



Configuring Advanced BGP

This chapter describes how to configure advanced features of the Border Gateway Protocol (BGP) on Cisco NX-OS switches.

This chapter includes the following sections:

- [Information About Advanced BGP, on page 1](#)
- [Prerequisites for BGP, on page 8](#)
- [Guidelines and Limitations for Advanced BGP, on page 9](#)
- [Default Settings for BGP, on page 9](#)
- [Configuring Advanced BGP, on page 10](#)
- [Configuring an Autonomous System Path Containing Your Own Autonomous System, on page 25](#)
- [Verifying the Advanced BGP Configuration, on page 40](#)
- [Displaying BGP Statistics, on page 41](#)
- [Related Topics, on page 41](#)
- [Additional References, on page 42](#)

Information About Advanced BGP

BGP is an interdomain routing protocol that provides loop-free routing between organizations or autonomous systems. Cisco NX-OS supports BGP version 4. BGP version 4 includes multiprotocol extensions that allow BGP to carry routing information for IP routes and multiple Layer 3 protocol address families. BGP uses TCP as a reliable transport protocol to create TCP sessions with other BGP-enabled switches called BGP peers. When connecting to an external organization, the router creates external BGP (eBGP) peering sessions. BGP peers within the same organization exchange routing information through internal BGP (iBGP) peering sessions.

Peer Templates

BGP peer templates allow you to create blocks of common configurations that you can reuse across similar BGP peers. Each block allows you to define a set of attributes that a peer then inherits. You can choose to override some of the inherited attributes as well, making it a very flexible scheme for simplifying the repetitive nature of BGP configurations.

Cisco NX-OS implements three types of peer templates:

- The **peer-session** template defines BGP peer session attributes, such as the transport details, remote autonomous system number of the peer, and session timers. A peer-session template can also inherit attributes from another peer-session template (with locally defined attributes that override the attributes from an inherited peer-session).
- A **peer-policy** template defines the address-family dependent policy aspects for a peer including the inbound and outbound policy, filter-lists, and prefix-lists. A peer-policy template can inherit from a set of peer-policy templates. Cisco NX-OS evaluates these peer-policy templates in the order specified by the preference value in the inherit configuration. The lowest number is preferred over higher numbers.
- The **peer** template can inherit the peer-session and peer-policy templates to allow for simplified peer definitions. It is not mandatory to use a peer template but it can simplify the BGP configuration by providing reusable blocks of configuration.

Authentication

You can configure authentication for a BGP neighbor session. This authentication method adds an MD5 authentication digest to each TCP segment sent to the neighbor to protect BGP against unauthorized messages and TCP security attacks.



Note The MD5 password must be identical between BGP peers.

Route Policies and Resetting BGP Sessions

You can associate a route policy to a BGP peer. Route policies use route maps to control or modify the routes that BGP recognizes. You can configure a route policy for inbound or outbound route updates. The route policies can match on different criteria, such as a prefix or AS_path attribute, and selectively accept or deny the routes. Route policies can also modify the path attributes.

When you change a route policy applied to a BGP peer, you must reset the BGP sessions for that peer. Cisco NX-OS supports the following three mechanisms to reset BGP peering sessions:

- **Hard reset**—A hard reset tears down the specified peering sessions, including the TCP connection, and deletes routes coming from the specified peer. This option interrupts packet flow through the BGP network. Hard reset is disabled by default.
- **Soft reconfiguration inbound**—A soft reconfiguration inbound triggers routing updates for the specified peer without resetting the session. You can use this option if you change an inbound route policy. Soft reconfiguration inbound saves a copy of all routes received from the peer before processing the routes through the inbound route policy. If you change the inbound route policy, Cisco NX-OS passes these stored routes through the modified inbound route policy to update the route table without tearing down existing peering sessions. Soft reconfiguration inbound can use significant memory resources to store the unfiltered BGP routes. Soft reconfiguration inbound is disabled by default.
- **Route Refresh**—A route refresh updates the inbound routing tables dynamically by sending route refresh requests to supporting peers when you change an inbound route policy. The remote BGP peer responds with a new copy of its routes that the local BGP speaker processes with the modified route policy. Cisco NX-OS automatically sends an outbound route refresh of prefixes to the peer.

- BGP peers advertise the route refresh capability as part of the BGP capability negotiation when establishing the BGP peer session. Route refresh is the preferred option and enabled by default.



Note BGP also uses route maps for route redistribution, route aggregation, route dampening, and other features.

eBGP

External BGP (eBGP) allows you to connect BGP peers from different autonomous systems to exchange routing updates. Connecting to external networks enables traffic from your network to be forwarded to other networks and across the Internet.

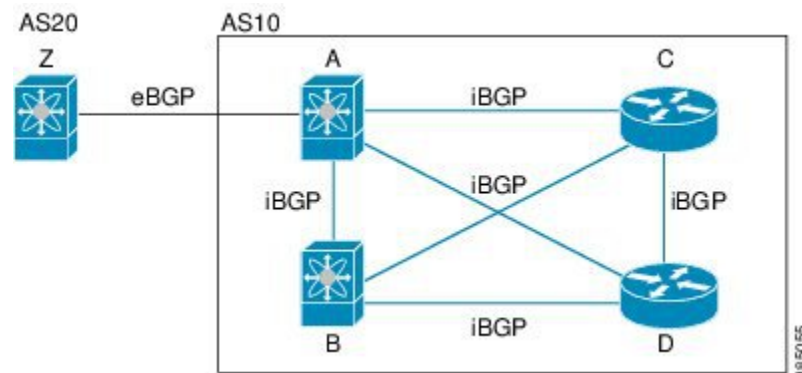
You should use loopback interfaces for establishing eBGP peering sessions because loopback interfaces are less susceptible to interface flapping. An interface flap occurs when the interface is administratively brought up or down because of a failure or maintenance issue. See the [Configuring eBGP](#) section for information on multihop, fast external failovers, and limiting the size of the AS-path attribute

iBGP

Internal BGP (iBGP) allows you to connect BGP peers within the same autonomous system. You can use iBGP for multihomed BGP networks (networks that have more than one connection to the same external autonomous system).

The following figure shows an iBGP network within a larger BGP network.

Figure 1: iBGP Network



iBGP networks are fully meshed. Each iBGP peer has a direct connection to all other iBGP peers to prevent network loops.



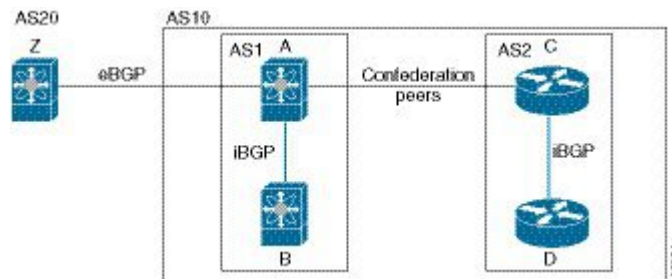
Note You should configure a separate interior gateway protocol in the iBGP network.

AS Confederations

A fully meshed iBGP network becomes complex as the number of iBGP peers grows. You can reduce the iBGP mesh by dividing the autonomous system into multiple subautonomous systems and grouping them into a single confederation. A confederation is a group of iBGP peers that use the same autonomous system number to communicate to external networks. Each subautonomous system is fully meshed within itself and has a few connections to other subautonomous systems in the same confederation.

The following figure shows the BGP network from Figure below, split into two subautonomous systems and one confederation.

Figure 2: AS Confederation



In this example, AS10 is split into two subautonomous systems, AS1 and AS2. Each subautonomous system is fully meshed, but there is only one link between the subautonomous systems. By using AS confederations, you can reduce the number of links compared to the fully meshed autonomous system in Figure 1.

Route Reflector

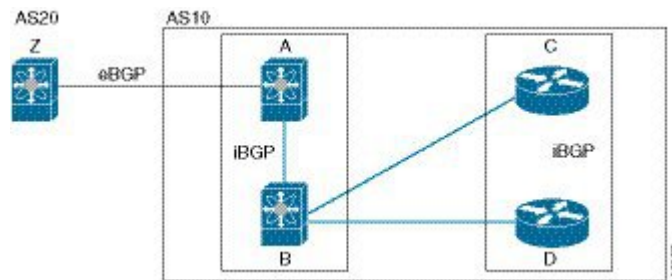
You can alternately reduce the iBGP mesh by using a route reflector configuration. Route reflectors pass learned routes to neighbors so that all iBGP peers do not need to be fully meshed.

Figure 1 shows a simple iBGP configuration with four meshed iBGP speakers (router A, B, C, and D). Without route reflectors, when router A receives a route from an external neighbor, it advertises the route to all three iBGP neighbors.

When you configure an iBGP peer to be a route reflector, it becomes responsible for passing iBGP learned routes to a set of iBGP neighbors.

In the following figure, router B is the route reflector. When the route reflector receives routes advertised from router A, it advertises (reflects) the routes to routers C and D. Router A no longer has to advertise to both routers C and D.

Figure 3: Route Reflector



The route reflector and its client peers form a cluster. You do not have to configure all iBGP peers to act as client peers of the route reflector. You must configure any nonclient peer as fully meshed to guarantee that complete BGP updates reach all peers.

Capabilities Negotiation

A BGP speaker can learn about BGP extensions supported by a peer by using the capabilities negotiation feature. Capabilities negotiation allows BGP to use only the set of features supported by both BGP peers on a link.

