 description INSIDE
 vlan 100
 nameif inside
 security-level 100
 ip address 192.168.100.111 255.255.255.0
!
interface Port-Channel11.200
 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 TenGigabitEthernet0/6
 description CONNECTED_TO_SWITCH_A_VPC
 channel-group 11 mode active
 no nameif
 no security-level

interface TenGigabitEthernet0/7
 description CONNECTED_TO_SWITCH_B_VPC
 channel-group 11 mode active
 no nameif
 no security-level
!
```

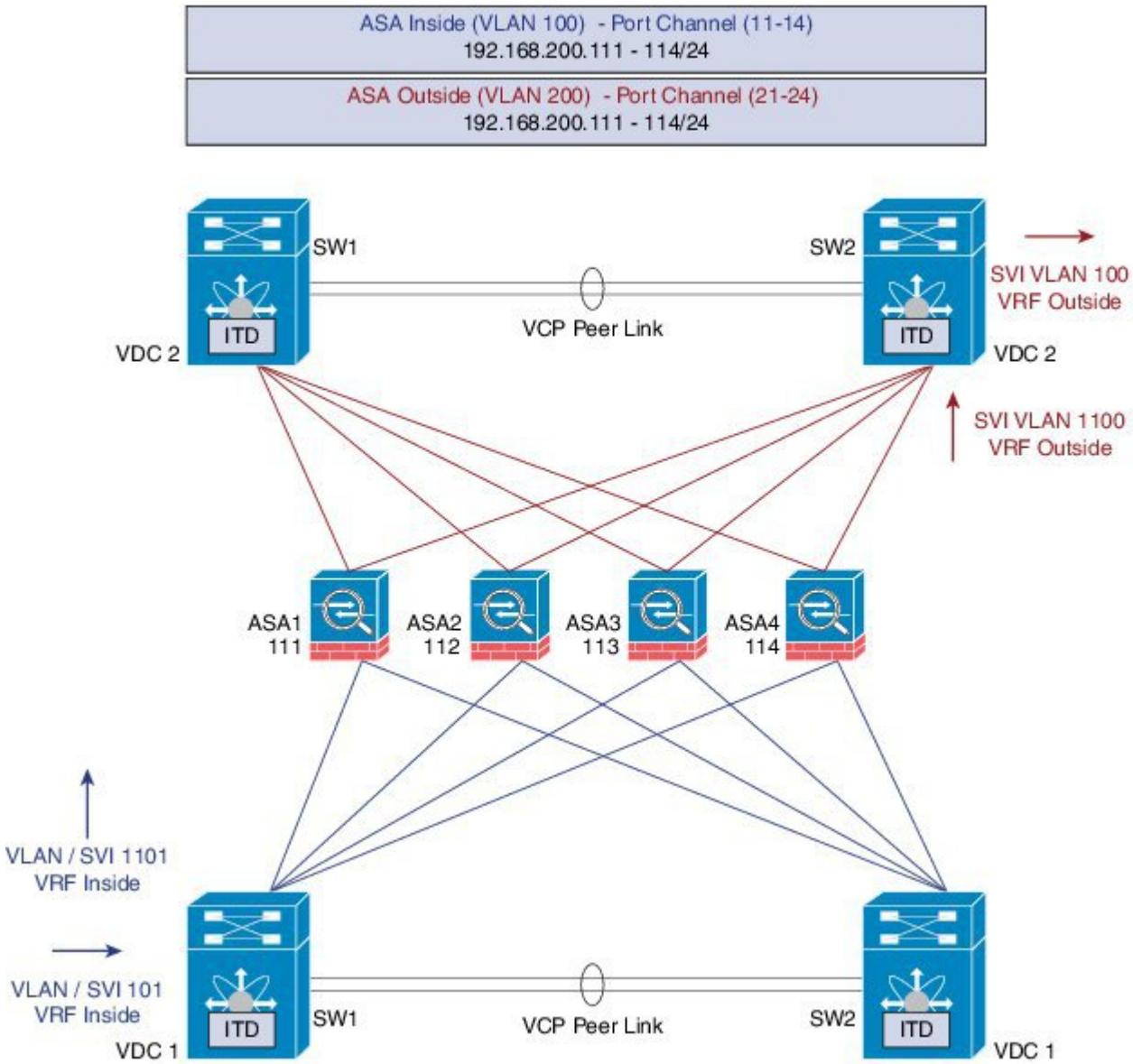
Note the following points from the above configuration and topology:

- VLANs 10, 20,100, and, 200 and their SVI mapping to appropriate VRFs.
- ITD device-group configuration for ASA - inside and outside.
- ITD – load-balancing configuration for achieving flow-symmetry.
- In a vPC scenario, as long as one member of the vPC is up, there will be no change to the ITD. The ITD redirection on the switch with the failed vPC leg traverses the peer-switch through the peer-link as in a typical vPC case.
- This topology and deployment method does not blackhole traffic when a physical link failure occurs, as the inside and outside interfaces are tied to the same physical or virtual interface on the ASA (dot1q sub-interfaces).
- To support Routing Protocol neighborship over vPC (Cisco NX-OS 7.2(0)D1(1) and later releases), the command **layer3 peer-router** should be configured within the vPC domain.
- The 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 towards ASAs via PBR, thus routes are not needed.

Configuration Example: Firewall in Dual VDC Sandwich Mode with vPC

Sandwich mode with vPC the Inside and Outside ASA interfaces are each assigned to separate port-channel bundles. This topology illustrated in the figure below, with Nexus 7000 currently does not support the node-state synchronization feature. As a consequence of vPC, a single link failure does not impede traffic-flow and ITD will continue to forward throught the peer-switch's link towards the ASA, similar to the other scenarios with vPC.

Figure 14: Firewall in Dual VDC Two Switch Sandwich Mode with vPC



Configuration steps in Nexus 7000

The main differences in this topology from the single switch topology are that there are vPC port-channels instead of single links between the Nexus switch and the ASA. Secondly, as in the previous case, the inside and outside interfaces on the switches are configured in different VDCs.

The following is the configuration from the VDC1:

