Configuring Basic BGP

This chapter describes how to configure Border Gateway Protocol (BGP) on the Cisco NX-OS device.

This chapter includes the following sections:

About Basic BGP

Cisco NX-OS supports BGP version 4, which includes multiprotocol extensions that allow BGP to carry routing information for IP multicast routes and multiple Layer 3 protocol address families. BGP uses TCP as a reliable transport protocol to create TCP sessions with other BGP-enabled devices.

BGP uses a path-vector routing algorithm to exchange routing information between BGP-enabled networking devices or BGP speakers. Based on this information, each BGP speaker determines a path to reach a particular destination while detecting and avoiding paths with routing loops. The routing information includes the actual route prefix for a destination, the path of autonomous systems to the destination, and additional path attributes.

BGP selects a single path, by default, as the best path to a destination host or network. Each path carries well-known mandatory, well-known discretionary, and optional transitive attributes that are used in BGP best-path analysis. You can influence BGP path selection by altering some of these attributes by configuring BGP policies. See the "Route Policies and Resetting BGP Sessions" section on page 10-3 for more information.

BGP also supports load balancing or equal-cost multipath (ECMP). See the "Load Sharing and Multipath" section on page 10-6 for more information.

This section includes the following topics:

BGP Autonomous Systems

An autonomous system (AS) is a network controlled by a single administration entity. An autonomous system forms a routing domain with one or more interior gateway protocols (IGPs) and a consistent set of routing policies. BGP supports 16-bit and 32-bit autonomous system numbers. For more information, see the "Autonomous Systems" section on page 1-5.

Separate BGP autonomous systems dynamically exchange routing information through external BGP (eBGP) peering sessions. BGP speakers within the same autonomous system can exchange routing information through internal BGP (iBGP) peering sessions.

4-Byte AS Number Support

BGP supports 2-byte autonomous system (AS) numbers in plain-text notation or as.dot notation and 4-byte AS numbers in plain-text notation.

Administrative Distance

An administrative distance is a rating of the trustworthiness of a routing information source. By default, BGP uses the administrative distances shown in Table 9-1.

Table 9-1 BGP Default Administrative Distances

Distance	Default Value	Function
External	20	Applied to routes learned from eBGP.
Internal	200	Applied to routes learned from iBGP.
Local	220	Applied to routes originated by the router.

Note The administrative distance does not influence the BGP path selection algorithm, but it does influence whether BGP-learned routes are installed in the IP routing table.

For more information, see the "Administrative Distance" section on page 1-7.

BGP Peers

A BGP speaker does not discover another BGP speaker automatically. You must configure the relationships between BGP speakers. A BGP peer is a BGP speaker that has an active TCP connection to another BGP speaker.

BGP Sessions

BGP uses TCP port 179 to create a TCP session with a peer. When a TCP connection is established between peers, each BGP peer initially exchanges all of its routes—the complete BGP routing table—with the other peer. After this initial exchange, the BGP peers send only incremental updates when a topology change occurs in the network or when a routing policy change occurs. In the periods of inactivity between these updates, peers exchange special messages called keepalives. The hold time is the maximum time limit that can elapse between receiving consecutive BGP update or keepalive messages.

Cisco NX-OS supports the following peer configuration options:

- Individual IPv4 or IPv4 address—BGP establishes a session with the BGP speaker that matches the remote address and AS number.
- IPv4 or IPv6 prefix peers for a single AS number—BGP establishes sessions with BGP speakers that match the prefix and the AS number.
- Dynamic AS number prefix peers—BGP establishes sessions with BGP speakers that match the prefix and an AS number from a list of configured AS numbers.

Dynamic AS Numbers for Prefix Peers

Cisco NX-OS accepts a range or list of AS numbers to establish BGP sessions. For example, if you configure BGP to use IPv4 prefix 192.0.2.0/8 and AS numbers 33, 66, and 99, BGP establishes a session with 192.0.2.1 with AS number 66 but rejects a session from 192.0.2.2 with AS number 50.

Cisco NX-OS does not associate prefix peers with dynamic AS numbers as either interior BGP (iBGP) or external BGP (eBGP) sessions until after the session is established. See Chapter 10, "Configuring Advanced BGP" for more information on iBGP and eBGP.

Note The dynamic AS number prefix peer configuration overrides the individual AS number configuration that is inherited from a BGP template. For more information, see Chapter 10, "Configuring Advanced BGP."

BGP Router Identifier

To establish BGP sessions between peers, BGP must have a router ID, which is sent to BGP peers in the OPEN message when a BGP session is established. The BGP router ID is a 32-bit value that is often represented by an IPv4 address. You can configure the router ID. By default, Cisco NX-OS sets the router ID to the IPv4 address of a loopback interface on the router. If no loopback interface is configured on the router, the software chooses the highest IPv4 address configured to a physical interface on the router to represent the BGP router ID. The BGP router ID must be unique to the BGP peers in a network.

If BGP does not have a router ID, it cannot establish any peering sessions with BGP peers.