If a BGP peer does not support capabilities negotiation, Cisco NX-OS will attempt a new session to the peer without capabilities negotiation if you have configured the address family as IPv4.

Route Dampening

Route dampening is a BGP feature that minimizes the propagation of flapping routes across an internetwork. A route flaps when it alternates between the available and unavailable states in rapid succession.

For example, consider a network with three BGP autonomous systems: AS1, AS2, and AS3. Suppose that a route in AS1 flaps (it becomes unavailable). Without route dampening, AS1 sends a withdraw message to AS2. AS2 propagates the withdrawal message to AS3. When the flapping route reappears, AS1 sends an advertisement message to AS2, which sends the advertisement to AS3. If the route repeatedly becomes unavailable, and then available, AS1 sends many withdrawal and advertisement messages that propagate through the other autonomous systems.

Route dampening can minimize flapping. Suppose that the route flaps. AS2 (in which route dampening is enabled) assigns the route a penalty of 1000. AS2 continues to advertise the status of the route to neighbors. Each time that the route flaps, AS2 adds to the penalty value. When the route flaps so often that the penalty exceeds a configurable suppression limit, AS2 stops advertising the route, regardless of how many times that it flaps. The route is now dampened.

The penalty placed on the route decays until the reuse limit is reached. At that time, AS2 advertises the route again. When the reuse limit is at 50 percent, AS2 removes the dampening information for the route.



Note The router does not apply a penalty to a resetting BGP peer when route dampening is enabled, even though the peer reset withdraws the route.

Load Sharing and Multipath

BGP can install multiple equal-cost eBGP or iBGP paths into the routing table to reach the same destination prefix. Traffic to the destination prefix is then shared across all the installed paths.

The BGP best-path algorithm considers the paths as equal-cost paths if the following attributes are identical:

- Weight
- Local preference
- AS_path
- Origin code

- Multi-exit discriminator (MED)
- IGP cost to the BGP next hop

BGP selects only one of these multiple paths as the best path and advertises the path to the BGP peers.



Note Paths received from different AS confederations are considered as equal-cost paths if the external AS_path values and the other attributes are identical.



Note When you configure a route reflector for iBGP multipath, and the route reflector advertises the selected best path to its peers, the next hop for the path is not modified.

Route Aggregation

You can configure aggregate addresses. Route aggregation simplifies route tables by replacing a number of more specific addresses with an address that represents all the specific addresses. For example, you can replace these three more specific addresses, 10.1.1.0/24, 10.1.2.0/24, and 10.1.3.0/24 with one aggregate address, 10.1.0.0/16.

Aggregate prefixes are present in the BGP route table so that fewer routes are advertised.



Note Cisco NX-OS does not support automatic route aggregation.

Route aggregation can lead to forwarding loops. To avoid this problem, when BGP generates an advertisement for an aggregate address, it automatically installs a summary discard route for that aggregate address in the local routing table. BGP sets the administrative distance of the summary discard to 220 and sets the route type to discard. BGP does not use discard routes for next-hop resolution.

BGP Conditional Advertisement

BGP conditional advertisement allows you to configure BGP to advertise or withdraw a route based on whether or not a prefix exists in the BGP table. This feature is useful, for example, in multihomed networks, in which you want BGP to advertise some prefixes to one of the providers only if information from the other provider is not present.

Consider an example network with three BGP autonomous systems: AS1, AS2, and AS3, where AS1 and AS3 connect to the Internet and to AS2. Without conditional advertisement, AS2 propagates all routes to both AS1 and AS3. With conditional advertisement, you can configure AS2 to advertise certain routes to AS3 only if routes from AS1 do not exist (if for example, the link to AS1 fails).

BGP conditional advertisement adds an exist or not-exist test to each route that matches the configured route map. See the [Configuring BGP Conditional Advertisement](#) section for more information.

BGP Next-Hop Address Tracking

BGP monitors the next-hop address of installed routes to verify next-hop reachability and to select, install, and validate the BGP best path. BGP next-hop address tracking speeds up this next-hop reachability test by triggering the verification process when routes change in the RIB that may affect BGP next-hop reachability.

BGP receives notifications from the RIB when next-hop information changes (event-driven notifications). BGP is notified when any of the following events occurs:

- Next hop becomes unreachable.
- Next hop becomes reachable.
- Fully recursed IGP metric to the next hop changes.
- First hop IP address or first hop interface changes.
- Next hop becomes connected.
- Next hop becomes unconnected.
- Next hop becomes a local address.
- Next hop becomes a nonlocal address.



Note Reachability and recursed metric events trigger a best-path recalculation.

Event notifications from the RIB are classified as critical and noncritical. Notifications for critical and noncritical events are sent in separate batches. However, a noncritical event is sent with the critical events if the noncritical event is pending and there is a request to read the critical events.

- Critical events are related to the reachability (reachable and unreachable), connectivity (connected and unconnected), and locality (local and nonlocal) of the next hops. Notifications for these events are not delayed.
- Noncritical events include only the IGP metric changes.

See the [Configuring BGP Next-Hop Address Tracking](#) section for more information.

Route Redistribution

You can configure BGP to redistribute static routes or routes from other protocols. You configure a route policy with the redistribution to control which routes are passed into BGP. A route policy allows you to filter routes based on attributes such as the destination, origination protocol, route type, route tag, and so on. See [Configuring Route Policy Manager](#), for more information.

Tuning BGP

You can modify the default behavior of BGP through BGP timers and by adjusting the best-path algorithm.

BGP Timers

BGP uses different types of timers for neighbor session and global protocol events. Each established session has a minimum of two timers for sending periodic keepalive messages and for timing out sessions when peer keepalives do not arrive within the expected time. In addition, there are other timers for handling specific features. Typically, you configure these timers in seconds. The timers include a random adjustment so that the same timers on different BGP peers trigger at different times.

Tuning the Best-Path Algorithm

You can modify the default behavior of the best-path algorithm through optional configuration parameters, including changing how the algorithm handles the MED attribute and the router ID.

Multiprotocol BGP

BGP on Cisco NX-OS supports multiple address families. Multiprotocol BGP (MP-BGP) carries different sets of routes depending on the address family. For example, BGP can carry one set of routes for IPv4 unicast routing. You can use MP-BGP for reverse-path forwarding (RPF) checks in IP multicast networks.



Note Because Multicast BGP does not propagate multicast state information, you need a multicast protocol, such as Protocol Independent Multicast (PIM).

Use the router address-family and neighbor address-family configuration modes to support multiprotocol BGP configurations. MP-BGP maintains separate RIBs for each configured address family, such as a unicast RIB and a multicast RIB for BGP.

A multiprotocol BGP network is backward compatible but BGP peers that do not support multiprotocol extensions cannot forward routing information, such as address family identifier information, that the multiprotocol extensions carry.

Prerequisites for BGP

BGP has the following prerequisites:

- You must enable the BGP feature (see the [Enabling the BGP Feature](#) section).
- You should have a valid router ID configured on the system.
- You must have an AS number, either assigned by a Regional Internet Registry (RIR) or locally administered.
- You must have reachability (such as an interior gateway protocol (IGP), a static route, or a direct connection) to the peer that you are trying to make a neighbor relationship with.
- You must explicitly configure an address family under a neighbor for the BGP session establishment.

Guidelines and Limitations for Advanced BGP

BGP has the following configuration guidelines and limitations:

- Prefix peering operates only in passive TCP mode. It accepts incoming connections from remote peers if the peer address falls within the prefix.
- The dynamic autonomous system number prefix peer configuration overrides the individual autonomous system number configuration that is inherited from a BGP template.
- If you configure a dynamic autonomous system number for prefix peers in an autonomous system confederation, BGP establishes sessions with only the autonomous system numbers in the local confederation.
- BGP sessions that are created through a dynamic autonomous system number prefix peer ignore any configured eBGP multihop time-to-live (TTL) value or a disabled check for directly connected peers.
- Configure a router ID for BGP to avoid automatic router ID changes and session flaps.
- Use the maximum-prefix configuration option per peer to restrict the number of routes that are received and system resources used.
- Configure the update-source to establish a session with eBGP multihop sessions.
- Specify a BGP route map if you configure redistribution.
- Configure the BGP router ID within a VRF.
- If you decrease the keepalive and hold timer values, the network might experience session flaps.
- For the **maximum-paths** argument, the Nexus 3500 hardware supports a maximum of 32 paths although the NX-OS CLI shows a supported maximum of 64. Use 32 as the upper limit for **maximum-paths**. If you configure more than 32 paths for **maximum-paths**, some features (for example, consistency checker) can operate unpredictably.

Default Settings for BGP

Table below lists the default settings for BGP parameters.

Table 1: Default BGP Parameters

Parameters	Default
BGP feature	disabled
keep alive interval	60 seconds
hold timer	180 seconds

Configuring Advanced BGP



Note If you are familiar with the Cisco IOS CLI, be aware that the Cisco NX-OS commands for this feature might differ from the Cisco IOS commands that you would use.

Configuring BGP Session Templates

You can use BGP session templates to simplify BGP configuration for multiple BGP peers with similar configuration needs. BGP templates allow you to reuse common configuration blocks. You configure BGP templates first, and then apply these templates to BGP peers.

With BGP session templates, you can configure session attributes such as inheritance, passwords, timers, and security.

A peer-session template can inherit from one other peer-session template. You can configure the second template to inherit from a third template. The first template also inherits this third template. This indirect inheritance can continue for up to seven peer-session templates.

Any attributes configured for the neighbor take priority over any attributes inherited by that neighbor from a BGP template.

