Configuring Static and Dynamic NAT Translation

- Network Address Translation Overview, on page 1
- Information About Static NAT, on page 2
- Dynamic NAT Overview, on page 3
- Timeout Mechanisms, on page 3
- NAT Inside and Outside Addresses, on page 5
- Pool Support for Dynamic NAT, on page 5
- Static and Dynamic Twice NAT Overview, on page 6
- VRF Aware NAT, on page 7
- Guidelines and Limitations for Static NAT, on page 8
- Restrictions for Dynamic NAT, on page 9
- Guidelines and Limitations for Dynamic Twice NAT, on page 11
- Guidelines and Limitations for TCP Aware NAT, on page 11
- Configuring Static NAT, on page 12
- Configuring Dynamic NAT, on page 22

Network Address Translation Overview

Network Address Translation (NAT) enables private IP internetworks that use nonregistered IP addresses to connect to the Internet. NAT operates on a device, usually connecting two networks, and translates private (not globally unique) IP addresses in the internal network into legal IP addresses before packets are forwarded to another network. You can configure NAT to advertise only one IP address for the entire network to the outside world. This ability provides additional security, effectively hiding the entire internal network behind one IP address.

A device configured with NAT has at least one interface to the inside network and one to the outside network. In a typical environment, NAT is configured at the exit router between a stub domain and a backbone. When a packet leaves the domain, NAT translates the locally significant source IP address into a globally unique IP address. When a packet enters the domain, NAT translates the globally unique destination IP address into a local IP address. If more than one exit point exists, NAT configured at each point must have the same translation table.

NAT is described in RFC 1631.

Information About Static NAT

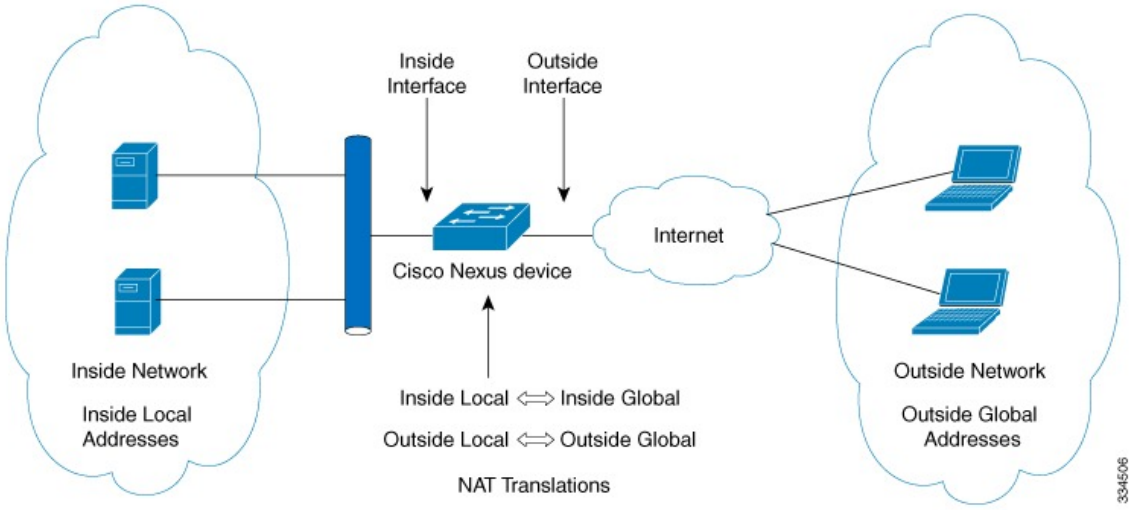
Static Network Address Translation (NAT) allows the user to configure one-to-one translations of the inside local addresses to the outside global addresses. It allows both IP addresses and port number translations from the inside to the outside traffic and the outside to the inside traffic. The Cisco Nexus device supports Hitless NAT, which means that you can add or remove a NAT translation in the NAT configuration without affecting the existing NAT traffic flows.

Static NAT creates a fixed translation of private addresses to public addresses. Because static NAT assigns addresses on a one-to-one basis, you need an equal number of public addresses as private addresses. Because the public address is the same for each consecutive connection with static NAT, and a persistent translation rule exists, static NAT enables hosts on the destination network to initiate traffic to a translated host if an access list exists that allows it .

With dynamic NAT and Port Address Translation (PAT), each host uses a different address or port for each subsequent translation. The main difference between dynamic NAT and static NAT is that static NAT allows a remote host to initiate a connection to a translated host if an access list exists that allows it, while dynamic NAT does not.

The figure shows a typical static NAT scenario. The translation is always active so both translated and remote hosts can originate connections, and the mapped address is statically assigned by the **static** command.

Figure 1: Static NAT



These are key terms to help you understand static NAT:

- NAT inside interface—The Layer 3 interface that faces the private network.
- NAT outside interface—The Layer 3 interface that faces the public network.
- Local address—Any address that appears on the inside (private) portion of the network.
- Global address—Any address that appears on the outside (public) portion of the network.
- Legitimate IP address—An address that is assigned by the Network Information Center (NIC) or service provider.

- Inside local address—The IP address assigned to a host on the inside network. This address does not need to be a legitimate IP address.
- Outside local address—The IP address of an outside host as it appears to the inside network. It does not have to be a legitimate address, because it is allocated from an address space that can be routed on the inside network.
- Inside global address—A legitimate IP address that represents one or more inside local IP addresses to the outside world.
- Outside global address—The IP address that the host owner assigns to a host on the outside network. The address is a legitimate address that is allocated from an address or network space that can be routed.

Dynamic NAT Overview

Dynamic Network Address Translation (NAT) translates a group of real IP addresses into mapped IP addresses that are routable on a destination network. Dynamic NAT establishes a one-to-one mapping between unregistered and registered IP addresses; however, the mapping can vary depending on the registered IP address that is available at the time of communication.

A dynamic NAT configuration automatically creates a firewall between your internal network and outside networks or the Internet. Dynamic NAT allows only connections that originate inside the stub domain—a device on an external network cannot connect to devices in your network, unless your device has initiated the contact.

Dynamic NAT translations do not exist in the NAT translation table until a device receives traffic that requires translation. Dynamic translations are cleared or timed out when not in use to make space for new entries. Usually, NAT translation entries are cleared when the ternary content addressable memory (TCAM) entries are limited. The default minimum timeout for dynamic NAT translations is 30 minutes.

Note The **ip nat translation sampling-timeout** command is not supported. Statistics are collected every 60 seconds for the installed NAT policies. These statistics are used to determine if the flow is active or not.

Dynamic NAT supports Port Address Translation (PAT) and access control lists (ACLs). PAT, also known as overloading, is a form of dynamic NAT that maps multiple unregistered IP addresses to a single registered IP address by using different ports. Your NAT configuration can have multiple dynamic NAT translations with same or different ACLs. However, for a given ACL, only one interface can be specified.

Timeout Mechanisms

The following NAT translation timeout timers are supported on the switch:

- **syn-timeout** - Timeout value for TCP data packets that send the SYN request, but do not receive a SYN-ACK reply.

  The timeout value ranges from 1 second to 172800 seconds. The default value is 60 seconds.

Note The **syn-timeout** option is supported only on Cisco Nexus 9200 and 9300-EX, -FX, -FX2, -FX3, -FXP, -GX platform switches.

- **finrst-timeout** - Timeout value for the flow entries when a connection is terminated by receiving RST or FIN packets. Use the same keyword to configure the behavior for both RST and FIN packets.

  The timeout value ranges from 1 second to 172800 seconds. The default value is 60 seconds.

  - If a FIN packet is received after the connection is established, SYN-->SYN-ACK-->FIN, the finrst timer starts.
  - If a FIN-ACK is received from the other side, the translation entry is cleared immediately, else it clears after the timeout value completes.
  - If an RST packet is received after the connection is established, SYN-->SYN-ACK-->RST, the translation entry is cleared immediately.