BGP Path Selection

BGP supports sending and receiving multiple paths per prefix and advertising such paths. For information on configuring additional BGP paths, see Chapter 10, "Configuring Advanced BGP."

The best-path algorithm runs each time that a path is added or withdrawn for a given network. The best-path algorithm also runs if you change the BGP configuration. BGP selects the best path from the set of valid paths available for a given network.

Cisco NX-OS implements the BGP best-path algorithm in the following steps:

Step 1 Compares two paths to determine which is better (see the "Step 1—Comparing Pairs of Paths" section on page 9-4).

Step 2 Explores all paths and determines in which order to compare the paths to select the overall best path (see the "Step 2—Determining the Order of Comparisons" section on page 9-6).

Step 3 Determines whether the old and new best paths differ enough so that the new best path should be used (see the "Step 3—Determining the Best-Path Change Suppression" section on page 9-6).

Note The order of comparison determined in Step 2 is important. Consider the case where you have three paths, A, B, and C. When Cisco NX-OS compares A and B, it chooses A. When Cisco NX-OS compares B and C, it chooses B. But when Cisco NX-OS compares A and C, it might not choose A because some BGP metrics apply only among paths from the same neighboring autonomous system and not among all paths.

The path selection uses the BGP AS-path attribute. The AS-path attribute includes the list of autonomous system numbers (AS numbers) traversed in the advertised path. If you subdivide your BGP autonomous system into a collection or confederation of autonomous systems, the AS-path contains confederation segments that list these locally defined autonomous systems.

Step 1—Comparing Pairs of Paths

This first step in the BGP best-path algorithm compares two paths to determine which path is better. The following sequence describes the basic steps that Cisco NX-OS uses to compare two paths to determine the better path:

1. Cisco NX-OS chooses a valid path for comparison. (For example, a path that has an unreachable next hop is not valid.)
2. Cisco NX-OS chooses the path with the highest weight.
3. Cisco NX-OS chooses the path with the highest local preference.
4. If one of the paths is locally originated, Cisco NX-OS chooses that path.
5. Cisco NX-OS chooses the path with the shorter AS-path.

Note When calculating the length of the AS-path, Cisco NX-OS ignores confederation segments and counts AS sets as 1. See the "AS Confederations" section on page 10-4 for more information.

6. Cisco NX-OS chooses the path with the lower origin. The Interior Gateway Protocol (IGP) is considered lower than EGP.
7. Cisco NX-OS chooses the path with the lower multiexit discriminator (MED).

   You can configure a number of options that affect whether or not this step is performed. In general, Cisco NX-OS compares the MED of both paths if the paths were received from peers in the same autonomous system; otherwise, Cisco NX-OS skips the MED comparison.

   You can configure Cisco NX-OS to always perform the best-path algorithm MED comparison, regardless of the peer autonomous system in the paths. See the "Tuning the Best-Path Algorithm" section on page 10-10 for more information. Otherwise, Cisco NX-OS performs a MED comparison that depends on the AS-path attributes of the two paths being compared:

   a. If a path has no AS-path or the AS-path starts with an AS_SET, the path is internal and Cisco NX-OS compares the MED to other internal paths.
   b. If the AS-path starts with an AS_SEQUENCE, the peer autonomous system is the first AS number in the sequence and Cisco NX-OS compares the MED to other paths that have the same peer autonomous system.
   c. If the AS-path contains only confederation segments or starts with confederation segments followed by an AS_SET, the path is internal and Cisco NX-OS compares the MED to other internal paths.
   d. If the AS-path starts with confederation segments that are followed by an AS_SEQUENCE, the peer autonomous system is the first AS number in the AS_SEQUENCE and Cisco NX-OS compares the MED to other paths that have the same peer autonomous system.

   Note If Cisco NX-OS receives no MED attribute with the path, Cisco NX-OS considers the MED to be 0 unless you configure the best-path algorithm to set a missing MED to the highest possible value. See the "Tuning the Best-Path Algorithm" section on page 10-10 for more information.

   e. If the nondeterministic MED comparison feature is enabled, the best-path algorithm uses the Cisco IOS style of MED comparison. See the "Tuning the Best-Path Algorithm" section on page 10-10 for more information.
8. If one path is from an internal peer and the other path is from an external peer, Cisco NX-OS chooses the path from the external peer.
9. If the paths have different IGP metrics to their next-hop addresses, Cisco NX-OS chooses the path with the lower IGP metric.
10. Cisco NX-OS uses the path that was selected by the best-path algorithm the last time that it was run.

If all path parameters in Step 1 through Step 9 are the same, you can configure the best-path algorithm to compare the router IDs. See the "Tuning the Best-Path Algorithm" section on page 10-10 for more information. If the path includes an originator attribute, Cisco NX-OS uses that attribute as the router ID to compare to; otherwise, Cisco NX-OS uses the router ID of the peer that sent the path. If the paths have different router IDs, Cisco NX-OS chooses the path with the lower router ID.