Before you begin

Ensure that you have enabled the BGP feature (see the [Enabling the BGP Feature](#) section).

SUMMARY STEPS

1. **configure terminal**
2. **router bgp** *autonomous-system-number*
3. **template peer-session** *template-name*
4. (Optional) **password** *number password*
5. (Optional) **timers** *keepalive hold*
6. **exit**
7. **neighbor** *ip-address* **remote-as** *as-number*
8. **inherit peer-session** *template-name*
9. (Optional) **description** *text*
10. (Optional) **show bgp peer-session** *template-name*
11. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure terminal Example: <pre>switch# configure terminal switch(config)#</pre>	Enters configuration mode.

	Command or Action	Purpose
Step 2	router bgp <i>autonomous-system-number</i> Example: <pre>switch(config)# router bgp 65536 switch(config-router)#</pre>	Enables BGP and assigns the autonomous system number to the local BGP speaker.
Step 3	template peer-session <i>template-name</i> Example: <pre>switch(config-router)# template peer-session BaseSession switch(config-router-stmp)#</pre>	Enters peer-session template configuration mode.
Step 4	(Optional) password <i>number password</i> Example: <pre>switch(config-router-stmp)# password 0 test</pre>	Adds the clear text password <i>test</i> to the neighbor. The password is stored and displayed in type 3 encrypted form (3DES).
Step 5	(Optional) timers <i>keepalive hold</i> Example: <pre>switch(config-router-stmp)# timers 30 90</pre>	Adds the BGP keepalive and holdtimer values to the peer-session template. The default keepalive interval is 60. The default hold time is 180.
Step 6	exit Example: <pre>switch(config-router-stmp)# exit switch(config-router)#</pre>	Exits peer-session template configuration mode.
Step 7	neighbor ip-address remote-as <i>as-number</i> Example: <pre>switch(config-router)# neighbor 192.168.1.2 remote-as 65536 switch(config-router-neighbor)#</pre>	Places the router in the neighbor configuration mode for BGP routing and configures the neighbor IP address.
Step 8	inherit peer-session <i>template-name</i> Example: <pre>switch(config-router-neighbor)# inherit peer-session BaseSession switch(config-router-neighbor)</pre>	Applies a peer-session template to the peer.
Step 9	(Optional) description <i>text</i> Example: <pre>switch(config-router-neighbor)# description Peer Router A switch(config-router-neighbor)</pre>	Adds a description for the neighbor.
Step 10	(Optional) show bgp peer-session <i>template-name</i> Example: <pre>switch(config-router-neighbor)# show bgp peer-session BaseSession</pre>	Displays the peer-policy template.

	Command or Action	Purpose
Step 11	(Optional) copy running-config startup-config Example: <pre>switch(config-router-neighbor)# copy running-config startup-config</pre>	Saves this configuration change.

Example

Use the **show bgp neighbor** command to see the template applied. See the [Cisco Nexus 3000 Series Command Reference](#) for details on all commands available in the template.

This example shows how to configure a BGP peer-session template and apply it to a BGP peer:

```
switch# configure terminal
switch(config)# router bgp 65536
switch(config-router)# template peer-session BaseSession
switch(config-router-stmp)# timers 30 90
switch(config-router-stmp)# exit
switch(config-router)# neighbor 192.168.1.2 remote-as 65536
switch(config-router-neighbor)# inherit peer-session BaseSession
switch(config-router-neighbor)# description Peer Router A
switch(config-router-neighbor)# address-family ipv4 unicast
switch(config-router-neighbor)# copy running-config startup-config
```

Configuring BGP Peer-Policy Templates

You can configure a peer-policy template to define attributes for a particular address family. You assign a preference to each peer-policy template and these templates are inherited in the order specified, for up to five peer-policy templates in a neighbor address family.

Cisco NX-OS evaluates multiple peer policies for an address family using the preference value. The lowest preference value is evaluated first. Any attributes configured for the neighbor take priority over any attributes inherited by that neighbor from a BGP template.

Peer-policy templates can configure address family-specific attributes such as AS-path filter lists, prefix lists, route reflection, and soft reconfiguration.

Before you begin

Ensure that you have enabled the BGP feature (see the [Enabling the BGP Feature](#) section).

SUMMARY STEPS

1. **configure terminal**
2. **router bgp** *autonomous-system-number*
3. **template peer-policy** *template-name*
4. (Optional) **advertise-active-only**
5. (Optional) **maximum-prefix** *number*
6. **exit**
7. **neighbor** *ip-address* **remote-as** *as-number*
8. **address-family ipv4 unicast**

9. **inherit peer-policy** *template-name preference*
10. (Optional) **show bgp peer-policy** *template-name*
11. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure terminal Example: <pre>switch# configure terminal switch(config)#</pre>	Enters configuration mode.
Step 2	router bgp <i>autonomous-system-number</i> Example: <pre>switch(config)# router bgp 65536 switch(config-router)#</pre>	Enables BGP and assigns the autonomous system number to the local BGP speaker.
Step 3	template peer-policy <i>template-name</i> Example: <pre>switch(config-router)# template peer-policy BasePolicy switch(config-router-ptmp)#</pre>	Creates a peer-policy template.
Step 4	(Optional) advertise-active-only Example: <pre>switch(config-router-ptmp)# advertise-active-only</pre>	Advertises only active routes to the peer.
Step 5	(Optional) maximum-prefix <i>number</i> Example: <pre>switch(config-router-ptmp)# maximum-prefix 20</pre>	Sets the maximum number of prefixes allowed from this peer.
Step 6	exit Example: <pre>switch(config-router-ptmp)# exit switch(config-router)#</pre>	Exits peer-policy template configuration mode.
Step 7	neighbor ip-address remote-as <i>as-number</i> Example: <pre>switch(config-router)# neighbor 192.168.1.2 remote-as 65536 switch(config-router-neighbor)#</pre>	Places the router in neighbor configuration mode for BGP routing and configures the neighbor IP address.
Step 8	address-family ipv4 unicast Example: <pre>switch(config-router-neighbor)# address-family ipv4 unicast switch(config-router-neighbor-af)#</pre>	Enters global address family configuration mode for the specified address family.

	Command or Action	Purpose
Step 9	inherit peer-policy <i>template-name preference</i> Example: switch(config-router-neighbor-af)# inherit peer-policy BasePolicy 1	Applies a peer-policy template to the peer address family configuration and assigns the preference value for this peer policy.
Step 10	(Optional) show bgp peer-policy <i>template-name</i> Example: switch(config-router-neighbor-af)# show bgp peer-policy BasePolicy	Displays the peer-policy template.
Step 11	(Optional) copy running-config startup-config Example: switch(config-router-neighbor)# copy running-config startup-config	Saves this configuration change.

Example

Use the **show bgp neighbor** command to see the template applied. See the [Cisco Nexus 3000 Series Command Reference](#) for details on all commands available in the template.

This example shows how to configure a BGP peer-session template and apply it to a BGP peer:

```
switch# configure terminal
switch(config)# router bgp 65536
switch(config-router)# template peer-session BasePolicy
switch(config-router-ptmp)# maximum-prefix 20
switch(config-router-ptmp)# exit
switch(config-router)# neighbor 192.168.1.1 remote-as 65536
switch(config-router-neighbor)# address-family ipv4 unicast
switch(config-router-neighbor-af)# inherit peer-policy BasePolicy
switch(config-router-neighbor-af)# copy running-config startup-config
```

Configuring BGP Peer Templates

You can configure BGP peer templates to combine session and policy attributes in one reusable configuration block. Peer templates can also inherit peer-session or peer-policy templates. Any attributes configured for the neighbor take priority over any attributes inherited by that neighbor from a BGP template. You configure only one peer template for a neighbor, but that peer template can inherit peer-session and peer-policy templates.

Peer templates support session and address family attributes, such as eBGP multihop time-to-live, maximum prefix, next-hop self, and timers.

Before you begin

Ensure that you have enabled the BGP feature (see the [Enabling the BGP Feature](#) section).

SUMMARY STEPS

1. **configure terminal**
2. **router bgp** *autonomous-system-number*

3. **template peer-session** *template-name*
4. (Optional) **inherit peer-session** *template-name*
5. **address-family ipv4 unicast**
6. (Optional) **inherit peer** *template-name*
7. **exit**
8. (Optional) **timers keepalive hold**
9. **exit**
10. **neighbor ip-address remote-as as-number**
11. **inherit peer** *template-name*
12. (Optional) **timers keepalive hold**
13. (Optional) **show bgp peer-template** *template-name*
14. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure terminal Example: <pre>switch# configure terminal switch(config)#</pre>	Enters configuration mode.
Step 2	router bgp <i>autonomous-system-number</i> Example: <pre>switch(config)# router bgp 65536 switch(config-router)#</pre>	Enables BGP and assigns the autonomous system number to the local BGP speaker.
Step 3	template peer-session <i>template-name</i> Example: <pre>switch(config-router)# template peer-session BaseSession switch(config-router-stmp)#</pre>	Enters peer-session template configuration mode.
Step 4	(Optional) inherit peer-session <i>template-name</i> Example: <pre>switch(config-router-neighbor)# inherit peer-session BaseSession</pre>	Inherits a peer-session template in the peer template.
Step 5	address-family ipv4 unicast Example: <pre>switch(config-router-neighbor)# address-family ipv4 unicast switch(config-router-neighbor-af)#</pre>	Configures the global address family configuration mode for the specified address family.
Step 6	(Optional) inherit peer <i>template-name</i> Example: <pre>switch(config-router-neighbor-af)# inherit peer BasePolicy</pre>	Applies a peer template to the neighbor address family configuration.