```
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-Channel11
 description To_ASA-1-INSIDE
 switchport mode access
 switchport access vlan 100
 vpc 11

interface Ethernet4/1
 description To_ASA-1-INSIDE
 switchport mode access
 switchport access vlan 100
 channel-group 11 mode active
```

The following is the configuration from the VDC2:

```
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-Channel21
 description To_ASA-1-OUTSIDE
 switchport mode access
 switchport access vlan 200
 vpc 11

interface Ethernet4/25
 description To_ASA-1-OUTSIDE
 switchport mode access
 switchport access vlan 200
 channel-group 21 mode active
```

Configuration steps in ASA

The following is the configuration snippet from the ASA.

```
interface Port-Channel11
 description INSIDE
 vlan 100
 nameif inside
 security-level 100
 ip address 192.168.100.111 255.255.255.0

interface Port-Channel21
 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 TenGigabitEthernet0/6
 description CONNECTED_TO_SWITCH0-A-VPC
 channel-group 11 mode active
 no nameif
```

```
 no security-level

interface TenGigabitEthernet0/7
 description CONNECTED_TO_SWITCH-B-VPC
 channel-group 11 mode active
 no nameif
 no security-level

interface TenGigabitEthernet0/8
 description CONNECTED_TO_SWITCH-A-VPC
 channel-group 21 mode active
 no nameif
 no security-level