Note When using the attribute originator as the router ID, it is possible that two paths have the same router ID. It is also possible to have two BGP sessions with the same peer router, so you could receive two paths with the same router ID.

11. Cisco NX-OS selects the path with the shorter cluster length. If a path was not received with a cluster list attribute, the cluster length is 0.
12. Cisco NX-OS chooses the path received from the peer with the lower IP address. Locally generated paths (for example, redistributed paths) have a peer IP address of 0.

Note Paths that are equal after Step 9 can be used for multipath if you configure multipath. See the "Load Sharing and Multipath" section on page 10-6 for more information.

Step 2—Determining the Order of Comparisons

The second step of the BGP best-path algorithm implementation is to determine the order in which Cisco NX-OS compares the paths:

1. Cisco NX-OS partitions the paths into groups. Within each group, Cisco NX-OS compares the MED among all paths. Cisco NX-OS uses the same rules as in the "Step 1—Comparing Pairs of Paths" section on page 9-4 to determine whether MED can be compared between any two paths. Typically, this comparison results in one group being chosen for each neighbor autonomous system. If you configure the **bgp bestpath med always** command, Cisco NX-OS chooses just one group that contains all the paths.
2. Cisco NX-OS determines the best path in each group by iterating through all paths in the group and keeping track of the best one so far. Cisco NX-OS compares each path with the temporary best path found so far and if the new path is better, it becomes the new temporary best path and Cisco NX-OS compares it with the next path in the group.
3. Cisco NX-OS forms a set of paths that contain the best path selected from each group in Step 2. Cisco NX-OS selects the overall best path from this set of paths by going through them as in Step 2.

Step 3—Determining the Best-Path Change Suppression

The next part of the implementation is to determine whether Cisco NX-OS uses the new best path or suppresses the new best path. The router can continue to use the existing best path if the new one is identical to the old path (if the router ID is the same). Cisco NX-OS continues to use the existing best path to avoid route changes in the network.

You can turn off the suppression feature by configuring the best-path algorithm to compare the router IDs. See the "Tuning the Best-Path Algorithm" section on page 10-10 for more information. If you configure this feature, the new best path is always preferred to the existing one.

You cannot suppress the best-path change if any of the following conditions occur:

- The existing best path is no longer valid.
- Either the existing or new best paths were received from internal (or confederation) peers or were locally generated (for example, by redistribution).
- The paths were received from the same peer (the paths have the same router ID).
- The paths have different weights, local preferences, origins, or IGP metrics to their next-hop addresses.
- The paths have different MEDs.

BGP and the Unicast RIB

BGP communicates with the unicast Routing Information Base (unicast RIB) to store IPv4 routes in the unicast routing table. After selecting the best path, if BGP determines that the best path change needs to be reflected in the routing table, it sends a route update to the unicast RIB.

BGP receives route notifications regarding changes to its routes in the unicast RIB. It also receives route notifications about other protocol routes to support redistribution.

BGP also receives notifications from the unicast RIB regarding next-hop changes. BGP uses these notifications to keep track of the reachability and IGP metric to the next-hop addresses.

Whenever the next-hop reachability or IGP metrics in the unicast RIB change, BGP triggers a best-path recalculation for affected routes.

BGP communicates with the IPv6 unicast RIB to perform these operations for IPv6 routes.

BGP Prefix Independent Convergence Core

The BGP prefix independent convergence (PIC) core feature allows for faster convergence for traffic destined to BGP prefixes that share the same remote next hop in case of a failure in the core of the network. Pure IP traffic can benefit from this feature. It is enabled by default and cannot be disabled.

BGP Virtualization

BGP supports virtual routing and forwarding (VRF) instances.

Licensing Requirements for Basic BGP

The following table shows the licensing requirements for this feature:

Product	License Requirement
Cisco NX-OS	BGP requires an Enterprise Services license. For a complete explanation of the Cisco NX-OS licensing scheme and how to obtain and apply licenses, see the *Cisco NX-OS Licensing Guide.*

Prerequisites for BGP

BGP has the following prerequisites:

- You must enable BGP (see the "Enabling BGP" section on page 9-11).
- 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 configure at least one IGP that is capable of recursive next-hop resolution.
- You must configure an address family under a neighbor for the BGP session establishment.

Guidelines and Limitations for BGP

BGP has the following configuration guidelines and limitations:

- The dynamic AS number prefix peer configuration overrides the individual AS number configuration inherited from a BGP template.
- If you configure a dynamic AS number for prefix peers in an AS confederation, BGP establishes sessions with only the AS numbers in the local confederation.
- BGP sessions created through a dynamic AS 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 received and system resources used.
- Configure the update source to establish a session with BGP/eBGP multihop sessions.
- Specify a BGP policy if you configure redistribution.
- Define the BGP router ID within a VRF.
- For IPv6 neighbors, Cisco recommends that you configure a router ID per VRF. If a VRF does not have any IPv4 interfaces, the IPv6 BGP neighbor will not come up because its router ID must be an IPv4 address. The numerically lowest loopback IPv4 address is elected to be the router ID. If a loopback address does not exist, the lowest IP address from the VRF interfaces is elected. If that does not exist, the BGP neighbor relationship is not established.
- If you decrease the keepalive and hold timer values, you might experience BGP session flaps.
- The BGP minimum route advertisement interval (MRAI) value for all iBGP and eBGP sessions is zero and is not configurable.
- Although the **show ip bgp** commands are available for verifying the BGP configuration, Cisco recommends using the **show bgp** commands instead.