	Command or Action	Purpose
Step 7	exit Example: switch(config-router-neighbor-af) # exit switch(config-router-neighbor) #	Exits BGP neighbor address family configuration mode.
Step 8	(Optional) timers <i>keepalive hold</i> Example: switch(config-router-neighbor) # timers 45 100	Adds the BGP timer values to the peer. These values override the timer values in the peer-session template, BaseSession.
Step 9	exit Example: switch(config-router-stmp) # exit switch(config-router) #	Exits BGP peer template configuration mode.
Step 10	neighbor <i>ip-address remote-as as-number</i> Example: switch(config-router) # neighbor 192.168.1.2 remote-as 65536 switch(config-router-neighbor) #	Places the router in the neighbor configuration mode for BGP routing and configures the neighbor IP address.
Step 11	inherit peer <i>template-name</i> Example: switch(config-router-neighbor) # inherit peer BasePeer	Inherits the peer template.
Step 12	(Optional) timers <i>keepalive hold</i> Example: switch(config-router-neighbor) # timers 60 120	Adds the BGP timer values to this neighbor. These values override the timer values in the peer template and the peer-session template.
Step 13	(Optional) show bgp peer-template <i>template-name</i> Example: switch(config-router-neighbor-af) # show bgp peer-template BasePeer	Displays the peer template.
Step 14	(Optional) copy running-config startup-config Example: switch(config-router-neighbor) # copy running-config startup-config	Saves this configuration change.

Example

Use the **show bgp neighbor** command to see the template applied. See the [Cisco Nexus 3000 Series Command Reference](#) for details on all commands available in the template.

This example shows how to configure a BGP peer template and apply it to a BGP peer:


```

switch# configure terminal
switch(config)# router bgp 65536
switch(config-router)# template peer BasePeer
switch(config-router-neighbor)# inherit peer-session BaseSession
switch(config-router-neighbor)# address-family ipv4 unicast
switch(config-router-neighbor-af)# inherit peer-policy BasePolicy 1
switch(config-router-neighbor-af)# exit
switch(config-router-neighbor)# exit
switch(config-router)# neighbor 192.168.1.2 remote-as 65536
switch(config-router-neighbor)# inherit peer BasePeer
switch(config-router-neighbor)# copy running-config startup-config

```

Configuring BGP Interface Peering via IPv6 Link-Local for IPv4 and IPv6 Address Families

You can configure BGP Interface Peering via IPv6 Link-Local for IPv4 and IPv6 Address Families for automatic BGP neighbor discovery using unnumbered interfaces. Doing so allows you to set up BGP sessions using an interface name as a BGP peer (rather than interface-scoped addresses). This feature relies on ICMPv6 neighbor discovery (ND) route advertisement (RA) for automatic neighbor discovery and on RFC 5549 for sending IPv4 routes with IPv6 next hop.

Before you begin

You must enable BGP (see the [Enabling the BGP Feature](#) section).

SUMMARY STEPS

1. **configure terminal**
2. **router bgp** *autonomous-system-number*
3. **neighbor** *interface-name* **remote-as** *as-number*
4. **inherit peer** *template-name*
5. **address-family** {**ipv4** | **ipv6**} **unicast**
6. (Optional) **show bgp** {**ipv4** | **ipv6**} **unicast neighbors** *interface*
7. (Optional) **show ip bgp neighbors** *interface-name*
8. (Optional) **show ipv6 routers** [**interface** *interface*]
9. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure terminal Example: switch# configure terminal	Enters configuration mode.
Step 2	router bgp <i>autonomous-system-number</i> Example: switch(config)# router bgp 65535 switch(config-router)#	Enables BGP and assigns the autonomous system number to the local BGP speaker. The AS number can be a 16-bit integer or a 32-bit integer in the form of a higher 16-bit decimal number and a lower 16-bit decimal number in xx.xx format.

	Command or Action	Purpose
Step 3	neighbor <i>interface-name</i> remote-as <i>as-number</i> Example: <pre>switch(config-router)# neighbor Ethernet1/1 remote-as 65535 switch(config-router-neighbor)#</pre>	Places the router in the neighbor configuration mode for BGP routing and configures the interface for BGP peering. Note You can specify only Ethernet interfaces, port-channel interfaces, subinterfaces, and breakout interfaces.
Step 4	inherit peer <i>template-name</i> Example: <pre>switch(config-router-neighbor)# inherit peer PEER</pre>	Inherits the peer template.
Step 5	address-family { ipv4 ipv6 } unicast Example: <pre>switch(config-router-neighbor)# address-family ipv4 unicast switch(config-router-neighbor-af)#</pre>	Enters global address family configuration mode for the address family specified.
Step 6	(Optional) show bgp { ipv4 ipv6 } unicast neighbors <i>interface</i> Example: <pre>switch(config-router-neighbor-af)# show bgp ipv4 unicast neighbors e1/25</pre> Example: <pre>switch(config-router-neighbor-af)# show bgp ipv6 unicast neighbors 3FFE:700:20:1::11</pre>	Displays information about BGP peers.
Step 7	(Optional) show ip bgp neighbors <i>interface-name</i> Example: <pre>switch(config-router-neighbor-af)# show ip bgp neighbors Ethernet1/1</pre>	Displays the interface used as a BGP peer.
Step 8	(Optional) show ipv6 routers [interface <i>interface</i>] Example: <pre>switch(config-router-neighbor-af)# show ipv6 routers interface Ethernet1/1</pre>	Displays the link-local address of remote IPv6 routers, which is learned through IPv6 ICMP router advertisement.
Step 9	(Optional) copy running-config startup-config Example: <pre>switch(config-router-neighbor-af)# copy running-config startup-config</pre>	Saves this configuration change.

Example

This example shows how to configure BGP Interface Peering via IPv6 Link-Local for IPv4 and IPv6 Address Families.

iBGP Interface Peering Configuration for Leaf 1:

```

switch# configure terminal
switch(config)# router bgp 65000
switch(config-router)# neighbor Ethernet1/1 remote-as 65000
switch(config-router-neighbor)# inherit peer PEER
switch(config-router-neighbor)# address-family ipv4 unicast
switch(config-router-neighbor)# address-family ipv6 unicast
switch(config-router-neighbor-af)# copy running-config startup-config

```

This example shows sample output for BGP Interface Peering via IPv6 Link-Local for IPv4 and IPv6 Address Families:

```

switch(config-router-neighbor)# show bgp ipv4 unicast neighbors e1/15.1
BGP neighbor is fe80::2, remote AS 100, ibgp link, Peer index 4
Peer is an instance of interface peering Ethernet1/15.1
BGP version 4, remote router ID 5.5.5.5
Neighbor previous state = OpenConfirm
BGP state = Established, up for 2d16h
Neighbor vrf: default
Peer is directly attached, interface Ethernet1/15.1
Last read 00:00:54, hold time = 180, keepalive interval is 60 seconds
Last written 00:00:08, keepalive timer expiry due 00:00:51
Received 3869 messages, 0 notifications, 0 bytes in queue
Sent 3871 messages, 0 notifications, 0(0) bytes in queue
Enhanced error processing: On
0 discarded attributes
Connections established 2, dropped 1
Last reset by peer 2d16h, due to session closed
Last error length received: 0
Reset error value received 0
Reset error received major: 104 minor: 0
Notification data received:
Last reset by us never, due to No error
Last error length sent: 0
Reset error value sent: 0
Reset error sent major: 0 minor: 0
--More--

```

Interface Configuration:

IPv6 needs to be enabled on the corresponding interface using one of the following commands:

- **ipv6 address** *ipv6-address*
- **ipv6 address use-link-local-only**
- **ipv6 link-local** *link-local-address*

```

switch# configure terminal
switch(config)# interface Ethernet1/1
switch(config-if)# ipv6 address use-link-local-only

```



Note If an IPv4 address is not configured on the interface, the **ip forward** command must be configured on the interface to enable IPv4 forwarding.



Note IPv6 ND timers can be tuned to speed up neighbor discovery and for BGP faster route convergence.

```
switch(config-if) # ipv6 nd ra-interval 4 min 3
switch(config-if) # ipv6 nd ra-lifetime 10
```

Configuring BGP Authentication

You can configure BGP to authenticate route updates from peers using MD5 digests.

To configure BGP to use MD5 authentication, use the following command in neighbor configuration mode:

Command	Purpose
password [0 3 7] <i>string</i> Example : <pre>switch(config-router-neighbor) # password BGPpassword</pre>	Configures an MD5 password for BGP neighbor sessions.

Resetting a BGP Session

If you modify a route policy for BGP, you must reset the associated BGP peer sessions. If the BGP peers do not support route refresh, you can configure a soft reconfiguration for inbound policy changes. Cisco NX-OS automatically attempts a soft reset for the session.

To configure soft reconfiguration inbound, use the following command in neighbor address-family configuration mode:

Command	Purpose
soft-reconfiguration inbound Example : <pre>switch(config-router-neighbor-af) # soft-reconfiguration inbound</pre>	Enables soft reconfiguration to store the inbound BGP route updates. This command triggers an automatic soft clear or refresh of BGP neighbor sessions.

To reset a BGP neighbor session, use the following command in any mode:

Command	Purpose
clear bgp ip { unicast } <i>ip-address</i> soft { in out } Example : <pre>switch# clear bgp ip unicast 192.0.2.1 soft in</pre>	Resets the BGP session without tearing down the TCP session.