Note The **finrst-timeout** option is supported only on Cisco Nexus 9200 and 9300-EX, -FX, -FX2, -FX3, -FXP, -GX platform switches.

- **tcp-timeout** - Timeout value for TCP translations for which connections have been established after a three-way handshake (SYN, SYN-ACK, ACK). If no active flow occurs after the connection has been established, the translations expire as per the configured timeout value.

  The timeout value ranges from 60 seconds to 172800 seconds. The default value is 3600 seconds.

- **udp-timeout** - Timeout value for all NAT UDP packets.

  The timeout value ranges from 60 seconds to 172800 seconds. The default value is 3600 seconds.

- **timeout** - Timeout value for dynamic NAT translations.

  The timeout value ranges from 60 seconds to 172800 seconds. The default value is 3600 seconds.

- **icmp-timeout** - Timeout value for ICMP packets.

  The timeout value ranges from 60 seconds to 172800 seconds. The default value is 3600 seconds.

- **sampling-timeout** - Time after which the device checks for dynamic translation activity.

  The timeout value ranges from 900 seconds to 172800 seconds.

Note When you create dynamic entries without timeouts configured, they take the default timeout of 3600 seconds. After you change the default timeout values to the new values, the translation entries created after will pick up the latest timeout values.

The **udp-timeout** and the **timeout** value timers are triggered after the timeout configured for the **ip nat translation sampling-timeout** command expires.

Note There are three different options that can be configured for aging:

- Time-out: This is applicable for all type of flows(both TCP and UDP).
- TCP TIME-OUT: This is applicable for only TCP flows.
- UDP TIME-OUT: This is applicable for only UDP flows.

Note When you create dynamic entries without timeouts configured, they take the default timeout of one hour (60 minutes). If you enter the **clear ip nat translations all** command after configuring timeouts, the configured timeout take effect. A timeout can be configured from 60 to 172800 seconds.

NAT Inside and Outside Addresses

NAT inside refers to networks owned by an organization that must be translated. When NAT is configured, hosts within this network will have addresses in one space (known as the local address space) that will appear to those outside the network as being in another space (known as the global address space).

Similarly, NAT outside refers to those networks to which the stub network connects. They are not generally under the control of the organization. Hosts in outside networks can be subject to translation and can have local and global addresses.

NAT uses the following definitions:

- Local address—A local IP address that appears on the inside of a network.
- Global address—A global IP address that appears on the outside of a network.
- Inside local address—The IP address that is assigned to a host on the inside network. The address is probably not a legitimate IP address assigned by the Internet Network Information Center (InterNIC) or a service provider.
- Inside global address—A legitimate IP address (assigned by InterNIC or a service provider) that represents one or more inside local IP addresses to the outside world.
- Outside local address—The IP address of an outside host as it appears to the inside network. The address is not necessarily legitimate; it was allocated from the address space that is routable on the inside.
- Outside global address—The IP address that is assigned to a host on the outside network by the owner of the host. The address was allocated from a globally routable address or a network space.

Pool Support for Dynamic NAT

Cisco NX-OS provides pool support for dynamic NAT. Dynamic NAT allows the configuration of a pool of global addresses that can be used to dynamically allocate a global address from the pool for every new translation. The addresses are returned to the pool after the session ages out or is closed. This allows for a more efficient use of addresses based on requirements.

Support for PAT includes the use of the global address pool. This further optimizes IP address utilization. PAT exhausts one IP address at a time with the use of port numbers. If no port is available from the appropriate group and more than one IP address is configured, PAT moves to the next IP address and gets the allocation based on the user defined pool (ignoring the source port or attempting to preserve it).

With dynamic NAT and PAT, each host uses a different address or port for each subsequent translation. The main difference between dynamic NAT and static NAT is that static NAT allows a remote host to initiate a connection to a translated host if an access list exists that allows it, while dynamic NAT does not.

When dynamic NAT is configured to use a pool of IP addresses, that are not locally available or configured locally, the out-to-in traffic is considered as DEST MISS. Due to this behavior, the **`show system internal access-list dest-miss stats`** command output displays increment in DEST MISS counters. The DEST MISS statistics is supported from Cisco NX-OS Release 9.3(5) onwards.

Static and Dynamic Twice NAT Overview

When both the source IP address and the destination IP address are translated as a single packet that goes through a Network Address Translation (NAT) device, it is referred to as twice NAT. Twice NAT is supported for static and dynamic translations.

Twice NAT allows you to configure two NAT translations (one inside and one outside) as part of a group of translations. These translations can be applied to a single packet as it flows through a NAT device. When you add two translations as part of a group, both the individual translations and the combined translation take effect.

A NAT inside translation modifies the source IP address and port number when a packet flows from inside to outside. It modifies the destination IP address and port number when the packet returns from outside to inside. NAT outside translation modifies the source IP address and port number when the packet flows from outside to inside, and it modifies the destination IP address and port number when the packet returns from inside to outside.

Without twice NAT, only one of the translation rules is applied on a packet, either the source IP address and port number or the destination IP address and port number.

Static NAT translations that belong to the same group are considered for twice NAT configuration. If a static configuration does not have a configured group ID, the twice NAT configuration will not work. All inside and outside NAT translations that belong to a single group that is identified by the group ID are paired to form twice NAT translations.

Dynamic twice NAT translations dynamically select the source IP address and port number information from pre-defined **ip nat pool** or **interface overload** configurations. Packet filtration is done by configuring ACLs, and traffic must originate from the dynamic NAT translation rule direction such that source translation is done by using dynamic NAT rules.

Dynamic twice NAT allows you to configure two NAT translations (one inside and one outside) as part of a group of translations. One translation must be dynamic and other translation must be static. When these two translations are part of a group of translations, both the translations can be applied on a single packet as it goes through the NAT device either from inside to outside or from outside to inside.

VRF Aware NAT

The VRF aware NAT feature enables a switch to understand an address space in a VRF (virtual routing and forwarding instances) and to translate the packet. This allows the NAT feature to translate traffic in an overlapping address space that is used between two VRFs.

Notes for VRF aware NAT:

- The VRF aware NAT feature is supported on N9K-9408PC-CFP2, N9K-X9564PX, N9K-C9272Q, N9K-C9272Q, N9K-X9464TX, N9K-X9464TX2, N9K-X9564TX, N9K-X9464PX, N9K-X9536PQ, N9K-X9636PQ, N9K-X9432PQ, N9K-C9332PQ, N9K-C9372PX, N9K-C9372PX-E, N9K-C9372TX, N9K-C9372TX-E, N9K-C93120TX.
- The VRF aware NAT feature is not supported on the Cisco Nexus 9300-EX, 9300-FX, 9300-FX2 and 9300-GX platform switches.

Note This is a NAT TCAM limitation for the Cisco Nexus 9300-EX and 9300-FX platform switches. NAT TCAM is not VRF aware. NAT does not work with overlapping IP addresses on Cisco Nexus 9300-EX, 9300-FX, 9300-FX2 and 9300-GX platform switches.

- Traffic flowing from one non-default-vrf to another non-default-vrf is not translated. (For example, vrfA to vrfB.)
- For traffic flowing from a VRF to a global-VRF, a nat-outside configuration is not supported on a non-default VRF interface.
- VRF aware NAT is supported by static and dynamic NAT configurations.
  - When traffic is configured to flow from a non-default VRF (inside) to a default VRF (outside), the **match-in-vrf** option of the **ip nat** command cannot be specified.
  - When traffic is configured to flow from a non-default VRF (inside) to the same non-default VRF (outside), the **match-in-vrf** option of the **ip nat** command must be specified.

    The following is an example configuration:

```
Switch(config)# ip nat inside source {list <acl-name>} {pool <pool-name> [vrf
<vrf-name> [match-in-vrf]] [overload] | interface <globalAddrInterface> [vrf
<vrf-name> [match-in-vrf]] overload} [group <group-id> dynamic]


Switch(config)#ip nat outside source list <acl-name> pool <pool-name> [vrf <vrf-name>
 [match-in-vrf]] [group <group-id> dynamic]}
```

- VRF aware NAT does not support fragmented packets.
- VRF aware NAT does not support application layer translations.