interface TenGigabitEthernet0/9
 description CONNECTED_TO_SWITCH-B-VPC
 channel-group 21 mode active
 no nameif
 no security-level
```

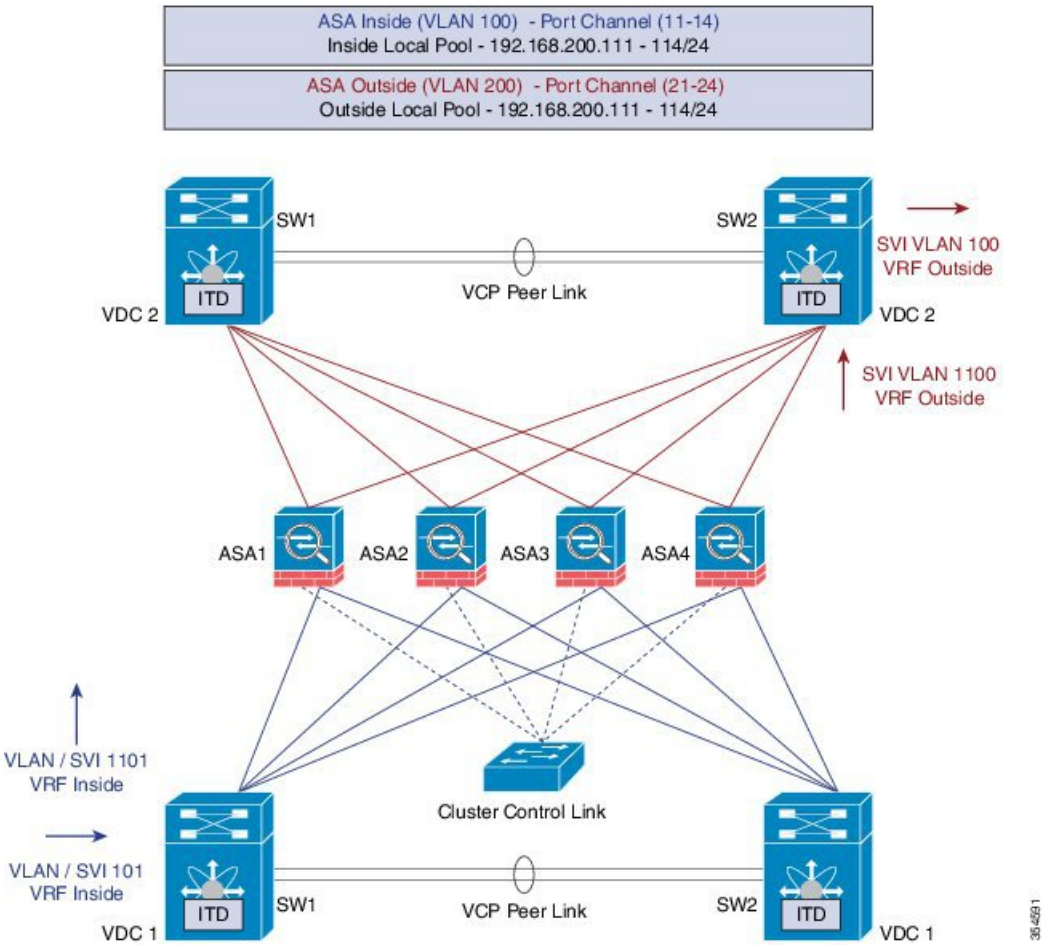
Note the following points from the above configuration and topology:

- ITD – Load-balancing configuration for achieving flow-symmetry.
- In a vPC scenario, as long as one member of the vPC is up, there will be no change to ITD. The ITD redirection on the switch with the failed vPC leg would now traverse the peer-switch through the peer-link as in a typical vPC case.
- In this topology or deployment method blackholing of traffic can occur, if one of the port channels on the ASA, or single physical link in non-VPC case, fails.
- To support Routing Protocol neighborship over vPC, in Cisco NX-OS 7.2(0)D1(1) and later releases, the command **layer3 peer-router** should be configured within the vPC domain.
- Traffic is directed towards ASAs via PBR, 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, in terms of management and integration into a network, while achieving the increased throughput and redundancy of multiple devices. Refer the figure below.

Figure 15: ASA Cluster with Dual VDC Sandwich with vPC



ACL Clustering

The following table is a summary comparison of the impact on CCL that occurs with ECMP versus the impact that occurs with ITD, when the ASA device status changes:

ASA Status	ITD	ECMP
Steady state	Minimal traffic on CCL. Expected traffic types. Exactly same load-distribution irrespective of the type of line card and switch.	Minimal traffic on CCL if same line card type and switch model is used everywhere. If differing hardware is used, a higher level of asymmetry may occur causing traffic on the CCL network. Each hardware has different hash function. Two switches (eg in vPC) might send same flow to different ASA, causing CCL traffic.

ASA Status	ITD	ECMP
Single ASA failure	No additional traffic on CCL. ITD offers IP stickiness and Resilient Hashing.	All flows are rehashed and additional traffic redirection occurs on CCL. This may effect a degree of traffic to all ASAs in the cluster.
Single ASA recovery	Traffic redirection can occur on the CCL between two ASAs in the cluster, the recovered ASA that receives a bucket and the ASA that serviced that bucket prior.	Additional traffic redirection can occur on CCL. This may effect a degree of traffic to all ASAs in the cluster.
ASA addition	Minimal additional traffic on CCL.	All flows are rehashed and additional traffic redirection occurs on CCL. This may effect a degree of traffic to all ASAs in the cluster.

ITD can load balance to individual mode Layer 3 (L3) ASA clusters. ITD is complimentary to clustering in that ITD provides predictability of knowing which flows handled by each firewall. ITD buckets determine this instead of relying on OSPF ECMP and Port-Channel hashing algorithms.

With L3 clusters, the flow owner can be pre-determined based on the bucket allocation. Without ITD and L3 clustering the initial choice of Owner is typically unpredictable however with ITD this can be predetermined.

ASA clustering also uses the implementation of 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 causes all flows in 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 appropriate backup owner for the flows it receives, it should receive the flow state information from the backup owner and process traffic seamlessly, for more information, refer the Cisco ASA Series CLI Configuration Guide, 9.0.

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, there may be a firewall performance improvement with using non-clustered firewalls. However, the assurance of knowing that existing connections should not timeout if an ASA cluster member were to fail may be of greater value to customers.

Configuration steps in Nexus 7000

Introduction of Clustering does not change the ITD configuration. The ITD Nexus configuration depends on the type of topology, and in this example it is the same as the Firewall with Dual VDC Sandwich with vPC topology.

The ITD configuration remains similar to the previous method except that the node-state synchronization are removed.

Configuration steps in ASA

The ASA clustering is configured as an L3 cluster, similar to the PBR Deployment Scenario, described in the following document. The detailed information can be found at the link below regarding ASA Cluster configuration. The following is a sample configuration from the ASA is shown below for the Firewall in Layer 3 Clustering topology, for more information, refer: Cisco ASA Series CLI Configuration Guide, 9.0