Modifying the Next-Hop Address

You can modify the next-hop address used in a route advertisement in the following ways:

- Disable the next-hop calculation and use the local BGP speaker address as the next-hop address.

- Set the next-hop address as a third-party address. Use this feature in situations where the original next-hop address is on the same subnet as the peer that the route is being sent to. Using this feature saves an extra hop during forwarding.

To modify the next-hop address, use the following parameters in commands address-family configuration mode:

Command	Purpose
next-hop-self Example : <pre>switch(config-router-neighbor-af) # next-hop-self</pre>	Uses the local BGP speaker address as the next-hop address in route updates. This command triggers an automatic soft clear or refresh of BGP neighbor sessions.
next-hop-third-party Example : <pre>switch(config-router-neighbor-af) # next-hop-third-party</pre>	Sets the next-hop address as a third-party address. Use this command for single-hop EBGP peers that do not have next-hop-self configured.

Configuring BGP Next-Hop Address Tracking

BGP next-hop address tracking is enabled by default and cannot be disabled.

You can modify the delay interval between RIB checks to increase the performance of BGP next-hop tracking. You can configure the critical timer for routes that affect BGP next-hop reachability, and you can configure the noncritical timer for all other routes in the BGP table.

To modify the BGP next-hop address tracking, use the following commands address-family configuration mode:

Command	Purpose
nexthop trigger-delay {critical non-critical} milliseconds Example : <pre>switch(config-router-af) # nexthop trigger-delay critical 5000</pre>	Specifies the next-hop address tracking delay timer for critical next-hop reachability routes and for noncritical routes. The range is from 1 to 4294967295 milliseconds. The critical timer default is 3000. The noncritical timer default is 10000.
nexthop route-map name Example : <pre>switch(config-router-af) # nexthop route-map nextHopLimits</pre>	Specifies a route map to match the BGP next-hop addresses to. The name can be any case-sensitive, alphanumeric string up to 63 characters.

Configuring Next-Hop Filtering

BGP next-hop filtering allows you to specify that when a next-hop address is checked with the RIB, the underlying route for that next-hop address is passed through the route map. If the route map rejects the route, the next-hop address is treated as unreachable.

BGP marks all next hops that are rejected by the route policy as invalid and does not calculate the best path for the routes that use the invalid next-hop address.

To configure BGP next-hop filtering, use the following command in address-family configuration mode:

Command	Purpose
next-hop route-map <i>name</i> Example : <pre>switch(config-router-af)# next-hop route-map nextHopLimits</pre>	Specifies a route map to match the BGP next-hop route to. The name can be any case-sensitive, alphanumeric string up to 63 characters.

Controlling Reflected Routes Through Next-Hop-Self

NX-OS enables controlling the iBGP routes being sent to a specific peer through the **next-hop-self** [all] arguments. By using these arguments, you can selectively change the next-hop of routes even if the route is reflected.

Command	Purpose
next-hop-self [all] Example: <pre>switch(config-router-af)# next-hop-self all</pre>	Uses the local BGP speaker address as the next-hop address in route updates. The all keyword is optional. If you specify all, all routes are sent to the peer with next-hop-self. If you do not specify all, the next hops of reflected routes are not changed.

Disabling Capabilities Negotiation

You can disable capabilities negotiations to interoperate with older BGP peers that do not support capabilities negotiation.

To disable capabilities negotiation, use the following command in neighbor configuration mode:

Command	Purpose
dont-capability-negotiate Example : <pre>switch(config-router-neighbor)# dont-capability-negotiate</pre>	Disables capabilities negotiation. You must manually reset the BGP sessions after configuring this command.

Configuring eBGP

This section includes the following topics:

Disabling eBGP Single-Hop Checking

You can configure eBGP to disable checking whether a single-hop eBGP peer is directly connected to the local router. Use this option for configuring a single-hop loopback eBGP session between directly connected switches.

To disable checking whether or not a single-hop eBGP peer is directly connected, use the following command in neighbor configuration mode:

Command	Purpose
disable-connected-check Example : <pre>switch(config-router-neighbor) # disable-connected-check</pre>	Disables checking whether or not a single-hop eBGP peer is directly connected. You must manually reset the BGP sessions after using this command.

Configuring eBGP Multihop

You can configure the eBGP time-to-live (TTL) value to support eBGP multihop. In some situations, an eBGP peer is not directly connected to another eBGP peer and requires multiple hops to reach the remote eBGP peer. You can configure the eBGP TTL value for a neighbor session to allow these multihop sessions.

To configure eBGP multihop, use the following command in neighbor configuration mode:

Command	Purpose
ebgp-multihop <i>ttl-value</i> Example : <pre>switch(config-router-neighbor) # ebgp-multihop 5</pre>	Configures the eBGP TTL value for eBGP multihop. The range is from 2 to 255. You must manually reset the BGP sessions after using this command.

Disabling a Fast External Failover

Typically, when a BGP router loses connectivity to a directly connected eBGP peer, BGP triggers a fast external failover by resetting the eBGP session to the peer. You can disable this fast external failover to limit the instability caused by link flaps.

To disable fast external failover, use the following command in router configuration mode:

Command	Purpose
no fast-external-failover Example : <pre>switch(config-router) # no fast-external-failover</pre>	Disables a fast external failover for eBGP peers. This command is enabled by default.

Limiting the AS-path Attribute

You can configure eBGP to discard routes that have a high number of AS numbers in the AS-path attribute.

To discard routes that have a high number of AS numbers in the AS-path attribute, use the following command in router configuration mode:

Command	Purpose
maxas-limit <i>number</i> Example : <pre>switch(config-router)# maxas-limit 50</pre>	Discards eBGP routes that have a number of AS-path segments that exceed the specified limit. The range is from 1 to 2000.

Configuring Local AS Support

The local AS feature allows a router to appear to be a member of a second autonomous system (AS), in addition to its real AS. Local AS allows two ISPs to merge without modifying peering arrangements. Routers in the merged ISP become members of the new autonomous system but continue to use their old AS numbers for their customers.

Local AS can only be used for true eBGP peers. You cannot use this feature for two peers that are members of different confederation subautonomous systems.

To configure eBGP local AS support, use the following command in neighbor configuration mode:

Command	Purpose
local-as <i>number</i> [no-prepend [replace-as [dual-as]]] Example : <pre>switch(config-router-neighbor)# local-as 1.1</pre>	<p>Configures eBGP to prepend the local AS number to the AS_PATH attribute.</p> <p>The local AS number can be a 16-bit integer or a 32-bit integer in the form of a higher 16-bit decimal number and a lower 16-bit decimal number in xx.xx format.</p> <p>The no-prepend keyword ensures that the local AS number is not prepended to any downstream BGP neighbors except for the partner who is peering with the local AS number.</p> <p>The replace-as keyword ensures that only the local AS number of the peering session is prepended to the AS_PATH attribute. The autonomous-system number from the local BGP routing process is not prepended.</p> <p>The dual-as keyword configures the eBGP neighbor to establish a peering session using the real autonomous-system number (from the local BGP routing process) or by using the autonomous-system number configured as the Local AS).</p>

Configuring AS Confederations

To configure an AS confederation, you must specify a confederation identifier. The group of autonomous systems within the AS confederation looks like a single autonomous system with the confederation identifier as the autonomous system number.

To configure a BGP confederation identifier, use the following command in router configuration mode:

Command	Purpose
confederation identifier <i>as-number</i> Example : <pre>switch(config-router)# confederation identifier 64512</pre>	Configures a confederation identifier for an AS confederation. Each confederation has a different sub-AS number, usually a private one (from 64512 to 65534). This command triggers an automatic notification and session reset for the BGP neighbor sessions.

To configure the autonomous systems that belong to the AS confederation, use the following command in router configuration mode:

Command	Purpose
bgp confederation peers <i>as-number</i> [<i>as-number2...</i>] Example : <pre>switch(config-router)# bgp confederation peers 5 33 44</pre>	Specifies a list of autonomous systems that belong to the confederation. This command triggers an automatic notification and session reset for the BGP neighbor sessions.

Configuring an Autonomous System Path Containing Your Own Autonomous System

Enable the feature for BGP to accept the autonomous system (AS) path that contains your own AS.

Before you begin

Ensure that you have enabled the BGP feature (see the [Enabling the BGP Feature](#) section).

SUMMARY STEPS

1. **configure terminal**
2. **router bgp** *as-number*
3. **neighbor** *ip-address* **remote-as** *as-number*
4. **address-family ipv4 unicast**
5. [**no** | **default**] **allowas-in** [*allowas-in-cnt*]
6. **end**
7. (Optional) **show running-config bgp**
8. (Optional) **copy running-config startup-config**

DETAILED STEPS

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

	Command or Action	Purpose
	switch# configure terminal switch(config)#	
Step 2	router bgp <i>as-number</i> Example: switch(config)# router bgp <i>as-number</i>	Enters BGP mode and assigns the autonomous system number to the local BGP speaker. The <i>as-number</i> value range is from 1 to 65535.
Step 3	neighbor <i>ip-address</i> remote-as <i>as-number</i> Example: switch(config-router)# neighbor 192.168.1.2 remote-as 65536 switch(config-router-neighbor)#	Places the router in neighbor configuration mode for BGP routing and configures the neighbor IP address.
Step 4	address-family ipv4 unicast Example: switch(config-router-neighbor)# address-family ipv4 unicast switch(config-router-neighbor-af)#	Enters router address family configuration for the specified address family.
Step 5	[no default] allowas-in [<i>allowas-in-cnt</i>] Example: switch(config-router-neighbor-af)# allowas-in 5	Enables the allowas-in feature for BGP and configures the number of occurrences of the AS number. For <i>allowas-in-cnt</i> , enter an integer between 1 and 10. By default, the number of occurrences of the AS number is set to 3.
Step 6	end Example: switch(config-router-neighbor-af)# end	Exits router address family configuration mode.
Step 7	(Optional) show running-config bgp Example: switch# show running-config bgp	Displays the BGP configuration.
Step 8	(Optional) copy running-config startup-config Example: switch(config-router-neighbor)# copy running-config startup-config	Saves this configuration change.