Default Settings

Table 9-2 lists the default settings for BGP parameters.

Table 9-2 Default BGP Parameters

Parameters	Default
BGP feature	Disabled
Keep alive interval	60 seconds
Hold timer	180 seconds
BGP PIC core	Enabled
Auto-summary	Always disabled
Synchronization	Always disabled

CLI Configuration Modes

The following sections describe how to enter each of the CLI configuration modes for BGP. From a mode, you can enter the **?** command to display the commands available in that mode.

Global Configuration Mode

Use global configuration mode to create a BGP process and configure advanced features such as AS confederation and route dampening. For more information, see Chapter 10, "Configuring Advanced BGP."

This example shows how to enter router configuration mode:

```
switch# configuration
switch(config)# router bgp 64496
switch(config-router)#
```

BGP supports VRF. You can configure BGP within the appropriate VRF if you are using VRFs in your network. See the "Configuring Virtualization" section on page 10-50 for more information.

This example shows how to enter VRF configuration mode:

```
switch(config)# router bgp 64497
switch(config-router)# vrf vrf_A
switch(config-router-vrf)#
```

Address Family Configuration Mode

You can optionally configure the address families that BGP supports. Use the **address-family** command in router configuration mode to configure features for an address family. Use the **address-family** command in neighbor configuration mode to configure the specific address family for the neighbor.

You must configure the address families if you are using route redistribution, address aggregation, load balancing, and other advanced features.

This example shows how to enter address family configuration mode from the router configuration mode:

```
switch(config)# router bgp 64496
switch(config-router)# address-family ipv6 unicast
switch(config-router-af)#
```

This example shows how to enter VRF address family configuration mode if you are using VRFs:

```
switch(config)# router bgp 64497
switch(config-router)# vrf vrf_A
switch(config-router-vrf)# address-family ipv6 unicast
switch(config-router-vrf-af)#
```

Neighbor Configuration Mode

Cisco NX-OS provides the neighbor configuration mode to configure BGP peers. You can use neighbor configuration mode to configure all parameters for a peer.

This example shows how to enter neighbor configuration mode:

```
switch(config)# router bgp 64496
switch(config-router)# neighbor 192.0.2.1
switch(config-router-neighbor)#
```

This example shows how to enter VRF neighbor configuration mode:

```
switch(config)# router bgp 64497
switch(config-router)# vrf vrf_A
switch(config-router-vrf)# neighbor 192.0.2.1
switch(config-router-vrf-neighbor)#
```

Neighbor Address Family Configuration Mode

An address family configuration submode inside the neighbor configuration submode is available for entering address family-specific neighbor configuration and enabling the address family for the neighbor. Use this mode for advanced features such as limiting the number of prefixes allowed for this neighbor and removing private AS numbers for eBGP.

This example shows how to enter neighbor address family configuration mode:

```
switch(config)# router bgp 64496
switch(config-router# neighbor 192.0.2.1
switch(config-router-neighbor)# address-family ipv4 unicast
switch(config-router-neighbor-af)#
```

This example shows how to enter VRF neighbor address family configuration mode:

```
switch(config)# router bgp 64497
switch(config-router)# vrf vrf_A
switch(config-router-vrf)# neighbor 209.165.201.1
switch(config-router-vrf-neighbor)# address-family ipv4 unicast
switch(config-router-vrf-neighbor-af)#
```

Configuring Basic BGP

To configure a basic BGP, you must enable BGP and configure a BGP peer. Configuring a basic BGP network consists of a few required tasks and many optional tasks. You must configure a BGP routing process and BGP peers.

This section includes the following topics:

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.

Enabling BGP

You must enable BGP before you can configure BGP.

SUMMARY STEPS

1. **configure terminal**
2. **feature bgp**
3. (Optional) **show feature**
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

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

Step 2	**feature bgp** **Example:** `switch(config)# feature bgp`	Enables BGP.
Step 3	**show feature** **Example:** `switch(config)# show feature`	(Optional) Displays enabled and disabled features.
Step 4	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Saves this configuration change.

Use the **no feature bgp** command to disable BGP and remove all associated configuration.

Command	Purpose
no feature bgp **Example:** `switch(config)# no feature bgp`	Disables BGP and removes all associated configuration.

Creating a BGP Instance

You can create a BGP instance and assign a router ID to the BGP instance. For more information, see the "BGP Router Identifier" section on page 9-3.