  Therefore, Layer 4 and other embedded IPs are not translated and the following will fail:
  - FTP

  - ICMP failures
  - IPSec
  - HTTPS

- VRF aware NAT supports NAT or VACL on an interface. (However, both features cannot be supported at the same time on an interface.)
- VRF aware NAT supports egress ACLs that are applied to the original packet, not on the NAT translated packet.
- VRF aware NAT supports only the default VRF.
- VRF aware NAT does not provide MIB support.
- VRF aware NAT does not provide DCNM support.
- VRF aware NAT supports only a single global VDC.
- VRF aware NAT does not support the active/standby supervisor model.
- VRFs with overlapping subnets cannot go to a common destination without NAT. However, you can achieve this functionality with inter-VRF NAT on dynamic NAT rule configuration. Static NAT configuration is not supported for overlapping address.

Guidelines and Limitations for Static NAT

Static NAT has the following configuration guidelines and limitations:

- For Broadcom-based Cisco Nexus 9000 Series switches, if the route to your inside global address on the translating device is reachable via the outside interface, packets for Network Address Translated flows coming from outside to inside get software forwarded, duplicated, and looped in the network. For this situation, you must enter the **add-route** CLI argument on the end of the NAT configuration for this flow. For example, **`ip nat inside source static 192.168.1.1 172.16.1.1 add-route`**.
- The static NAT feature over vPC is supported on Cisco Nexus 9300 platform switches.
- **show** commands with the **internal** keyword are not supported.
- The static NAT feature is supported on Cisco Nexus 9300 platform switches.
- The static NAT feature is supported on Cisco Nexus 9200 platform switches.
- On Cisco Nexus 9200 and 9300-EX, -FX, -FX2, -FX3, -FXP, -GX platform switches, the **add-route** option is required for both inside and outside policies.

Note NAT is not supported on Cisco Nexus 9500 platform switches.

- NAT supports up to 1024 translations which include both static and dynamic NAT.
- If the translated IP is part of the outside interface subnet, then use the **ip proxy-arp** command on the NAT outside interface. If the **add-route** keyword is used, **ip proxy-arp** should be enabled.

- NAT and sFlow are not supported on the same port.
- The Cisco Nexus device supports NAT on the following interface types:
  - Switch Virtual Interfaces (SVIs)
  - Routed ports
  - Layer 3 and Layer 3 subinterfaces.
- NAT is supported on the default Virtual Routing and Forwarding (VRF) table only.
- NAT is supported for IPv4 Unicast only.
- The Cisco Nexus device does not support the following:
  - Software translation. All translations are done in the hardware.
  - Application layer translation. Layer 4 and other embedded IPs are not translated, including FTP, ICMP failures, IPSec, and HTTPs.
  - NAT and VLAN Access Control Lists (VACLs) that are configured on an interface at the same time.
  - PAT translation of fragmented IP packets.
  - NAT translation on software forwarded packets. For example, packets with IP-options are not NAT translated.
- By default no TCAM entries are allocated for the NAT feature. You allocate the TCAM size for the NAT feature by adjusting the TCAM size of other features. The TCAM can be allocated with the **hardware access-list tcam region nat** *tcam-size* command.
- HSRP and VRRP are not supported on a NAT interface.
- If an IP address is used for Static NAT or PAT translations, it cannot be used for any other purpose. For example, it cannot be assigned to an interface.
- For Static NAT, the outside global IP address should be different from the outside interface IP address.
- When configuring a large number of translations (more than 100), it is faster to configure the translations before configuring the NAT interfaces.
- NAT supports (non-disruptive) In Service Software Upgrade (ISSU).
- UDF-based features may not work when NAT TCAM is carved.
- ECMP NAT is not supported on Cisco Nexus 9000 switches.
- NAT configurations such as **ip nat inside** or **ip nat outside** are not supported on loopback interfaces.

Restrictions for Dynamic NAT

The following restrictions apply to dynamic Network Address Translation (NAT):

- For Broadcom-based Cisco Nexus 9000 Series switches, if the route to your inside global address on the translating device is reachable via the outside interface, packets for Network Address Translated flows

coming from outside to inside get software forwarded, duplicated, and looped in the network. For this situation, you must enter the **add-route** CLI argument on the end of the NAT configuration for this flow. For example, **`ip nat inside source static 192.168.1.1 172.16.1.1 add-route`**.

- VRF aware NAT is not supported for overlapping inside/outside IP subnet addresses on Cisco Nexus 9200 and 9300-EX platform switches.
- **show** commands with the **internal** keyword are not supported.
- The dynamic NAT feature is supported on Cisco Nexus 9300 platform switches.
- The dynamic NAT feature is supported on Cisco Nexus 9200 platform switches.
- On Cisco Nexus 9200 and 9300-EX, -FX, -FX2, -FX3, -FXP, -GX platform switches, the **add-route** option is required for both inside and outside policies.
- The **interface overload option for inside policies** option is not supported on the on the Cisco Nexus 9200, 9300-EX, 9300-FX 9300-FX2, 9300-FX3, 9300-FXP, and 9300-GX platform switches for both outside and inside policies.
- VXLAN routing is not supported on Cisco Nexus devices.
- Fragmented packets are not supported.
- Application layer gateway (ALG) translations are not supported. ALG, also known as application-level gateway, is an application that translates IP address information inside the payload of an application packet.
- Egress ACLs are not applied to translated packets.
- Nondefault virtual routing and forwarding (VRF) instances are not supported.
- MIBs are not supported.
- Cisco Data Center Network Manager (DCNM) is not supported.
- Multiple global virtual device contexts (VDCs) are not supported on Cisco Nexus devices.
- Dynamic NAT translations are not synchronized with active and standby devices.
- Stateful NAT is not supported. However, NAT and Hot Standby Router Protocol (HSRP) can coexist.
- The timeout value for take up to the configured time-out + 119 seconds.
- Normally, ICMP NAT flows time out after the expiration of the configured sampling-timeout and translation-timeout. However, when ICMP NAT flows present in the switch become idle, they time out immediately after the expiration of the sampling-timeout configured.
- Hardware programming is introduced for ICMP on Cisco Nexus 9300 platform switches. Therefore, the ICMP entries consume the TCAM resources in the hardware. Because ICMP is in the hardware, the maximum limit for NAT translation in Cisco Nexus platform Series switches is changed to 1024. Maximum of 100 ICMP entries are allowed to make the best usage of the resources.
- When creating a new translation on a Cisco Nexus 9000 Series switch, the flow is software forwarded until the translation is programmed in the hardware, which might take a few seconds. During this period, there is no translation entry for the inside global address. Therefore, returning traffic is dropped. To overcome this limitation, create a loopback interface and give it an IP address that belongs to the NAT pool.

- For dynamic NAT, pool overload and interface overload are not supported for the outside NAT.
- Because the NAT overload uses PBR (Policy-Based Routing), the maximum number of available next-hop entries in the PBR table determines NAT scale. If the number of NAT inside interfaces are within the range of available next-hops entries in the PBR table, the maximum NAT translation scale remains same. Otherwise, the maximum number of supported translations may reduce. PBR and NAT-overload are not mutually exclusive; they are mutually limiting.
- The Cisco Nexus devices does not support NAT and VLAN Access Control Lists (VACLs) that are configured on an interface at the same time.
- NAT configurations such as **ip nat inside** or **ip nat outside** are not supported on loopback interfaces.
- The dynamic NAT feature over vPC is not supported.
- If traffic ingresses a PBR enabled interface, and has a NAT entry, the traffic will be routed via PBR but the IP address will not be translated.

Guidelines and Limitations for Dynamic Twice NAT

For Broadcom-based Cisco Nexus 9000 Series switches, if the route to your inside global address on the translating device is reachable via the outside interface, packets for Network Address Translated flows coming from outside to inside get software forwarded, duplicated, and looped in the network. For this situation, you must enter the **add-route** CLI argument on the end of the NAT configuration for this flow. For example, **`ip nat inside source static 192.168.1.1 172.16.1.1 add-route`**.