Example

This example shows how to configure the allowas-in feature for BGP and configure it for a unicast IPv4 address family:

```
switch# configure terminal
switch(config)# router bgp 77
switch(config-router)# neighbor 6.20.1.1 remote-as 66
switch(config-router-neighbor)# address-family ipv4 unicast
switch(config-router-neighbor-af)# allowas-in 5
switch(config-router-neighbor-af)# end
```

Configuring Route Reflector

You can configure iBGP peers as route reflector clients to the local BGP speaker, which acts as the route reflector. Together, a route reflector and its clients form a cluster. A cluster of clients usually has a single route reflector. In such instances, the cluster is identified by the router ID of the route reflector. To increase redundancy and avoid a single point of failure in the network, you can configure a cluster with more than one route reflector. You must configure all route reflectors in the cluster with the same 4-byte cluster ID so that a route reflector can recognize updates from route reflectors in the same cluster.

Before you begin

Ensure that you have enabled the BGP feature (see the [Enabling the BGP Feature](#) section).

SUMMARY STEPS

1. **configure terminal**
2. **router bgp** *as-number*
3. **cluster-id** *cluster-id*
4. **address-family ipv4 unicast**
5. (Optional) **client-to-client reflection**
6. **exit**
7. **neighbor** *ip-address* **remote-as** *as-number*
8. **address-family ipv4 unicast**
9. **route-reflector-client**
10. **show bgp ip unicast neighbors**
11. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure terminal Example: <pre>switch# configure terminal switch(config)#</pre>	Enters configuration mode.
Step 2	router bgp <i>as-number</i> Example: <pre>switch(config)# router bgp 65536 switch(config-router)#</pre>	Enables BGP and assigns the autonomous system number to the local BGP speaker.
Step 3	cluster-id <i>cluster-id</i> Example: <pre>switch(config-router)# cluster-id 192.0.2.1</pre>	Configures the local router as one of the route reflectors that serve the cluster. You specify a cluster ID to identify the cluster. This command triggers an automatic soft clear or refresh of BGP neighbor sessions.
Step 4	address-family ipv4 unicast Example:	Enters router address family configuration mode for the specified address family.

	Command or Action	Purpose
	<pre>switch(config-router-neighbor) # address-family ipv4 unicast switch(config-router-neighbor-af) #</pre>	
Step 5	<p>(Optional) client-to-client reflection</p> <p>Example:</p> <pre>switch(config-router-af) # client-to-client reflection</pre>	Configures client-to-client route reflection. This feature is enabled by default. This command triggers an automatic soft clear or refresh of BGP neighbor sessions.
Step 6	<p>exit</p> <p>Example:</p> <pre>switch(config-router-neighbor-af) # exit switch(config-router-neighbor) #</pre>	Exits router address configuration mode.
Step 7	<p>neighbor ip-address remote-as as-number</p> <p>Example:</p> <pre>switch(config-router) # neighbor 192.168.1.2 remote-as 65536 switch(config-router-neighbor) #</pre>	Configures the IP address and AS number for a remote BGP peer.
Step 8	<p>address-family ipv4 unicast</p> <p>Example:</p> <pre>switch(config-router-neighbor) # address-family ipv4 unicast switch(config-router-neighbor-af) #</pre>	Enters neighbor address family configuration mode for the specified address family.
Step 9	<p>route-reflector-client</p> <p>Example:</p> <pre>switch(config-router-neighbor-af) # route-reflector-client</pre>	Configures the switch as a BGP route reflector and configures the neighbor as its client. This command triggers an automatic notification and session reset for the BGP neighbor sessions.
Step 10	<p>show bgp ip unicast neighbors</p> <p>Example:</p> <pre>switch(config-router-neighbor-af) # show bgp ip unicast neighbors</pre>	Displays the BGP peers.
Step 11	<p>(Optional) copy running-config startup-config</p> <p>Example:</p> <pre>switch(config-router-neighbor) # copy running-config startup-config</pre>	Saves this configuration change.

Example

This example shows how to configure the router as a route reflector and add one neighbor as a client:

```
switch(config) # router bgp 65536
switch(config-router) # neighbor 192.0.2.10 remote-as 65536
switch(config-router-neighbor) # address-family ip unicast
```

```
switch(config-router-neighbor-af) # route-reflector-client
switch(config-router-neighbor-af) # copy running-config startup-config
```

Configuring Route Dampening

You can configure route dampening to minimize route flaps propagating through your iBGP network.

To configure route dampening, use the following command in address-family or VRF address family configuration mode:

Command	Purpose
<p>dampening [{ <i>half-life reuse-limit suppress-limit max-suppress-time</i> route-map map-name }]</p> <p>Example :</p> <pre>switch(config-router-af) # dampening route-map bgpDamp</pre>	<p>Disables capabilities negotiation. The parameter values are as follows:</p> <ul style="list-style-type: none"> • <i>half-life</i>—The range is from 1 to 45. • <i>reuse-limit</i>—The range is from 1 to 20000. • <i>suppress-limit</i>—The range is from 1 to 20000 • <i>max-suppress-time</i>—The range is from 1 to 255.

Configuring Load Sharing and ECMP

You can configure the maximum number of paths that BGP adds to the route table for equal-cost multipath load balancing.



Note ECMP is not supported on warp mode.

To configure the maximum number of paths, use the following command in router address-family configuration mode:

Command	Purpose
<p>maximum-paths [ibgp] <i>maxpaths</i></p> <p>Example :</p> <pre>switch(config-router-af) # maximum-paths 12</pre>	<p>Configures the maximum number of equal-cost paths for load sharing. The range is from 1 to 32. The default is 1.</p> <p>Note It is recommended you set the maximum-path to 1 in warp mode to prevent ECMP configuration.</p>

Configuring Maximum Prefixes

You can configure the maximum number of prefixes that BGP can receive from a BGP peer. If the number of prefixes exceeds this value, you can optionally configure BGP to generate a warning message or tear down the BGP session to the peer.

To configure the maximum allowed prefixes for a BGP peer, use the following command in neighbor address-family configuration mode:

Command	Purpose
maximum-prefix <i>maximum</i> [<i>threshold</i>] [restart <i>time</i> warming-only] Example : <pre>switch(config-router-neighbor-af)# maximum-prefix 12</pre>	Configures the maximum number of prefixes from a peer. The parameter ranges are as follows: <ul style="list-style-type: none"> • <i>maximum</i> —The range is from 1 to 300000 • <i>Threshold</i> —The range is from 1 to 100 percent. The default is 75 percent. • <i>time</i> —The range is from 1 to 65535 minutes. This command triggers an automatic notification and session reset for the BGP neighbor sessions if the prefix limit is exceeded.

Configuring Dynamic Capability

You can configure dynamic capability for a BGP peer.

To configure dynamic capability, use the following command in neighbor configuration mode:

Command	Purpose
dynamic-capability Example : <pre>switch(config-router-neighbor)# dynamic-capability</pre>	Enables dynamic capability. This command triggers an automatic notification and session reset for the BGP neighbor sessions. This command is disabled by default.

Configuring Aggregate Addresses

You can configure aggregate address entries in the BGP route table.

To configure an aggregate address, use the following command in router address-family configuration mode:

Command	Purpose
<p>aggregate-address <i>ip-prefix/length</i> [as-set] [summary-only] [advertise-map <i>map-name</i>] [attribute-map <i>map-name</i>] [suppress-map <i>map-name</i>]</p> <p>Example :</p> <pre>switch(config-router-af)# aggregate-address 192.0.2.0/8 as-set</pre>	<p>Creates an aggregate address. The path advertised for this route is an autonomous system set that consists of all elements contained in all paths that are being summarized:</p> <ul style="list-style-type: none"> • The as-set keyword generates autonomous system set path information and community information from contributing paths. • The summary-only keyword filters all more specific routes from updates. • The advertise-map keyword and argument specify the route map used to select attribute information from selected routes. • The attribute-map keyword and argument specify the route map used to select attribute information from the aggregate. • The suppress-map keyword and argument conditionally filters more specific routes.

Configuring BGP Conditional Advertisement

You can configure BGP conditional advertisement to limit the routes that BGP propagates. You define the following two route maps:

- **Advertise map**—Specifies the conditions that the route must match before BGP considers the conditional advertisement. This route map can contain any appropriate match statements.
- **Exist map or nonexist map**—Defines the prefix that must exist in the BGP table before BGP propagates a route that matches the advertise map. The nonexist map defines the prefix that must not exist in the BGP table before BGP propagates a route that matches the advertise map. BGP processes only the permit statements in the prefix list match statements in these route maps.

If the route does not pass the condition, BGP withdraws the route if it exists in the BGP table.