BEFORE YOU BEGIN

You must enable BGP (see the "Enabling BGP" section on page 9-11).

BGP must be able to obtain a router ID (for example, a configured loopback address).

SUMMARY STEPS

1. **configure terminal**
2. **router bgp** *autonomous-system-number*
3. (Optional) **router-id** *ip-address*
4. (Optional) **address-family** {**ipv4** | **ipv6**} {**unicast** | **multicast**}
5. (Optional) **network** {*ip-address*/*length* | *ip-address* **mask** *mask*} [**route-map** *map-name*]
6. (Optional) **show bgp all**
7. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

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

Step 2	**router bgp** *autonomous-system-number*
	
	Example:
	`switch(config)# router bgp 64496`
	`switch(config-router)#`
		Enables BGP and assigns the AS 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.

Step 3	**router-id** *ip-address* **Example:** `switch(config-router)# router-id 192.0.2.255`	(Optional) Configures the BGP router ID. This IP address identifies this BGP speaker.

Step 4	**address-family** {**ipv4** | **ipv6**} {**unicast** | **multicast**}
	
	Example:
	`switch(config-router)# address-family ipv4 unicast`
	`switch(config-router-af)#`
		(Optional) Enters global address family configuration mode for the IPv4 or IPv6 address family.

	Command	Purpose
Step 5	**network** {*ip-address*/*length* | *ip-address* **mask** *mask*} [**route-map** *map-name*] **Example**: switch(config-router-af)# network 10.10.10.0/24 **Example**: switch(config-router-af)# network 10.10.10.0 mask 255.255.255.0	(Optional) Specifies a network as local to this autonomous system and adds it to the BGP routing table. For exterior protocols, the network command controls which networks are advertised. Interior protocols use the **network** command to determine where to send updates.
Step 6	**show bgp all** **Example**: switch(config-router-af)# show bgp all	(Optional) Displays information about all BGP address families.
Step 7	**copy running-config startup-config** **Example:** switch(config-router-af)# copy running-config startup-config	(Optional) Saves this configuration change.

Use the **no router bgp** command to remove the BGP process and the associated configuration.

Command	Purpose
no router bgp *autonomous-system-number* **Example:** switch(config)# no router bgp 201	Deletes the BGP process and the associated configuration.

This example shows how to enable BGP with the IPv4 unicast address family and manually add one network to advertise:

```
switch# configure terminal
switch(config)# router bgp 64496
switch(config-router)# address-family ipv4 unicast
switch(config-router-af)# network 192.0.2.0
switch(config-router-af)# copy running-config startup-config
```

Restarting a BGP Instance

You can restart a BGP instance and clear all peer sessions for the instance.

To restart a BGP instance and remove all associated peers, use the following command:

Command	Purpose
restart bgp instance-tag **Example:** switch(config)# restart bgp 201	Restarts the BGP instance and resets or reestablishes all peering sessions.

Shutting Down BGP

You can shut down the BGP and gracefully disable BGP while retaining the configuration.

To shut down BGP, use the following command in router configuration mode:

Command	Purpose
shutdown **Example:** switch(config-router)# shutdown	Gracefully shuts down BGP.

Configuring BGP Peers

You can configure a BGP peer within a BGP process. Each BGP peer has an associated keepalive timer and hold timers. You can set these timers either globally or for each BGP peer. A peer configuration overrides a global configuration.

Note You must configure the address family under neighbor configuration mode for each peer.

BEFORE YOU BEGIN

You must enable BGP (see the "Enabling BGP" section on page 9-11).

SUMMARY STEPS

1. **configure terminal**
2. **router bgp** *autonomous-system-number*
3. **neighbor** {*ip-address* | *ipv6-address*} **remote-as** *as-number*
4. (Optional) **description** *text*
5. (Optional) **timers** *keepalive-time hold-time*
6. (Optional) **shutdown**
7. **address-family** {**ipv4** | **ipv6**} {**unicast** | **multicast**}
8. (Optional) **weight** *value*

9. (Optional) **show bgp** {**ipv4** | **ipv6**} {**unicast** | **multicast**} **neighbors**
10. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal** **Example:** switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	**router bgp** *autonomous-system-number* **Example:** switch(config)# router bgp 64496 switch(config-router)#	Enables BGP and assigns the AS 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.
Step 3	**neighbor** {*ip-address* | *ipv6-address*} **remote-as** *as-number* **Example:** switch(config-router)# neighbor 209.165.201.1 remote-as 64497 switch(config-router-neighbor)#	Configures the IPv4 or IPv6 address and AS number for a remote BGP peer. The *ip-address* format is x.x.x.x. The *ipv6-address* format is A:B::C:D.
Step 4	**description** *text* **Example:** switch(config-router-neighbor)# description Peer Router B switch(config-router-neighbor)#	(Optional) Adds a description for the neighbor. The description is an alphanumeric string up to 80 characters.
Step 5	**timers** *keepalive-time hold-time* **Example:** switch(config-router-neighbor)# timers 30 90	(Optional) Adds the keepalive and hold time BGP timer values for the neighbor. The range is from 0 to 3600 seconds. The default is 60 seconds for the keepalive time and 180 seconds for the hold time.
Step 6	**shutdown** **Example:** switch(config-router-neighbor)# shutdown	(Optional). Administratively shuts down this BGP neighbor. This command triggers an automatic notification and session reset for the BGP neighbor sessions.
Step 7	**address-family** {**ipv4** | **ipv6**} {**unicast** | **multicast**} **Example:** switch(config-router-neighbor)# address-family ipv4 unicast switch(config-router-neighbor-af)#	Enters neighbor address family configuration mode for the IPv4 or IPv6 address family.