IP packets without TCP/UDP/ICMP headers are not translated with dynamic NAT.

In dynamic twice NAT, if dynamic NAT flows are not created before creating static NAT flows, dynamic twice NAT flows are not created correctly.

When an empty ACL is created, the default rule of **permit ip any any** is configured. The NAT-ACL does not match further ACL entries if the first ACL is blank.

Guidelines and Limitations for TCP Aware NAT

TCP aware NAT has the following limitations:

- TCP aware NAT is supported on Cisco Nexus 9500 and Cisco Nexus 9300-EX, FX, and FX2 series switches.
- Beginning with Cisco NX-OS Release 9.3(5), TCP aware NAT is supported on Cisco Nexus N9K-C9316D-GX, N9K-C93600CD-GX, N9K-C9364C-GX switches.
- Only one match ACL can be associated with one range of addresses pool. After associating a pool to a match ACL you cannot change the interface IP or modify the pool range.
- You must define the pool before configuring or using it in a dynamic NAT configuration.
- The dynamic NAT rule must be reconfigured whenever there is a change in pool range or interface address in case of interface overload.

Configuring Static NAT

Enabling Static NAT

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **feature nat**
3. switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **feature nat**	Enables the static NAT feature on the device.
Step 3	switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Configuring Static NAT on an Interface

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface** *type slot*/*port*
3. switch(config-if)# **ip nat** {**inside** | **outside**}
4. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *type slot*/*port*	Specifies an interface to configure, and enters interface configuration mode.
Step 3	switch(config-if)# **ip nat** {**inside** | **outside**}	Specifies the interface as inside or outside. **Note** Only packets that arrive on a marked interface can be translated. This configuration is not supported on loopback interface.

	Command or Action	Purpose
Step 4	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to configure an interface with static NAT from the inside:

```
switch# configure terminal
switch(config)# interface ethernet 1/4
switch(config-if)# ip nat inside
```

Enabling Static NAT for an Inside Source Address

For inside source translation, the traffic flows from inside interface to the outside interface. NAT translates the inside local IP address to the inside global IP address. On the return traffic, the destination inside global IP address gets translated back to the inside local IP address.

Note When the Cisco Nexus device is configured to translate an inside source IP address (Src:ip1) to an outside source IP address (newSrc:ip2), the Cisco Nexus device implicitly adds a translation for an outside destination IP address (Dst: ip2) to an inside destination IP address (newDst: ip1).

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **ip nat inside source static** *local-ip-address global-ip-address* [**vrf** *vrf-name*] [**match-in-vrf**] [**group** *group-id*]
3. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **ip nat inside source static** *local-ip-address global-ip-address* [**vrf** *vrf-name*] [**match-in-vrf**] [**group** *group-id*]	Configures static NAT to translate the inside local address to the inside global address or to translate the opposite (the inside global traffic to the inside local traffic). Specifying **group** specifies the group to which this translation belongs on the static twice NAT.

	Command or Action	Purpose
		Note While performing twice NAT configuration in Cisco Nexus 9000 Series switches, you cannot use the same group ID across different VRFs. A unique group ID should be used for unique twice NAT rules.
Step 3	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to configure static NAT for an inside source address:

```
switch# configure terminal
switch(config)# ip nat inside source static 1.1.1.1 5.5.5.5
switch(config)# copy running-config startup-config
```

Enabling Static NAT for an Outside Source Address

For outside source translation, the traffic flows from the outside interface to the inside interface. NAT translates the outside global IP address to the outside local IP address. On the return traffic, the destination outside local IP address gets translated back to outside global IP address.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **ip nat outside source static** *outsideGlobalIP outsideLocalIP* [**vrf** *vrf-name* [**match-in-vrf**] [**group** *group-id*] [**dynamic**] [**add-route**]]
3. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **ip nat outside source static** *outsideGlobalIP outsideLocalIP* [**vrf** *vrf-name* [**match-in-vrf**] [**group** *group-id*] [**dynamic**] [**add-route**]]	Configures static NAT to translate the outside global address to the outside local address or to translate the opposite (the outside local traffic to the outside global traffic). Specifying **group** specifies the group to which this translation belongs on the static twice NAT. When an inside translation without ports is configured, an implicit add route is performed. The original add route functionality is an option while configuring an outside translation.
Step 3	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example show how to configure static NAT for an outside source address:

```
switch# configure terminal
switch(config)# ip nat outside source static 2.2.2.2 6.6.6.6
switch(config)# copy running-config startup-config
```

Configuring Static PAT for an Inside Source Address

You can map services to specific inside hosts using Port Address Translation (PAT).

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **ip nat inside source static** {*inside-local-address inside-global-address* | {**tcp**| **udp**} *inside-local-address* {*local-tcp-port* | *local-udp-port*} *inside-global-address* {*global-tcp-port* | *global-udp-port*}} {**vrf** *vrf-name* {**match-in-vrf**} {**group** *group-id*} }
3. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **ip nat inside source static** {*inside-local-address inside-global-address* | {**tcp**| **udp**} *inside-local-address* {*local-tcp-port* | *local-udp-port*} *inside-global-address* {*global-tcp-port* | *global-udp-port*}} {**vrf** *vrf-name* {**match-in-vrf**} {**group** *group-id*} }	Maps static NAT to an inside local port to an inside global port.
Step 3	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to map UDP services to a specific inside source address and UDP port:

```
switch# configure terminal
switch(config)#  ip nat inside source static udp 20.1.9.2 63 35.48.35.48 130
switch(config)# copy running-config startup-config
```

Configuring Static PAT for an Outside Source Address

You can map services to specific outside hosts using Port Address Translation (PAT).

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **ip nat outside source static** {*outside-global-address outside-local-address* | {**tcp** | **udp**} *outside-global-address* {*global-tcp-port* | *global-udp-port*} *outside-local-address* {*global-tcp-port* | *global-udp-port*}} {**group** *group-id*} {**add-route**} {**vrf** *vrf-name* {**match-in-vrf**}}
3. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **ip nat outside source static** {*outside-global-address outside-local-address* | {**tcp** | **udp**} *outside-global-address* {*global-tcp-port* | *global-udp-port*} *outside-local-address* {*global-tcp-port* | *global-udp-port*}} {**group** *group-id*} {**add-route**} {**vrf** *vrf-name* {**match-in-vrf**}}	Maps static NAT to an outside global port to an outside local port. Specifying **group** specifies the group to which this translation belongs on the static twice NAT. When an inside translation without ports is configured, an implicit add route is performed. The original add route functionality is an option while configuration an outside translation.
Step 3	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to map TCP services to a specific outside source address and TCP port:

```
switch# configure terminal
switch(config)#  ip nat outside source static tcp 20.1.9.2 63 35.48.35.48 130
switch(config)# copy running-config startup-config
```

Configuring Static Twice NAT

All translations within the same group are considered for creating static twice Network Address Translation (NAT) rules.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip nat inside source static** *inside-local-ip-address inside-global-ip-address* [**group** *group-id*] [**add-route**]
4. **ip nat outside source static** *outside-global-ip-address outside-local-ip-address* [**group** *group-id*] [**add-route**]
5. **interface** *type number*
6. **ip address** *ip-address mask*
7. **ip nat inside**
8. **exit**

9. **interface** *type number*
10. **ip address** *ip-address mask*
11. **ip nat outside**
12. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `switch> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `switch# configure terminal`	Enters privileged EXEC mode.
Step 3	**ip nat inside source static** *inside-local-ip-address inside-global-ip-address* [**group** *group-id*] [**add-route**] **Example:** `switch(config)# ip nat inside source static 10.1.1.1 192.168.34.4 group 4`	Configures static twice NAT to translate an inside local IP address to the corresponding inside global IP address. • The **group** keyword determines the group to which a translation belongs.
Step 4	**ip nat outside source static** *outside-global-ip-address outside-local-ip-address* [**group** *group-id*] [**add-route**] **Example:** `switch(config)# ip nat outside source static 209.165.201.1 10.3.2.42 group 4 add-route`	Configures static twice NAT to translate an outside global IP address to the corresponding outside local IP address. • The **group** keyword determines the group to which a translation belongs.
Step 5	**interface** *type number* **Example:** `switch(config)# interface ethernet 1/2`	Configures an interface and enters interface configuration mode.
Step 6	**ip address** *ip-address mask* **Example:** `switch(config-if)# ip address 10.2.4.1 255.255.255.0`	Sets a primary IP address for an interface.
Step 7	**ip nat inside** **Example:** `switch(config-if)# ip nat inside`	Connects the interface to an inside network, which is subject to NAT. **Note** Configuration not supported on loopback interface.
Step 8	**exit** **Example:** `switch(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.

