Configuring Basic BGP

This chapter contains the following sections:

Finding Feature Information

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

Information About 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.

BGP also supports load balancing or equal-cost multipath (ECMP).

For information on configuring BGP in an MPLS network, see the Cisco Nexus 7000 Series NX-OS MPLS Configuration Guide.

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.

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 or 4-byte AS numbers. Cisco NX-OS displays 4-byte AS numbers in plain-text notation (that is, as 32-bit integers). You can configure 4-byte AS numbers as either plain-text notation (for example, 1 to 4294967295) or AS.dot notation (for example, 1.0).

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 the table.

Table 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	200	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.

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.

Note The dynamic AS number prefix peer configuration overrides the individual AS number configuration that is inherited from a BGP template.

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

Although BGP might receive advertisements for the same route from multiple sources, BGP selects only one path as the best path. BGP puts the selected path in the IP routing table and propagates the path to its peers.

Note Beginning with Cisco NX-OS Release 6.1, BGP supports sending and receiving multiple paths per prefix and advertising such paths.

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.

Beginning with Cisco NX-OS Release 8.4(1), the behavior of the BGP pre-best path point of insertion (POI) is changed. In this release, the NX-OS RPM, BGP, and HMM software uses a single cost community ID (either 128 for internal routes or 129 for external routes) to identify a BGP VPNv4 route as an EIGRP originated route.

Only the routes that have the pre-best path value set to cost community ID 128 or 129 are installed in the URIB along with the cost extcommunity. Any non-eigrp originated route carrying the above described cost community ID would be installed in URIB along with pre-best path cost community. As a result, URIB would use this cost to identify the better route between the route learnt through the iBGP and backdoor-EIGRP instead of the administrative distance.

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

1. Compares two paths to determine which is better.
2. Explores all paths and determines in which order to compare the paths to select the overall best path.
3. Determines whether the old and new best paths differ enough so that the new best path should be used.

Note The order of comparison determined in Part 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.

BGP Path Selection - 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.

6. Cisco NX-OS chooses the path with the lower origin. 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. 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.

   e. If the non-deterministic MED comparison feature is enabled, the best-path algorithm uses the Cisco IOS style of MED comparison.

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 it was run.
11. If all path parameters in Step 1 through Step 9 are the same, and there is no current best path (for example, the current best path can be lost when the neighbor that offers the current best path goes

down), then the route from the BGP router with the lowest router ID is chosen. 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.

12. 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.
13. 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.

BGP Path Selection - 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 rule as in the section *Step 1—Comparing Pairs of Paths* 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.

BGP Path Selection - 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 11-10 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

The BGP Prefix Independent Convergence (PIC) feature achieves subsecond convergence in the forwarding plane for BGP IP and Layer 3 VPN routes, when there are BGP next-hop network reachability failures.

BGP PIC has two categories:

- PIC core
- PIC edge

PIC core ensures fast convergence for BGP routes when there is a link or node failure in the core that causes a change in the IGP reachability to a remote BGP next-hop address.

PIC edge ensures fast convergence to a BGP backup path when an external (eBGP) edge link or an external neighbor node fails.

BGP PIC Feature Support Matrix

BGP PIC feature support matrix is shown in the table below:

BGP PIC	IPv4 Unicast	IPv6 Unicast	VPNv4 (per prefix)	VPNv6 (per prefix)	VPNv4 (per VRF)	VPNv6 (per VRF)
Core Unipath	Yes	Yes	No	No	Yes	No
Edge Unipath	Yes	Yes	No	No	No	No

BGP PIC	IPv4 Unicast	IPv6 Unicast	VPNv4 (per prefix)	VPNv6 (per prefix)	VPNv4 (per VRF)	VPNv6 (per VRF)
Core with Multipath equal	Yes	Yes	No	No	Yes	No
Edge Multipath equal (multiple active ECMP, only one backup)	Yes	Yes	No	No	No	No

BGP PIC Core

The BGP PIC core feature is supported by Cisco NX-OS Release 5.2 and later. The 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. Both MPLS and pure IP traffic can benefit from this feature. It is enabled by default and cannot be disabled.

IPv4, VPNv4, 6PE, and VPNv6 (6VPE) support PIC core with the following constraints:

- For both IP and MPLS core, convergence for internet routes is prefix-independent on the order of BGP next hops.
- With per-VRF label allocation, VPN route convergence is also prefix-independent on the order of BGP next hops. That is, when a path to a remote PE changes, the number of VRFs on that PE determines convergence.
- With per-prefix label allocation, route convergence is not prefix-independent. Convergence moves to the order of VPN routes that are advertised by a remote PE if a failure or change occurs in the reachability to that PE.