	Command	Purpose
Step 8	**weight** *value* **Example:** `switch(config-router-neighbor-af)# weight 100`	(Optional) Sets the default weight for routes from this neighbor. The range is from 0 to 65535. All routes learned from this neighbor have the assigned weight initially. The route with the highest weight is chosen as the preferred route when multiple routes are available to a particular network. The weights assigned with the **set weight route-map** command override the weights assigned with this command. If you specify a BGP peer policy template, all the members of the template inherit the characteristics configured with this command.
Step 9	**show bgp** {**ipv4** | **ipv6**} {**unicast** | **multicast**} **neighbors** **Example:** `switch(config-router-neighbor-af)# show bgp ipv4 unicast neighbors`	(Optional) Displays information about BGP peers.
Step 10	**copy running-config startup-config** **Example:** `switch(config-router-neighbor-af) copy running-config startup-config`	(Optional) Saves this configuration change.

This example shows how to configure a BGP peer:

```
switch# configure terminal
switch(config)# router bgp 64496
switch(config-router)# neighbor 192.0.2.1 remote-as 64497
switch(config-router-neighbor)# description Peer Router B
switch(config-router-neighbor)# address-family ipv4 unicast
switch(config-router-neighbor-af)# weight 100
switch(config-router-neighbor-af)# copy running-config startup-config
```

Configuring Dynamic AS Numbers for Prefix Peers

You can configure multiple BGP peers within a BGP process. You can limit BGP session establishment to a single AS number or multiple AS numbers in a route map.

BGP sessions configured through dynamic AS numbers for prefix peers ignore the **ebgp-multihop** command and the **disable-connected-check** command.

You can change the list of AS numbers in the route map, but you must use the **no neighbor** command to change the route-map name. Changes to the AS numbers in the configured route map affect only new sessions.

BEFORE YOU BEGIN

You must enable BGP (see the "Enabling BGP" section on page 9-11).

SUMMARY STEPS

1. **configure terminal**
2. **router bgp** *autonomous-system-number*
3. **neighbor** *prefix* **remote-as route-map** *map-name*
4. (Optional) **show bgp** {**ipv4** | **ipv6**} {**unicast** | **multicast**} **neighbors**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

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

Step 2	**router bgp** *autonomous-system-number*
	
	Example:
	`switch(config)# router bgp 64496`
	`switch(config-router)#`
		Enables BGP and assigns the AS 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.

Step 3	**neighbor** *prefix* **remote-as route-map** *map-name*
	
	Example:
	`switch(config-router)# neighbor 192.0.2.0/8 remote-as routemap BGPPeers`
	`switch(config-router-neighbor)#`
		Configures the IPv4 or IPv6 prefix and a route map for the list of accepted AS numbers for the remote BGP peers. The *prefix* format for IPv4 is x.x.x.x/length. The length range is from 1 to 32. The *prefix* format for IPv6 is A:B::C:D/length. The length range is from 1 to 128.
		The *map-name* can be any case-sensitive, alphanumeric string up to 63 characters.

Step 4	**show bgp** {**ipv4** | **ipv6**} {**unicast** | **multicast**} **neighbors** **Example**: `switch(config-router-neighbor-af)# show bgp ipv4 unicast neighbors`	(Optional) Displays information about BGP peers.
Step 5	**copy running-config startup-config** **Example:** `switch(config-router-neighbor-af)# copy running-config startup-config`	(Optional) Saves this configuration change.

This example shows how to configure dynamic AS numbers for a prefix peer:

```
switch# configure terminal
switch(config)# route-map BGPPeers
switch(config-route-map)# match as-number 64496, 64501-64510
switch(config-route-map)# match as-number as-path-list List1, List2
switch(config-route-map)# exit
switch(config)# router bgp 64496
switch(config-router)# neighbor 192.0.2.0/8 remote-as route-map BGPPeers
switch(config-router-neighbor)# description Peer Router B
switch(config-router-neighbor)# address-family ipv4 unicast
switch(config-router-neighbor-af)# copy running-config startup-config
```

See Chapter 15, "Configuring Route Policy Manager." for information on route maps.