Before you begin

Ensure that you have enabled the BGP feature (see the [Enabling the BGP Feature](#) section).

SUMMARY STEPS

1. **configure terminal**
2. **router bgp** *as-number*
3. **neighbor** *ip-address* **remote-as** *as-number*
4. **address-family ipv4 unicast**
5. **advertise-map** *adv-map* { **exist-map** *exist-rmap* | **non-exist-map** *nonexist-rmap* }
6. (Optional) **show ip bgp neighbor**
7. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure terminal Example: <pre>switch# configure terminal switch(config)#</pre>	Enters configuration mode.
Step 2	router bgp <i>as-number</i> Example: <pre>switch(config)# router bgp 65536 switch(config-router)#</pre>	Enters BGP mode and assigns the autonomous system number to the local BGP speaker.
Step 3	neighbor <i>ip-address</i> remote-as <i>as-number</i> Example: <pre>switch(config-router)# neighbor 192.168.1.2 remote-as 65537 switch(config-router-neighbor)#</pre>	Places the router in neighbor configuration mode for BGP routing and configures the neighbor IP address.
Step 4	address-family ipv4 unicast Example: <pre>switch(config-router-neighbor)# address-family ipv4 unicast switch(config-router-neighbor-af)#</pre>	Enters address family configuration mode.
Step 5	advertise-map <i>adv-map</i> { exist-map <i>exist-rmap</i> non-exist-map <i>nonexist-rmap</i> } Example: <pre>switch(config-router-neighbor-af)# advertise-map advertise exist-map exist</pre>	Configures BGP to conditionally advertise routes based on the two configured route maps: <ul style="list-style-type: none"> • <i>adv-map</i>—Specifies a route map with match statements that the route must pass before BGP passes the route to the next route map. The <i>adv-map</i> is a case-sensitive, alphanumeric string up to 63 characters. • <i>exist-rmap</i>—Specifies a route map with match statements for a prefix list. A prefix in the BGP table must match a prefix in the prefix list before BGP will advertise the route. The <i>exist-rmap</i> is a case-sensitive, alphanumeric string up to 63 characters. • <i>nonexist-rmap</i>—Specifies a route map with match statements for a prefix list. A prefix in the BGP table must not match a prefix in the prefix list before BGP will advertise the route. The <i>nonexist-rmap</i> is a case-sensitive, alphanumeric string up to 63 characters.
Step 6	(Optional) show ip bgp neighbor Example: <pre>switch(config-router-neighbor-af)# show ip bgp neighbor</pre>	Displays information about BGP and the configured conditional advertisement route maps.

	Command or Action	Purpose
Step 7	(Optional) copy running-config startup-config Example: switch(config-router-neighbor)# copy running-config startup-config	Saves this configuration change.

Example

This example shows how to configure BGP conditional advertisement:

```
switch# configure terminal
switch(config)# router bgp 65536
switch(config-router)# neighbor 192.0.2.2 remote-as 65537
switch(config-router-neighbor)# address-family ipv4 unicast
switch(config-router-neighbor-af)# advertise-map advertise exist-map exist
switch(config-router-neighbor-af)# exit
switch(config-router-neighbor)# exit
switch(config-router)# exit
switch(config)# route-map advertise
switch(config-route-map)# match as-path pathList
switch(config-route-map)# exit
switch(config)# route-map exit
switch(config-route-map)# match ip address prefix-list plist
switch(config-route-map)# exit
switch(config)# ip prefix-list plist permit 209.165.201.0/27
```

Configuring Route Redistribution

You can configure BGP to accept routing information from another routing protocol and redistribute that information through the BGP network. Optionally, you can assign a default route for redistributed routes.

Before you begin

Ensure that you have enabled the BGP feature (see the [Enabling the BGP Feature](#) section).

SUMMARY STEPS

1. **configure terminal**
2. **router bgp** *as-number*
3. **address-family ipv4 unicast**
4. **redistribute** { **direct** | { **eigrp** | **ospf** | **ospfv3** | **rip** } *instance-tag* | **static** } **route-map** *map-name*
5. (Optional) **default-metric** *value*
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

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

	Command or Action	Purpose
	switch# configure terminal switch(config)#	
Step 2	router bgp <i>as-number</i> Example: switch(config)# router bgp 65536 switch(config-router)#	Enters BGP mode and assigns the autonomous system number to the local BGP speaker.
Step 3	address-family ipv4 unicast Example: switch(config-router-neighbor)# address-family ipv4 unicast switch(config-router-neighbor-af)#	Enters address family configuration mode.
Step 4	redistribute { direct { eigrp ospf ospfv3 rip } instance-tag static } route-map <i>map-name</i> Example: switch(config-router-af)# redistribute eigrp 201 route-map Eigrpmap	Redistributes routes from other protocols into BGP. See the Configuring Route Maps section for more information about route maps.
Step 5	(Optional) default-metric <i>value</i> Example: switch(config-router-af)# default-metric 33	Generates a default route into BGP.
Step 6	(Optional) copy running-config startup-config Example: switch(config-router-neighbor)# copy running-config startup-config	Saves this configuration change.

Example

This example shows how to redistribute EIGRP into BGP:

```
switch# configure terminal
switch(config)# router bgp 65536
switch(config-router)# address-family ipv4 unicast
switch(config-router-af)# redistribute eigrp 201 route-map Eigrpmap
switch(config-router-af)# copy running-config startup-config
```

Configuring Multiprotocol BGP

You can configure MP-BGP to support multiple address families, including IPv4unicast routes.

Before you begin

Ensure that you have enabled the BGP feature (see the [Enabling the BGP Feature](#) section).

SUMMARY STEPS

1. **configure terminal**
2. **router bgp *as-number***
3. **neighbor *ip-address* remote-as *as-number***
4. **address-family ipv4 unicast**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure terminal Example: <pre>switch# configure terminal switch(config)#</pre>	Enters configuration mode.
Step 2	router bgp <i>as-number</i> Example: <pre>switch(config)# router bgp 65536 switch(config-router)#</pre>	Enters BGP mode and assigns the autonomous system number to the local BGP speaker.
Step 3	neighbor <i>ip-address</i> remote-as <i>as-number</i> Example: <pre>switch(config-router)# neighbor 192.168.1.2 remote-as 65537 switch(config-router-neighbor)#</pre>	Places the router in neighbor configuration mode for BGP routing and configures the neighbor IP address.
Step 4	address-family ipv4 unicast Example: <pre>switch(config-router-neighbor)# address-family ipv4 unicast switch(config-router-neighbor-af)#</pre>	Enters address family configuration mode.
Step 5	(Optional) copy running-config startup-config Example: <pre>switch(config-router-neighbor)# copy running-config startup-config</pre>	Saves this configuration change.

Tuning BGP

You can tune BGP characteristics through a series of optional parameters.

To tune BGP, use the following optional commands in router configuration mode:

Command	Purpose
<p>bestpath [always-compare-med compare-routerid igp-metric ignore med { missing-as-worst non-deterministic } as-path multipath-relax]</p> <p>Example :</p> <pre>switch(config-router)# bestpath always-compare-med switch(config-router)# bestpath as-path multipath-relax</pre>	<p>Modifies the best-path algorithm. The optional parameters are as follows:</p> <ul style="list-style-type: none"> • always-compare-med—Compares MED on paths from different autonomous systems. • compare-routerid—Compares the router IDs for identical eBGP paths. • igp-metric ignore—Ignores the Interior Gateway Protocol (IGP) metric for next hop during best-path selection. This option is supported beginning with Cisco NX-OS Release 9.2(2). • med missing-as-worst—Handles a missing MED as the highest MED. • med non-deterministic—Does not always select the best MED path from among the paths from the same autonomous systems. • as-path multipath-relax—Allows the switch to handle the paths received from different autonomous systems for multipath, if their AS-path lengths are the same and other multipath conditions are met.
<p>enforce-first-as</p> <p>Example:</p> <pre>switch(config-router)# enforce-first-as</pre>	<p>Enforces the neighbor autonomous system to be the first AS number listed in the AS_path attribute for eBGP.</p>
<p>log-neighbor-changes</p> <p>Example :</p> <pre>switch(config-router)# log-neighbor-changes</pre>	<p>Generates a system message when a neighbor changes state.</p>
<p>router-id <i>id</i></p> <p>Example :</p> <pre>switch(config-router)# router-id 209.165.20.1</pre>	<p>Manually configures the router ID for this BGP speaker.</p>
<p>timers [bestpath-delay <i>delay</i> bgp keepalive holdtime prefix-peer-timeout timeout]</p> <p>Example :</p> <pre>switch(config-router)# timers bgp 90 270</pre>	<p>Sets the BGP timer values. The optional parameters are as follows:</p> <p>You must manually reset the BGP sessions after configuring this command.</p>

To tune BGP, use the following optional command in router address-family configuration mode:

Command	Purpose
distance <i>ebgp-distance ibgp distance local-distance</i> Example : <pre>switch(config-router-af)# distance 20 100 200</pre>	Sets the administrative distance for BGP. The range is from 1 to 255. The defaults are as follows:

To tune BGP, use the following optional commands in neighbor configuration mode:

Command	Purpose
description <i>string</i> Example : <pre>switch(config-router-neighbor)# description main site</pre>	Sets a descriptive string for this BGP peer. The string can be up to 80 alphanumeric characters.
low-memory exempt Example : <pre>switch(config-router-neighbor)# low-memory exempt</pre>	Exempts this BGP neighbor from a possible shutdown due to a low memory condition.
transport connection-mode passive Example: <pre>switch(config-router-neighbor)# transport connection-mode passive</pre>	Allows a passive connection setup only. This BGP speaker does not initiate a TCP connection to a BGP peer. You must manually reset the BGP sessions after configuring this command.
remove-private-as Example : <pre>switch(config-router-neighbor)# remove-private-as</pre>	Removes private AS numbers from outbound route updates to an eBGP peer. This command triggers an automatic soft clear or refresh of BGP neighbor sessions.
update-source <i>interface-type number</i> Example : <pre>switch(config-router-neighbor)# update-source ethernet 2/1</pre>	Configures the BGP speaker to use the source IP address of the configured interface for BGP sessions to the peer. This command triggers an automatic notification and session reset for the BGP neighbor sessions.