```
cluster group ASA-CLUSTER-L3
```

```
local-unit ASA1
cluster-interface port-channel1 ip 192.168.250.100 255.255.255.0
priority 1
health-check holdtime 1.5
clacp system-mac auto system-priority 1
enable

mac-address pool MAC-INSIDE aaaa.0101.0001 - aaa.0101.0008
mac-address pool MAC-OUTSIDE aaaa.0100.0001 - aaa.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-Channel11
 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-Channel21
 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-Channel31
 description Clustering Interface
 lacp max-bundle 8


interface TenGigabitEthernet0/6
 channel-group 11 mode active
 no nameif
 no security-level
 no ip address

interface TenGigabitEthernet0/7
 channel-group 11 mode active
 no nameif
 no security-level
 no ip address

interface TenGigabitEthernet0/8
 channel-group 21 mode active
 no nameif
 no security-level
 no ip address

interface TenGigabitEthernet0/9
channel-group 21 mode active
 no nameif
 no security-level
 no ip address

interface TenGigabitEthernet1/0
channel-group 31 mode active
 no nameif
 no security-level
 no ip address

interface TenGigabitEthernet1/1
```

```
channel-group 31 mode active
 no nameif
 no security-level
 no ip address
```

As seen in the above configuration, the Port-Channels 11 and 21 are used for either the inside or outside interfaces as in previous cases. However, there is an additional Port-channel 31 now for the Clustering Interface. Individual interfaces are normal routed interfaces, each with their 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 master unit. Similarly, a mac-address pool is also configured and used under the corresponding Port-channel, either inside or outside.

Configuration Example: ITD for WCCP-Type Scenarios

Design with Web-Proxy

In Web-Proxy deployment with ITD, the Nexus switch takes responsibility of matching the internet-bound Web traffic and Load-balancing it towards the proxy-servers.

The proxy-servers would 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 a group-level or at node-level for redundancy.

Number of Services

As illustrated in the packet-flow slides, ITD redirection is normally only required in the forward direction in the client facing VLAN. Subsequently, the packets get routed/forwarded without any ITD redirection or distribution. ITD with such Web-Proxy deployments would only need one ITD service, and this is configured for the forward direction.However, must there be a requirement for reverse traffic redirection, traffic selection would need to be based on the Source L4 Ports. Flow symmetry also needs to be maintained by reversing the LB parameter.

Probes for Proxy-Health-Monitoring