Clearing BGP Information

To clear BGP information, use the following commands:

Command	Purpose
clear bgp all {*neighbor* | * | *as-number* | **peer-template** *name* | *prefix*} [**vrf** *vrf-name*]	Clears one or more neighbors from all address families. * clears all neighbors in all address families. The arguments are as follows: • *neighbor*—IPv4 or IPv6 address of a neighbor. • *as-number*—Autonomous system number. The AS number can be a 16-bit integer or a 32-bit integer in the form of higher 16-bit decimal number and a lower 16-bit decimal number in xx.xx format. • *name*—Peer template name. The name can be any case-sensitive, alphanumeric string up to 64 characters. • *prefix*—IPv4 or IPv6 prefix. All neighbors within that prefix are cleared. • *vrf-name*—VRF name. All neighbors in that VRF are cleared. The name can be any case-sensitive, alphanumeric string up to 64 characters.
clear bgp all dampening [**vrf** *vrf-name*]	Clears route flap dampening networks in all address families. The *vrf-name* can be any case-sensitive, alphanumeric string up to 64 characters.
clear bgp all flap-statistics [**vrf** *vrf-name*]	Clears route flap statistics in all address families. The *vrf-name* can be any case-sensitive, alphanumeric string up to 64 characters.
clear bgp {**ipv4** | **ipv6**} {**unicast** | **multicast**} **dampening** [**vrf** *vrf-name*]	Clears route flap dampening networks in the selected address family. The *vrf-name* can be any case-sensitive, alphanumeric string up to 64 characters.
clear bgp {**ipv4** | **ipv6**} {**unicast** | **multicast**} **flap-statistics** [**vrf** *vrf-name*]	Clears route flap statistics in the selected address family. The *vrf-name* can be any case-sensitive, alphanumeric string up to 64 characters.

Command	Purpose
clear bgp {**ipv4** | **ipv6**} {**unicast** | **multicast**} {*neighbor* | ***** | *as-number* | **peer-template** *name* | *prefix*} [**vrf** *vrf-name*]	Clears one or more neighbors from the selected address family. ***** clears all neighbors in the address family. The arguments are as follows: • *neighbor*—The IPv4 or IPv6 address of a neighbor. • *as-number*— Autonomous system number. The AS number can be a 16-bit integer or a 32-bit integer in the form of higher 16-bit decimal number and a lower 16-bit decimal number in xx.xx format. • *name*—Peer template name. The name can be any case-sensitive, alphanumeric string up to 64 characters. • *prefix*—IPv4 or IPv6 prefix. All neighbors within that prefix are cleared. • *vrf-name*—VRF name. All neighbors in that VRF are cleared. The name can be any case-sensitive, alphanumeric string up to 64 characters.
clear ip bgp {**ip** {**unicast** | **multicast**}} {*neighbor* | ***** | *as-number* | **peer-template** *name* | *prefix*} [**vrf** *vrf-name*]	Clears one or more neighbors. ***** clears all neighbors in the address family. The arguments are as follows: • *neighbor*—IPv4 or IPv6 address of a neighbor. • *as-number*— Autonomous system number. The AS number can be a 16-bit integer or a 32-bit integer in the form of higher 16-bit decimal number and a lower 16-bit decimal number in xx.xx format. • *name*—Peer template name. The name can be any case-sensitive, alphanumeric string up to 64 characters. • *prefix*—IPv4 or IPv6 prefix. All neighbors within that prefix are cleared. • *vrf-name*—VRF name. All neighbors in that VRF are cleared. The name can be any case-sensitive, alphanumeric string up to 64 characters.

Command	Purpose
clear ip bgp dampening [*ip-neighbor* | *ip-prefix*] [**vrf** *vrf-name*]	Clears route flap dampening in one or more networks. The arguments are as follows: • *ip-neighbor*—IPv4 address of a neighbor. • *ip-prefix*—IPv4. All neighbors within that prefix are cleared. • *vrf-name*—VRF name. All neighbors in that VRF are cleared. The name can be any case-sensitive, alphanumeric string up to 64 characters.
clear ip bgp flap-statistics [*ip-neighbor* | *ip-prefix*] [**vrf** *vrf-name*]	Clears route flap statistics in one or more networks. The arguments are as follows: • *ip-neighbor*—IPv4 address of a neighbor. • *ip-prefix*—IPv4. All neighbors within that prefix are cleared. • *vrf-name*—VRF name. All neighbors in that VRF are cleared. The name can be any case-sensitive, alphanumeric string up to 64 characters.
clear ip mbgp {**ip** {**unicast** | **multicast**}} {*neighbor* | ***** | *as-number* | **peer-template** *name* | *prefix*} [**vrf** *vrf-name*]	Clears one or more neighbors. * clears all neighbors in the address family. The arguments are as follows: • *neighbor*—IPv4 or IPv6 address of a neighbor. • *as-number*— Autonomous system number. The AS number can be a 16-bit integer or a 32-bit integer in the form of higher 16-bit decimal number and a lower 16-bit decimal number in xx.xx format. • *name*—Peer template name. The name can be any case-sensitive, alphanumeric string up to 64 characters. • *prefix*—IPv4 or IPv6 prefix. All neighbors within that prefix are cleared. • *vrf-name*—VRF name. All neighbors in that VRF are cleared. The name can be any case-sensitive, alphanumeric string up to 64 characters.