To tune BGP, use the following optional commands in neighbor address-family configuration mode:

Command	Purpose
suppress-inactive Example : <pre>switch(config-router-neighbor-af)# suppress-inactive</pre>	Advertises the best (active) routes only to the BGP peer. This command triggers an automatic soft clear or refresh of BGP neighbor sessions.
default-originate [route-map <i>map-name</i>] Example : <pre>switch(config-router-neighbor-af)# default-originate</pre>	Generates a default route to the BGP peer.

Command	Purpose
filter-list <i>list-name</i> { in out } Example: <pre>switch(config-router-neighbor-af) # filter-list BGPFilter in</pre>	Applies an AS_path filter list to this BGP peer for inbound or outbound route updates. This command triggers an automatic soft clear or refresh of BGP neighbor sessions.
prefix-list <i>list-name</i> { in out } Example: <pre>switch(config-router-neighbor-af) # prefix-list PrefixFilter in</pre>	Applies a prefix list to this BGP peer for inbound or outbound route updates. This command triggers an automatic soft clear or refresh of BGP neighbor sessions.
send-community Example : <pre>switch(config-router-neighbor-af) # send-community</pre>	Sends the community attribute to this BGP peer. This command triggers an automatic soft clear or refresh of BGP neighbor sessions.
send-extended-community Example : <pre>switch(config-router-neighbor-af) # send-extended-community</pre>	Sends the extended community attribute to this BGP peer. This command triggers an automatic soft clear or refresh of BGP neighbor sessions.

Configuring Virtualization

Before you begin

Ensure that you have enabled the BGP feature (see the Enabling the BGP Feature section).

SUMMARY STEPS

1. **configure terminal**
2. **vrf context** *vrf-name*
3. **exit**
4. **router bgp** *as-number*
5. **vrf** *vrf-name*
6. **neighbor** *ip-address* **remote-as** *as-number*
7. (Optional) **bestpath as-path multipath-relax**
8. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure terminal Example: <pre>switch# configure terminal switch(config)#</pre>	Enters configuration mode.

	Command or Action	Purpose
Step 2	vrf context <i>vrf-name</i> Example: <pre>switch(config)# vrf context RemoteOfficeVRF switch(config-vrf)#</pre>	Creates a new VRF and enters VRF configuration mode.
Step 3	exit Example: <pre>switch(config-vrf)# exit switch(config)#</pre>	Exits VRF configuration mode.
Step 4	router bgp <i>as-number</i> Example: <pre>switch(config)# router bgp 65536 switch(config-router)#</pre>	Creates a new BGP process with the configured autonomous system number.
Step 5	vrf <i>vrf-name</i> Example: <pre>switch(config-router)# vrf RemoteOfficeVRF switch(config-router-vrf)#</pre>	Enters the router VRF configuration mode and associates this BGP instance with a VRF.
Step 6	neighbor <i>ip-address</i> remote-as <i>as-number</i> Example: <pre>switch(config-router-vrf)# neighbor 209.165.201.1 remote-as 65536 switch(config-router--vrf-neighbor)#</pre>	Configures the IP address and AS number for a remote BGP peer.
Step 7	(Optional) bestpath as-path multipath-relax Example: <pre>switch(config-router-vrf)# bestpath as-path multipath-relax</pre>	Allows the switch to treat paths received from different autonomous systems for multipath, if their autonomous path lengths are the same and other multipath conditions are met.
Step 8	(Optional) copy running-config startup-config Example: <pre>switch(config-router-neighbor)# copy running-config startup-config</pre>	Saves this configuration change.

Example

This example shows how to create a VRF and configure the router ID in the VRF:

```
switch# configure terminal
switch(config)# vrf context NewVRF
switch(config-vrf)# exit
switch(config)# router bgp 65536
switch(config-router)# vrf NewVRF
switch(config-router-vrf)# neighbor 209.165.201.1 remote-as 65536
switch(config-router-vrf-neighbor)# copy running-config startup-config
```

Verifying the Advanced BGP Configuration

To display the BGP configuration information, perform one of the following tasks:

Command	Purpose
show bgp all [summary] [vrf vrf-name]	Displays the BGP information for all address families.
show bgp convergence [vrf vrf-name]	Displays the BGP information for all address families.
show bgp ip {unicast} [<i>ip-address</i>] community { regexp expression [community] [no-advertise] [no-export] [no-export-subconfed] } [vrf vrf-name]	Displays the BGP routes that match a BGP community.
show bgp [vrf vrf-name] ip {unicast} [<i>ip-address</i>] community-list list-name [vrf vrf-name]	Displays the BGP routes that match a BGP community list.
show bgp ip {unicast} [<i>ip-address</i>] extcommunity { regexp expression generic [non-transitive transitive] <i>aa4:nn</i> [exact-match] } [vrf vrf-name]	Displays the BGP routes that match a BGP extended community.
show bgp ip {unicast} [<i>ip-address</i>] extcommunity-list list-name [exact-match] [vrf vrf-name]	Displays the BGP routes that match a BGP extended community list.
show bgp ip {unicast} [<i>ip-address</i>] { dampening dampened-paths [regexp expression] } [vrf vrf-name]	Displays the information for BGP route dampening. Use the clear bgp dampening command to clear the route flap dampening information.
show bgp ip {unicast} [<i>ip-address</i>] history-paths [regexp expression] [vrf vrf-name]	Displays the BGP route history paths.
show bgp ip {unicast} [<i>ip-address</i>] filter-list list-name [vrf vrf-name]	Displays the information for the BGP filter list.
show bgp ip {unicast} [<i>ip-address</i>] neighbors [<i>ip-address</i>] [vrf vrf-name]	Displays the information for BGP peers. Use the clear bgp neighbors command to clear these neighbors.
show bgp ip {unicast} [<i>ip-address</i>] { nexthop nexthop-database } [vrf vrf-name]	Displays the information for the BGP route next hop.
show bgp paths	Displays the BGP path information.
show bgp ip {unicast} [<i>ip-address</i>] policy name [vrf vrf-name]	Displays the BGP policy information. Use the clear bgp policy command to clear the policy information.
show bgp ip {unicast} [<i>ip-address</i>] prefix-list list-name [vrf vrf-name]	Displays the BGP routes that match the prefix list.
show bgp ip {unicast} [<i>ip-address</i>] received-paths [vrf vrf-name]	Displays the BGP paths stored for soft reconfiguration.

Command	Purpose
show bgp ip {unicast} [<i>ip-address</i>] regex <i>expression</i> [vrf <i>vrf-name</i>]	Displays the BGP routes that match the AS_path regular expression.
show bgp ip {unicast} [<i>ip-address</i>] route-map <i>map-name</i> [vrf <i>vrf-name</i>]	Displays the BGP routes that match the route map.
show bgp peer-policy <i>name</i> [vrf <i>vrf-name</i>]	Displays the information about BGP peer policies.
show bgp peer-session <i>name</i> [vrf <i>vrf-name</i>]	Displays the information about BGP peer sessions.
show bgp peer-template <i>name</i> [vrf <i>vrf-name</i>]	Displays the information about BGP peer templates. Use the clear bgp peer-template command to clear all neighbors in a peer template.
show bgp process	Displays the BGP process information.
show ip bgp <i>options</i>	Displays the BGP status and configuration information. This command has multiple options. See the Cisco Nexus 3000 Series Command Reference for more information.
show ip mbgp <i>options</i>	Displays the BGP status and configuration information. This command has multiple options. See the Cisco Nexus 3000 Series Command Reference for more information.
show running-configuration bgp	Displays the current running BGP configuration.

Displaying BGP Statistics

To display BGP statistics, use the following commands:

Command	Purpose
show bgp ip {unicast} [<i>ip-address</i>] flap-statistics [vrf <i>vrf-name</i>]	Displays the BGP route flap statistics. Use the clear bgp flap-statistics command to clear these statistics.
show bgp sessions [vrf <i>vrf-name</i>]	Displays the BGP sessions for all peers. Use the clear bgp sessions command to clear these statistics.
show bgp sessions [vrf <i>vrf-name</i>]	Displays the BGP sessions for all peers. Use the clear bgp sessions command to clear these statistics.
show bgp statistics	Displays the BGP statistics.

Related Topics

The following topics can give more information on BGP:

- [Configuring Basic BGP](#)
- [Configuring Route Policy Manager](#)

Additional References

For additional information related to implementing BGP, see the following sections:

- [Related Documents](#)
- [MIBs](#)

Related Documents

Related Topic	Document Title
BGP CLI commands	Cisco Nexus 3000 Series Command Reference

MIBs

MIBs	MIBs Link
BGP4-MIB CISCO-BGP4-MIB	To locate and download MIBs, go to the following: MIB Locator .