With ITD for Web-proxy deployments, ITD probes are used to check availability of the Web-Proxy server. This is important as traffic that is sent towards a failed proxy-server will be blackholed. The probes that are available at present, in the respective latest releases per platform, are:

- Nexus 7000 (7.2(1)D1(1)): ICMP, TCP/UDP, DNS, HTTP
- Nexus 5000 : ICMP
- Nexus 9000: ICMP

Traffic Selection Requirements

The following are the currently supported methods for traffic-filtering or traffic-selection for ITD:

- **Virtual IP (Supported on Nexus 5000, Nexus 6000, Nexus 7000 and Nexus 9000):**

  IP + Subnet mask combination used for traffic selection(filtering) for the destination-field only.

- **Exclude ACL:**

  ACL used to specify which traffic should bypass ITD.

  Traffic not permitted by this ACL will go through ITD.

  Exclude ACL can filter based on both Source and Destination fields. Exclude ACL precedes VIP.

Exclude ACL only supports permit ACE entries. deny ACE's are not supported on the exclude ACL.

- **Port-number based filtering**

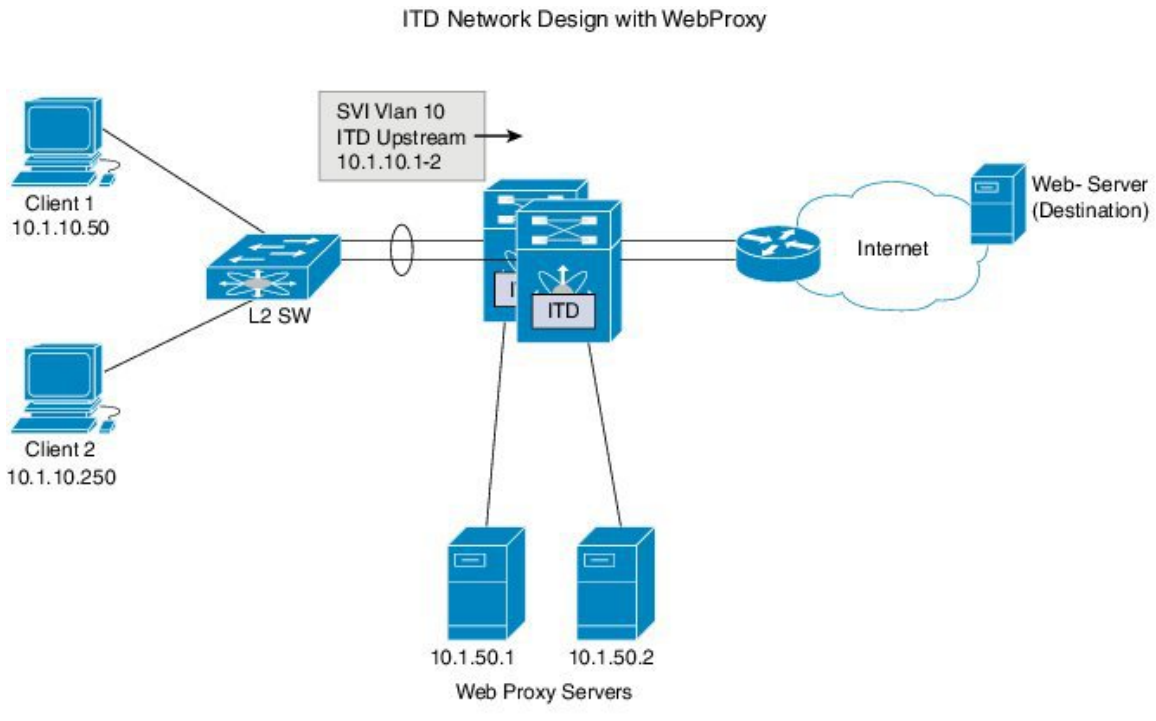
  For selecting traffic based on L4 Ports, for ex. "Port 80 needs ITD service" we can do it today with the following:

  - Matching Destination Ports: VIP – 0.0.0.0 0.0.0.0 tcp 80 (any source or destination IP, destination port 80 matched)
  - Matching Source Ports: Exclude ACL with "permit tcp any neq 80 any" (Any port not 80 will bypass ITD, port 80 is redirected).
  - Matching multiple Port-numbers: Multiple VIP lines in ITD can be configued, one for each port.

- For selecting traffic based on L4 Ports, for ex. "Port 80 needs ITD service" we can do it today with the following:
  - Include ACL used to permit the traffic that should be ITD-serviced. Both SRCand DST fields can be matched.
  - Only Permit lines allowed. Either VIP or Include ACL can be used at a time, but not both.
  - Load-balancing parameter will determine the max. length of the match possible in the include ACL. For example, with source-based LB and 8 buckets, maximum mask of source IP address that can be matched is /29. With destination LB with 8 buckets, maximum mask of destination IP that can be matched is /29.

Note The include ACL feature is a roadmap item and is not available on any current release. The information provided here is tentative only, and is subject to change.

Figure 16: ITD Network Design with WebProxy



As indicated in the above figure, the destination port 80/443 (ingressVLAN10) to Internet is distributed to Web-Proxy servers 10.1.50.1/10.1.50.2.

Traffic on VLAN 10 destined to private networks (10.0.0.0/8, 192.168.0.0/16 and 172.16.0.0/20) is not sent to the proxy server.