	Command or Action	Purpose
Step 9	**interface** *type number* **Example:** `switch(config)# interface ethernet 1/1`	Configures an interface and enters interface configuration mode.
Step 10	**ip address** *ip-address mask* **Example:** `switch(config-if)# ip address 10.5.7.9 255.255.255.0`	Sets a primary IP address for an interface.
Step 11	**ip nat outside** **Example:** `switch(config-if)# ip nat outside`	Connects the interface to an outside network, which is subject to NAT. **Note** Configuration not supported on loopback interface.
Step 12	**end** **Example:** `switch(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Enabling and Disabling no-alias Configuration

NAT devices own Inside Global (IG) and Outside Local (OL) addresses and they are responsible for responding to any ARP requests directed to these addresses. When the IG/OL address subnet matches with the local interface subnet, NAT installs an IP alias and an ARP entry, in this case the device uses local-proxy-arp to respond to ARP requests.

The *no-alias* feature responds to ARP requests of all the translated IPs from a given NAT pool address range if the address range is in same subnet of the outside interface.

If no-alias is enabled on an interface with NAT configuration, the outside interface will not respond to any ARP requests in its subnet. When no-alias is disabled, the ARP requests for IPs in same subnet as of outside interface are served.

Note When you downgrade to any older releases that does not support this feature, configurations with *no-alias* option may be deleted.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **feature nat**
3. switch(config)# **show run nat**
4. switch(config)# **show ip nat-alias**
5. switch(config)# **clear ip nat-alias** *ip address*/*all*

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **feature nat**	Enables the static NAT feature on the device.
Step 3	switch(config)# **show run nat**	Displays NAT configuration.
Step 4	switch(config)# **show ip nat-alias**	Displays the information whether or not the alias is created. **Note** By default, alias is created. To disable the alias, you must append *no-alias* keyword to the command.
Step 5	switch(config)# **clear ip nat-alias** *ip address*/*all*	Removes entries from alias list. To remove a specific entry you must provide the IP address that you want to remove. To remove all entries, use the all keyword.

Example

This example shows the interface information:

```
switch# configure terminal
switch(config)# show ip int b
IP Interface Status for VRF "default"(1)
Interface            IP Address      Interface Status
Lo0                  100.1.1.1       protocol-up/link-up/admin-up
Eth1/1               7.7.7.1         protocol-up/link-up/admin-up
Eth1/3               8.8.8.1         protocol-up/link-up/admin-up
```

This example shows the running configuration:

```
switch# configure terminal
switch(config)# show running-config nat
!Command: show running-config nat
!Running configuration last done at: Thu Aug 23 11:57:01 2018
!Time: Thu Aug 23 11:58:13 2018

version 9.2(2) Bios:version 07.64
feature nat
interface Ethernet1/1
  ip nat inside
interface Ethernet1/3
  ip nat outside
switch(config)#
```

This example shows how to configure alias:

```
switch# configure terminal
switch(config)# ip nat pool p1 7.7.7.2 7.7.7.20 prefix-length 24
switch(config)# ip nat inside source static 1.1.1.2 8.8.8.3
switch(config)# ip nat outside source static 2.2.2.1 7.7.7.3
switch(config)# show ip nat-alias
Alias Information for Context: default
Address         Interface
7.7.7.2          Ethernet1/1
8.8.8.2          Ethernet1/3
switch(config)#
```

This example shows the output of *show ip nat-alias*. By default, alias is enabled.

```
switch# configure terminal
switch(config)# show ip nat-alias
Alias Information for Context: default
Address         Interface
7.7.7.2          Ethernet1/1
8.8.8.2          Ethernet1/3
switch(config)#
```

This example shows how to disable alias:

```
switch# configure terminal
switch(config)# ip nat pool p1 7.7.7.2 7.7.7.20 prefix-length 24 no-alias
switch(config)# ip nat inside source static 1.1.1.2 8.8.8.3 no-alias
switch(config)# ip nat outside source static 2.2.2.1 7.7.7.3 no-alias
switch(config)# show ip nat-alias
Alias Information for Context: default
Address         Interface
7.7.7.2          Ethernet1/1
8.8.8.2          Ethernet1/3
switch(config)#

** None of the entry got appended as alias is disabled for above CLIs.
switch(config)#
```

This example shows how to clear alias. Use *clear ip nat-alias* to remove an entry from alias list. You can remove a single entry by specifying the IP address or remove all the alias entries.

```
switch# configure terminal
switch(config)# clear ip nat-alias address 7.7.7.2
switch(config)# show ip nat-alias
Alias Information for Context: default
Address         Interface
8.8.8.2          Ethernet1/3
switch(config)#
switch(config)# clear ip nat-alias all
switch(config)# show ip nat-alias
switch(config)#
```

Configuration Example for Static NAT and PAT

This example shows the configuration for static NAT:

```
ip nat inside source static 103.1.1.1 11.3.1.1
ip nat inside source static 139.1.1.1 11.39.1.1
ip nat inside source static 141.1.1.1 11.41.1.1
ip nat inside source static 149.1.1.1 95.1.1.1
ip nat inside source static 149.2.1.1 96.1.1.1
ip nat outside source static 95.3.1.1 95.4.1.1
ip nat outside source static 96.3.1.1 96.4.1.1
ip nat outside source static 102.1.2.1 51.1.2.1
ip nat outside source static 104.1.1.1 51.3.1.1
ip nat outside source static 140.1.1.1 51.40.1.1
```

This example shows the configuration for static PAT:

```
ip nat inside source static tcp 10.11.1.1 1 210.11.1.1 101
ip nat inside source static tcp 10.11.1.1 2 210.11.1.1 201
ip nat inside source static tcp 10.11.1.1 3 210.11.1.1 301
ip nat inside source static tcp 10.11.1.1 4 210.11.1.1 401
```

```
ip nat inside source static tcp 10.11.1.1 5 210.11.1.1 501
ip nat inside source static tcp 10.11.1.1 6 210.11.1.1 601
ip nat inside source static tcp 10.11.1.1 7 210.11.1.1 701
ip nat inside source static tcp 10.11.1.1 8 210.11.1.1 801
ip nat inside source static tcp 10.11.1.1 9 210.11.1.1 901
ip nat inside source static tcp 10.11.1.1 10 210.11.1.1 1001
ip nat inside source static tcp 10.11.1.1 11 210.11.1.1 1101
ip nat inside source static tcp 10.11.1.1 12 210.11.1.1 1201
```

Example: Configuring Static Twice NAT

The following example shows how to configure the inside source and outside source static twice NAT configurations:

```
Switch> enable
Switch# configure terminal
Switch(config)# ip nat inside source static 10.1.1.1 192.168.34.4 group 4
Switch(config)# ip nat outside source static 209.165.201.1 10.3.2.42 group 4
Switch(config)# interface ethernet 1/2
Switch(config-if)# ip address 10.2.4.1 255.255.255.0
Switch(config-if)# ip nat inside
switch(config-if)# exit
switch(config)# interface ethernet 1/1
switch(config-if)# ip address 10.5.7.9 255.255.255.0
switch(config-if)# ip nat outside
Switch(config-if)# end
```

Verifying the Static NAT Configuration

To display the static NAT configuration, perform this task:

SUMMARY STEPS

1. switch# **show ip nat translations**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **show ip nat translations**	Shows the translations for the inside global, inside local, outside local, and outside global IP addresses.