Command	Purpose
clear ip mbgp dampening [*ip-neighbor* | *ip-prefix*] [**vrf** *vrf-name*]	Clears route flap dampening in one or more networks. The arguments are as follows: • *ip-neighbor*—IPv4 address of a neighbor. • *ip-prefix*—IPv4. All neighbors within that prefix are cleared. • *vrf-name*—VRF name. All neighbors in that VRF are cleared. The name can be any case-sensitive, alphanumeric string up to 64 characters.
clear ip mbgp flap-statistics [*ip-neighbor* | *ip-prefix*] [**vrf** *vrf-name*]	Clears route flap statistics one or more networks. The arguments are as follows: • *ip-neighbor*—IPv4 address of a neighbor. • *ip-prefix*—IPv4. All neighbors within that prefix are cleared. • *vrf-name*—VRF name. All neighbors in that VRF are cleared. The name can be any case-sensitive, alphanumeric string up to 64 characters.

Verifying the Basic BGP Configuration

To display the BGP configuration, 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 {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **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*] {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **community-list** *list-name* [**vrf** *vrf-name*]	Displays the BGP routes that match a BGP community list.
show bgp {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **extcommunity** {**regexp** *expression* | **generic** [**non-transitive** | **transitive**] *aa4:nn* [**exact-match**]} [**vrf** *vrf-name*]	Displays the BGP routes that match a BGP extended community.
show bgp {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **extcommunity-list** *list-name* [**exact-match**]} [**vrf** *vrf-name*]	Displays the BGP routes that match a BGP extended community list.

Command	Purpose
show bgp {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] {**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 {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **history-paths** [**regexp** *expression*] [**vrf** *vrf-name*]	Displays the BGP route history paths.
show bgp {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **filter-list** *list-name* [**vrf** *vrf-name*]	Displays the information for the BGP filter list.
show bgp {**ipv4** | **ipv6**} {**unicast** | **multicast**} [i*p-address* | *ipv6-prefix*] **neighbors** [*ip-address* | *ipv6-prefix*] [**vrf** *vrf-name*]	Displays the information for BGP peers. Use the **clear bgp neighbors** command to clear these neighbors.
show bgp {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] {**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 {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **policy** *name* [**vrf** *vrf-name*]	Displays the BGP policy information. Use the **clear bgp policy** command to clear the policy information.
show bgp {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **prefix-list** *list-name* [**vrf** *vrf-name*]	Displays the BGP routes that match the prefix list.
show bgp {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **received-paths** [**vrf** *vrf-name*]	Displays the BGP paths stored for soft reconfiguration.
show bgp {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **regexp** *expression* [**vrf** *vrf-name*]	Displays the BGP routes that match the AS_path regular expression.
show bgp {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **route-map** *map-name* [**vrf** *vrf-name*]	Displays the BGP routes that match the route map.
show bgp peer-policy *name* [**vrf** *vrf-name*]	Displays the information about BGP peer policies.
show bgp peer-session *name* [**vrf** *vrf-name*]	Displays the information about BGP peer sessions.
show bgp peer-template *name* [**vrf** *vrf-name*]	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** | **ipv6**} **bgp** *options*	Displays the BGP status and configuration information.

Command	Purpose
show {**ip** | **ipv6**} **mbgp** *options*	Displays the BGP status and configuration information.
show running-configuration bgp	Displays the current running BGP configuration.

Monitoring BGP Statistics

To display BGP statistics, use the following commands:

Command	Purpose
show bgp {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **flap-statistics** [**vrf** *vrf-name*]	Displays the BGP route flap statistics.
show bgp sessions [**vrf** *vrf-name*]	Displays the BGP sessions for all peers.
show bgp statistics	Displays the BGP statistics.

Configuration Examples for Basic BGP

This example shows a basic BGP configuration:

```
switch(config)# feature bgp
switch(config)# router bgp 64496
switch(config-router)# neighbor 2001:DB8:0:1::55 remote-as 64496
switch(config-router-neighbor)# address-family ipv6 unicast
switch(config-router-af)# next-hop-self
```

Related Topics

The following topics relate to BGP:

- Chapter 10, “Configuring Advanced BGP”
- Chapter 15, “Configuring Route Policy Manager”

Where to Go Next

See Chapter 10, "Configuring Advanced BGP," for details on the following features:

- Peer templates
- Route redistribution
- Route maps

Additional References

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

- MIBs, page 9-24

MIBs

MIBs	MIBs Link
MIBs related to BGP	To locate and download supported MIBs, go to the following URL: ftp://ftp.cisco.com/pub/mibs/supportlists/nexus9000/Nexus9000MIBSupportList.html