```
 itd device-group Web_Proxy_Servers              <<<< Configure ITD Device-group
Web_Proxy_Servers and point to server IP addresses.
  probe icmp
  node ip 10.1.50.1
  node ip 10.1.50.2

ip access-list itd_exclude_ACL                   <<<< Configure Exclude ACL to exclude all
 traffic destined to Private IP addresses.
  10 permit ip any 10.0.0.0 255.0.0.0
  20 permit ip any 192.168.0.0 255.255.0.0
  30 permit ip any 172.16.0.0 255.255.240.0

Itd Web_proxy_SERVICE
  device-group Web_Proxy_Servers                 <<<< Apply Exclude ACL.
  exclude access-list itd_exclude_ACL
  virtual ip 0.0.0.0 0.0.0.0 tcp 80              <<<< Any Traffic TO DESTINATION Port-80
redirect to group  Web_Proxy_Servers
  virtual ip 0.0.0.0 0.0.0.0 tcp 443             <<<< Any Traffic TO DESTINATION Port-443
redirect to group Web_Proxy_Servers
  ingress interface Vlan 10
  failaction node reassign
  load-balance method src ip
  no shutdown
```

When there is a need for return traffic redirection, the following additional configuration is required.

Note Only port filtering is possible using the Layer 4 range' operator. Exclude ACL supports only permit entries.

```
ip access-list itd_exclude_return            <<<< Configure Exclude ACL (Return) to exclude
 all but port 80 & 443
  10 permit tcp any range 0 79 any
  20 permit tcp any range 81 442 any
  10 permit tcp any range 444 65535 any

itd Web_proxy_SERVICE                          <<<< Configure Return ITD service for return
 traffic:
  device-group Web_Proxy_Servers
  exclude access-list itd_exclude_return      <<<< Apply Exclude ACL for Return ITD service.
  ingress interface Vlan 20                     <<<<  Internet-facing ingress interface on
the Nexus Switch.
  failaction node reassign
  load-balance method dst ip                    <<<<  Flow symmetry between forward/retrun
flow achieved by flipping LB parameter.
  no shutdown
```

As seen in the above configuration, the Port-Channels 11 and 21 are used for either the inside or outside interfaces as in previous cases. However, there is an additional Port-channel 31 now for the clustering interface. Individual interfaces are normal routed interfaces, each with their 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 master unit. Similarly, a mac-address pool is also configured and used under the corresponding Port-channel, either inside or outside.