Example

This example shows how to display the static NAT configuration:

```
switch# sh ip nat translations
Pro Inside global      Inside local       Outside local      Outside global

--- ---                ---                51.3.1.1           104.1.1.1
--- ---                ---                95.4.1.1           95.3.1.1
--- ---                ---                96.4.1.1           96.3.1.1
--- ---                ---                51.40.1.1          140.1.1.1
--- ---                ---                51.42.1.1          142.1.2.1
```

```
--- ---                 ---                 51.1.2.1          102.1.2.1
--- 11.1.1.1            101.1.1.1           ---               ---
--- 11.3.1.1            103.1.1.1           ---               ---
--- 11.39.1.1           139.1.1.1           ---               ---
--- 11.41.1.1           141.1.1.1           ---               ---
--- 95.1.1.1            149.1.1.1           ---               ---
--- 96.1.1.1            149.2.1.1           ---               ---
    130.1.1.1:590       30.1.1.100:5000     ---               ---
    130.2.1.1:590       30.2.1.100:5000     ---               ---
    130.3.1.1:590       30.3.1.100:5000     ---               ---
    130.4.1.1:590       30.4.1.100:5000     ---               ---
    130.1.1.1:591       30.1.1.101:5000     ---               ---
```

```
switch# sh ip nat translations verbose
Pro Inside global      Inside local       Outside local      Outside global
any ---                ---                22.1.1.3           22.1.1.2
    Flags:0x200009 time-left(secs):-1 id:0 state:0x0 grp_id:10
any 11.1.1.130         11.1.1.3           ---                ---
    Flags:0x1 time-left(secs):-1 id:0 state:0x0 grp_id:0
any 11.1.1.133         11.1.1.33          ---                ---
    Flags:0x1 time-left(secs):-1 id:0 state:0x0 grp_id:10
any 11.1.1.133         11.1.1.33          22.1.1.3           22.1.1.2
    Flags:0x200009 time-left(secs):-1 id:0 state:0x0 grp_id:0
tcp 10.1.1.100:64490   10.1.1.2:0         20.1.1.2:0         20.1.1.2:0
    Flags:0x82 time-left(secs):43192 id:31 state:0x3 grp_id:0 vrf: default
N9300-1#
```

Configuring Dynamic NAT

Configuring Dynamic Translation and Translation Timeouts

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip access-list** *access-list-name*
4. **permit** *protocol source source-wildcard* **any**
5. **deny** *protocol source source-wildcard* **any**
6. **exit**
7. **ip nat inside source list** *access-list-name* **interface** *type number* [**vrf** *vrf-name* [**match-in-vrf**] [**add-route**] [**overload**]
8. **interface** *type number*
9. **ip address** *ip-address mask*
10. **ip nat inside**
11. **exit**
12. **interface** *type number*
13. **ip address** *ip-address mask*
14. **ip nat outside**
15. **exit**

16. **ip nat translation max-entries** *number-of-entries*
17. **ip nat translation timeout** *seconds*
18. **ip nat translation creation-delay** *seconds*
19. **ip nat translation icmp-timeout** *seconds*
20. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Switch> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Switch# configure terminal`	Enters global configuration mode.
Step 3	**ip access-list** *access-list-name* **Example:** `Switch(config)# ip access-list acl1`	Defines an access list and enters access-list configuration mode.
Step 4	**permit** *protocol source source-wildcard* **any** **Example:** `Switch(config-acl)# permit ip 10.111.11.0/24 any`	Sets conditions in an IP access list that permit traffic matching the conditions.
Step 5	**deny** *protocol source source-wildcard* **any** **Example:** `Switch(config-acl)# deny udp 10.111.11.100/32 any`	Sets conditions in an IP access list that deny packets from entering a network.
Step 6	**exit** **Example:** `Switch(config-acl)# exit`	Exits access-list configuration mode and returns to global configuration mode.
Step 7	**ip nat inside source list** *access-list-name* **interface** *type number* [**vrf** *vrf-name* [**match-in-vrf**] [**add-route**] [**overload**] **Example:** `Switch(config)# ip nat inside source list acl1 interface ethernet 1/1 overload`	Establishes dynamic source translation by specifying the access list defined in Step 3.
Step 8	**interface** *type number* **Example:** `Switch(config)# interface ethernet 1/4`	Configures an interface and enters interface configuration mode.
Step 9	**ip address** *ip-address mask* **Example:**	Sets a primary IP address for the interface.

	Command or Action	Purpose
	`Switch(config-if)# ip address 10.111.11.39 255.255.255.0`	
Step 10	**ip nat inside** **Example:** `Switch(config-if)# ip nat inside`	Connects the interface to an inside network, which is subject to NAT. **Note** Configuration not supported on loopback interface.
Step 11	**exit** **Example:** `Switch(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 12	**interface** *type number* **Example:** `Switch(config)# interface ethernet 1/1`	Configures an interface and enters interface configuration mode.
Step 13	**ip address** *ip-address mask* **Example:** `Switch(config-if)# ip address 172.16.232.182 255.255.255.240`	Sets a primary IP address for an interface.
Step 14	**ip nat outside** **Example:** `Switch(config-if)# ip nat outside`	Connects the interface to an outside network. **Note** Configuration not supported on loopback interface.
Step 15	**exit** **Example:** `Switch(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 16	**ip nat translation max-entries** *number-of-entries* **Example:** `Switch(config)# ip nat translation max-entries 300`	Specifies the maximum number of dynamic NAT translations. The number of entries can be between 1 and 1023.
Step 17	**ip nat translation timeout** *seconds* **Example:** `switch(config)# ip nat translation timeout 13000`	Specifies the timeout value for dynamic NAT translations.
Step 18	**ip nat translation creation-delay** *seconds* **Example:**	Specifies the ICMP timeout value for dynamic NAT translations.

	Command or Action	Purpose
	`switch(config)# ip nat translation creation-delay 250`	**Note** To reduce the frequency of programming the NAT entries in the hardware, NAT batches and programs the translations for one second. Frequently programming the hardware burdens the CPU but delaying the programming delays establishing sessions. You can disable batching or reduce the creation delay using this command. It is not recommended to set creation delay to 0.
Step 19	**ip nat translation icmp-timeout** *seconds* **Example:** `switch(config)# ip nat translation icmp-timeout 100`	Specifies the ICMP timeout value for dynamic NAT translations.
Step 20	**end** **Example:** `Switch(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Configuring Dynamic NAT Pool

You can create a NAT pool by either defining the range of IP addresses in a single **ip nat pool** command or by using the **ip nat pool** and **address** commands

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **feature nat**
3. switch(config)# **ip nat pool** *pool-name* [*startip endip*] {**prefix** *prefix-length* | **netmask** *network-mask*}
4. (Optional) switch(config-ipnat-pool)# **address** *startip endip*
5. (Optional) switch(config)# **no ip nat pool** *pool-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **feature nat**	Enables the NAT feature on the device.
Step 3	switch(config)# **ip nat pool** *pool-name* [*startip endip*] {**prefix** *prefix-length* | **netmask** *network-mask*}	Creates a NAT pool with a range of global IP addresses. The IP addresses are filtered by using either a prefix length or a network mask.
Step 4	(Optional) switch(config-ipnat-pool)# **address** *startip endip*	Specifies the range of global IP addresses if they were not specified during creation of the pool.
Step 5	(Optional) switch(config)# **no ip nat pool** *pool-name*	Deletes the specified NAT pool.

Example

This example shows how to create a NAT pool with a prefix length:

```
switch# configure terminal
switch(config)# ip nat pool pool1 30.1.1.1 30.1.1.2 prefix-length 24
switch(config)#
```

This example shows how to create a NAT pool with a network mask:

```
switch# configure terminal
switch(config)# ip nat pool pool5 20.1.1.1 20.1.1.5 netmask 255.0.255.0
switch(config)#
```

This example shows how to create a NAT pool and define the range of global IP addresses using the **ip nat pool** and **address** commands:

```
switch# configure terminal
switch(config)# ip nat pool pool7 netmask 255.255.0.0
switch(config-ipnat-pool)# address 40.1.1.1 40.1.1.5
switch(config-ipnat-pool)#
```

This example shows how to delete a NAT pool:

```
switch# configure terminal
switch(config)# no ip nat pool pool4
switch(config)#
```

Configuring Source Lists

You can configure a source list of IP addresses for the inside interface and the outside interface.

Before you begin

Ensure that you configure a pool before configuring the source list for the pool.

SUMMARY STEPS

1. switch# **configure terminal**
2. (Optional) switch# **ip nat inside source list** *list-name* **pool** *pool-name* [**overload**]
3. (Optional) switch# **ip nat outside source list** *list-name* **pool** *pool-name* [**add-route**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	(Optional) switch# **ip nat inside source list** *list-name* **pool** *pool-name* [**overload**]	Creates a NAT inside source list with pool with or without overloading.

	Command or Action	Purpose
Step 3	(Optional) switch# **ip nat outside source list** *list-name* **pool** *pool-name* [**add-route**]	Creates a NAT outside source list with pool without overloading.

Example

This example shows how to create a NAT inside source list with pool without overloading:

```
switch# configure terminal
switch(config)# ip nat inside source list list1 pool pool1
switch(config)#
```

This example shows how to create a NAT inside source list with pool with overloading:

```
switch# configure terminal
switch(config)# ip nat inside source list list2 pool pool2 overload
switch(config)#
```

This example shows how to create a NAT outside source list with pool without overloading:

```
switch# configure terminal
switch(config)# ip nat outside source list list3 pool pool3
switch(config)#
```

Configuring Dynamic Twice NAT for an Inside Source Address

For an inside source address translation, the traffic flows from the inside interface to the outside interface. You can configure dynamic twice NAT for an inside source address.

Before you begin

Ensure that you enable NAT on the switch.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **ip nat outside source static** *outside-global-ip-address outside-local-ip-address* | [**tcp** | **udp**] *outside-global-ip-address outside-global-port outside-local-ip-address outside-local-port* [**group** *group-id*] [**dynamic**] [**add-route**]
3. switch(config)# **ip nat inside source list** *access-list-name* [**interface** *type slot/port* **overload** | **pool** *pool-name* **overload**] [**group** *group-id*] [**dynamic**] [**add-route**]
4. switch(config)# **ip nat pool** *pool-name* [*startip endip*] {**prefix** *prefix-length* | **netmask** *network-mask*}
5. switch(config)# **interface** *type slot/port*
6. switch(config-if)# **ip nat outside**
7. switch(config-if)# **exit**
8. switch(config)# **interface** *type slot/port*
9. switch(config-if)# **ip nat inside**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **ip nat outside source static** *outside-global-ip-address outside-local-ip-address* | [**tcp** | **udp**] *outside-global-ip-address outside-global-port outside-local-ip-address outside-local-port* [**group** *group-id*] [**dynamic**] [**add-route**]	Configures static NAT to translate an outside global address to an inside local address or to translate inside local traffic to inside global traffic. The **group** keyword determines the group to which a translation belongs.
Step 3	switch(config)# **ip nat inside source list** *access-list-name* [**interface** *type slot/port* **overload** | **pool** *pool-name* **overload**] [**group** *group-id*] [**dynamic**] [**add-route**]	Establishes dynamic source translation by creating a NAT inside source list with pool with or without overloading. The **group** keyword determines the group to which a translation belongs.
Step 4	switch(config)# **ip nat pool** *pool-name* [*startip endip*] {**prefix** *prefix-length* | **netmask** *network-mask*}	Creates a NAT pool with a range of global IP addresses. The IP addresses are filtered by using either a prefix length or a network mask.
Step 5	switch(config)# **interface** *type slot/port*	Configures an interface and enters interface configuration mode.
Step 6	switch(config-if)# **ip nat outside**	Connects the interface to an outside network.
Step 7	switch(config-if)# **exit**	Exits interface configuration mode and returns to global configuration mode.
Step 8	switch(config)# **interface** *type slot/port*	Configures an interface and enters interface configuration mode.
Step 9	switch(config-if)# **ip nat inside**	Connects the interface to an inside network, which is subject to NAT.

Example

This example shows how to configure dynamic twice NAT for an inside source address:

```
switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
switch(config)# ip nat outside source static 2.2.2.2 4.4.4.4 group 20 dynamic
switch(config)# ip nat inside source list acl_1 pool pool_1 overload group 20 dynamic
switch(config)# ip nat pool pool_1 3.3.3.3 3.3.3.10 prefix-length 24
switch(config)# interface Ethernet1/8
switch(config-if)# ip nat outside
switch(config-if)# exit
switch(config)# interface Ethernet1/15
switch(config-if)# ip nat inside
```

Configuring Dynamic Twice NAT for an Outside Source Address

For an outside source address translation, the traffic flows from the outside interface to the inside interface. You can configure dynamic twice NAT for an outside source address.

Before you begin

Ensure that you enable NAT on the switch.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **ip nat inside source static** *inside-local-ip-address inside-global-ip-address* | [**tcp** | **udp**] *inside-local-ip-address local-port inside-global-ip-address global-port* [**group** *group-id*] [**dynamic**] [**add-route**]
3. switch(config)# **ip nat outside source list** *access-list-name* **pool** *pool-name* [**group** *group-id*] **dynamic** [**add-route**]
4. switch(config)# **ip nat pool** *pool-name* [*startip endip*] {**prefix** *prefix-length* | **netmask** *network-mask*}
5. switch(config)# **interface** *type slot/port*
6. switch(config-if)# **ip nat outside**
7. switch(config-if)# **exit**
8. switch(config)# **interface** *type slot/port*
9. switch(config-if)# **ip nat inside**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **ip nat inside source static** *inside-local-ip-address inside-global-ip-address* | [**tcp** | **udp**] *inside-local-ip-address local-port inside-global-ip-address global-port* [**group** *group-id*] [**dynamic**] [**add-route**]	Configures static NAT to translate an inside global address to an inside local address or to translate inside local traffic to inside global traffic. The **group** keyword determines the group to which a translation belongs.
Step 3	switch(config)# **ip nat outside source list** *access-list-name* **pool** *pool-name* [**group** *group-id*] **dynamic** [**add-route**]	Establishes dynamic source translation by creating a NAT outside source list with pool with or without overloading.
Step 4	switch(config)# **ip nat pool** *pool-name* [*startip endip*] {**prefix** *prefix-length* | **netmask** *network-mask*}	Creates a NAT pool with a range of global IP addresses. The IP addresses are filtered by using either a prefix length or a network mask.
Step 5	switch(config)# **interface** *type slot/port*	Configures an interface and enters interface configuration mode.
Step 6	switch(config-if)# **ip nat outside**	Connects the interface to an outside network.
Step 7	switch(config-if)# **exit**	Exits interface configuration mode and returns to global configuration mode.

	Command or Action	Purpose
Step 8	switch(config)# **interface** *type slot/port*	Configures an interface and enters interface configuration mode.
Step 9	switch(config-if)# **ip nat inside**	Connects the interface to an inside network, which is subject to NAT.

Example

This example shows how to configure dynamic twice NAT for an outside source address:

```
switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
switch(config)# ip nat inside source static 7.7.7.7 5.5.5.5 group 30 dynamic
switch(config)# ip nat outside source list acl_1 pool pool_1 group 30 dynamic
switch(config)# ip nat pool pool_2 4.4.4.4 4.4.4.10 prefix-length 24
switch(config)# interface Ethernet1/6
switch(config-if)# ip nat outside
switch(config-if)# exit
switch(config)# interface Ethernet1/11
switch(config-if)# ip nat inside
```

Configuring FINRST and SYN Timers

This section describes how to configure FINRST and SYN timer values. When you reload the switch, restoring or erasing the configured FINRST and/or SYN timer values depends on whether or not the TCP TCAM carved. If the TCAM is carved, the switch restores the currently configured values. If the timer values are not configured, it sets a default value of 60. If the TCAM is not carved, the switch removes any currently configured values and sets a default value as never. This is because the the TCP AWARE feature gets disabled when the TCP TCAM is not carved.

Before you begin

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config-if)# **ip nat translation syn-timeout** {**seconds** | **never**}
3. switch(config-if)# **ip nat translation finrst-timeout** {**seconds** | **never**}

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config-if)# **ip nat translation syn-timeout** {**seconds** | **never**}	Specifies the timeout value for TCP data packets that sends the SYN request, but do not receive a SYN-ACK reply. The timeout value ranges from 1 to 172800 seconds. When the TCP TCAM is carved the default value is 60 seconds. When the TCP TCAM is not carved the default value is *never*. The *never* keyword deactivates SYN timer.

	Command or Action	Purpose
		Note You cannot configure SYN timer when TCP TCAM is not carved..
Step 3	switch(config-if)# **ip nat translation finrst-timeout** {**seconds** | **never**}	Specifies the timeout value for the flow entries when a connection is terminated by receiving finish (FIN) or reset (RST) packets. You must use the configure the behavior for both RST and FIN. The timeout value ranges from 1 to 172800 seconds. When the TCP TCAM is carved the default value is 60 seconds. When the TCP TCAM is not carved the default value is *never*. The *never* keyword deactivates FIN or RST timers. **Note** You cannot configure FINRST timer when TCP TCAM is not carved..

Example

The following example that shows when TCP TCAM is carved

```
switch(config)# ip nat translation syn-timeout 20
```

The following example that shows when TCP TCAM is not carved

```
switch(config)# ip nat translation syn-timeout 20
Error: SYN TIMER CONFIG FAILED.TCP TCAM NOT CONFIGURED
```

Clearing Dynamic NAT Translations

To clear dynamic translations, perform the following task:

Command	Purpose
clear ip nat translation [**all** | **inside** *global-ip-address local-ip-address* [**outside** *local-ip-address global-ip-address*] | **outside** *local-ip-address global-ip-address*]	Deletes all or specific dynamic NAT translations.

Example

This example shows how to clear all dynamic translations:

```
switch# clear ip nat translation all
```

This example shows how to clear dynamic translations for inside and outside addresses:

```
switch# clear ip nat translation inside 2.2.2.2 4.4.4.4 outside 5.5.5.5 7.7.7.7
```

Verifying Dynamic NAT Configuration

To display dynamic NAT configuration, perform the following tasks:

Command	Purpose
show ip nat translations	Displays active Network Address Translation (NAT) translations. Displays additional information for each translation table entry, including when an entry was created and used.
show run nat	Displays NAT configuration.
show ip nat max	Displays active Network Address Translation (NAT) maximum values.
show ip nat statistics	Monitor NAT statistics.

Example

This example shows how to display IP NAT Max values:

```
switch# show ip nat max

IP NAT Max values
====================
Max Dyn Translations:80
Max all-host:0
No.Static:0
No.Dyn:1
No.Dyn-ICMP:1
====================
Switch(config)#
```

This example shows how to display NAT Statistics:

```
switch# show ip nat statistics

IP NAT Statistics
==================================================
Stats Collected since: Mon Feb 24 18:27:34 2020
--------------------------------------------------
Total active translations: 1
No.Static: 0
No.Dyn:    1
No.Dyn-ICMP:   1
--------------------------------------------------
Total expired Translations: 0
SYN timer expired:          0
FIN-RST timer expired:      0
Inactive timer expired:     0
--------------------------------------------------
Total Hits:  2                  Total Misses:  2
In-Out Hits: 0                  In-Out Misses: 2
Out-In Hits: 2                  Out-In Misses: 0
--------------------------------------------------
Total SW Translated Packets: 2
```

```
In-Out SW Translated: 2
Out-In SW Translated: 0
--------------------------------------------------
Total SW Dropped Packets: 0
In-Out SW Dropped: 0
Out-In SW Dropped: 0

Address alloc. failure drop:      0
Port alloc.  failure drop:        0
Dyn. Translation max limit drop:  0
ICMP max limit drop:              0
Allhost max limit drop:           0
--------------------------------------------------
Total TCP session established: 0
Total TCP session closed:      0
--------------------------------------------------
NAT Inside Interfaces:  1
Ethernet1/34

NAT Outside Interfaces: 1
Ethernet1/32
--------------------------------------------------
Inside source list:
++++++++++++++++++++

Access list: T2
RefCount: 1
Pool: T2     Overload
Total addresses: 10
Allocated: 1    percentage: 10%
Missed: 0


Outside source list:
++++++++++++++++++++
--------------------------------------------------
==================================================
Switch(config)#
Switch(config)#


**No.Dyn-ICMP  field is to display the no of icmp dynamic translations , its a subset of
"No.Dyn" field.
```

Note Beginning with Cisco NX-OS Release 9.3(5), the **No.Dyn-ICMP** field is a subset of **No.Dyn** field and it displays the number of ICMP dynamic translations.

This example shows how to display running configuration for NAT:

```
switch# show run nat

!Command: show running-config nat
!Time: Wed Apr 23 11:17:43 2014

version 6.0(2)A3(1)
feature nat

ip nat inside source list list1 pool pool1
ip nat inside source list list2 pool pool2 overload
ip nat inside source list list7 pool pool7 overload
```

```
ip nat outside source list list3 pool pool3
ip nat pool pool1 30.1.1.1 30.1.1.2 prefix-length 24
ip nat pool pool2 10.1.1.1 10.1.1.2 netmask 255.0.255.0
ip nat pool pool3 30.1.1.1 30.1.1.8 prefix-length 24
ip nat pool pool5 20.1.1.1 20.1.1.5 netmask 255.0.255.0
ip nat pool pool7 netmask 255.255.0.0
  address 40.1.1.1 40.1.1.5
```

This example shows how to display active NAT translations:

Inside pool with overload

```
switch# show ip nat translation
Pro  Inside global      Inside local       Outside local      Outside global
icmp 20.1.1.3:64762     10.1.1.2:133       20.1.1.1:0         20.1.1.1:0
icmp 20.1.1.3:64763     10.1.1.2:134       20.1.1.1:0         20.1.1.1:0
```

Outside pool without overload

```
switch# show ip nat translation
Pro  Inside global      Inside local       Outside local      Outside global
any  ---                ---                177.7.1.1:0        77.7.1.64:0
any  ---                ---                40.146.1.1:0       40.46.1.64:0
any  ---                ---                10.4.146.1:0       10.4.46.64:0
```

Example: Configuring Dynamic Translation and Translation Timeouts

The following example shows how to configure dynamic overload Network Address Translation (NAT) by specifying an access list:

```
Switch> enable
Switch# configure terminal
Switch(config)# ip access-list acl1
Switch(config-acl)# permit ip 10.111.11.0/24 any
Switch(config-acl)# deny udp 10.111.11.100/32 any
Switch(config-acl)# exit
Switch(config)# ip nat inside source list acl1 interface ethernet 1/1 overload
Switch(config)# interface ethernet 1/4
Switch(config-if)# ip address 10.111.11.39 255.255.255.0
Switch(config-if)# ip nat inside
Switch(config-if)# exit
Switch(config)# interface ethernet 1/1
Switch(config-if)# ip address 172.16.232.182 255.255.255.240
Switch(config-if)# ip nat outside
Switch(config-if)# exit
Switch(config)# ip nat translation max-entries 300
Switch(config)# ip nat translation timeout 13000
Switch(config)# end
```