For additional considerations when using BGP PIC core in MPLS networks, see the *Cisco Nexus 7000 Series NX-OS MPLS Configuration Guide*.

BGP PIC Edge

The BGP PIC for Edge feature improves BGP convergence after a network failure. This convergence is applicable to edge failures in an IP network. The BGP PIC Edge feature creates and stores a backup path in the routing information base (RIB) and forwarding information base (FIB) so that when a failure on an eBGP link to SP is detected (the primary path fails), the backup path can immediately take over, enabling fast fail over in the forwarding plane.

Note From Cisco NX-OS Release 7.3(0)D1(1) onwards BGP PIC Edge feature supports both IPv4 and IPv6 address families.

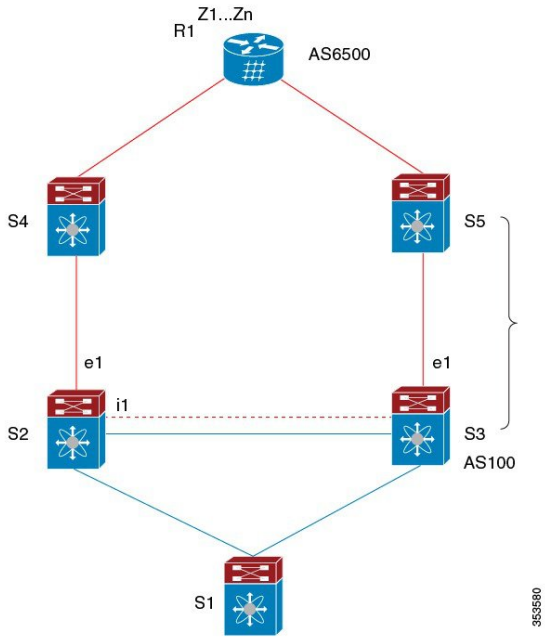
If BGP PIC edge is configured, BGP calculates an additional second best-path (the backup path) along with the primary best-path. BGP installs both best and backup paths for the prefixes with PIC support into the BGP RIB. BGP also downloads the backup path along with the RNH via APIs to the URIB, which then updates the FIB with the next hop marked as a backup. The backup path provides a fast reroute mechanism to counter a singular network failure.

This feature detects both the local interface failure and remote interface/link failure and triggers the use of the backup path.

BGP PIC Edge Unipath

A BGP PIC edge unipath topology is shown in the figure below:

Figure 1: BGP PIC Edge Unipath



In the above figure:

- eBGP sessions are between S2-S4 and S3-S5
- iBGP session is between S2-S3
- Traffic from S1 uses S2 and uses the e1 interface to reach prefixes Z1..Zn.
- S2 has two paths to reach Z1…Zn
  - Primary path via S4
  - Backup/alternate via S5

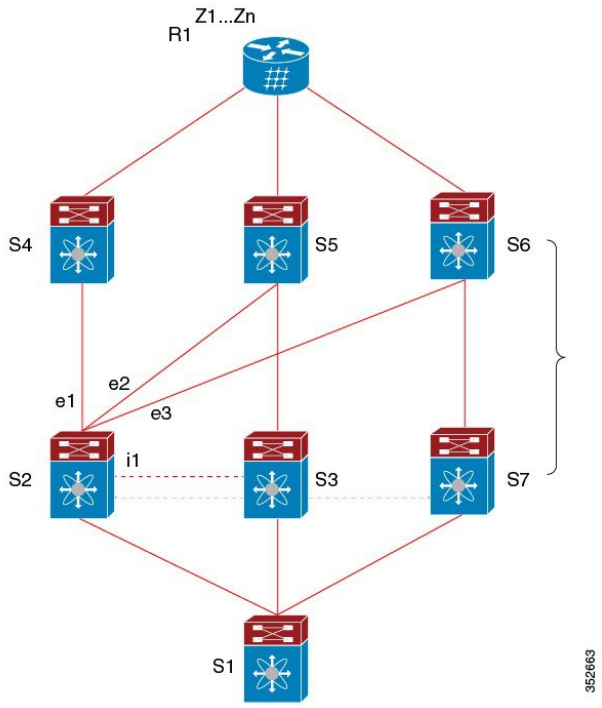
In this example, S3 advertises to S2 the prefixes Z1…Zn to reach with itself as the next hop. BGP on S2, with BGP PIC feature enabled, installs both bestpath (via S4) and backup path (via S3/S5) towards the AS6500 into the RIB and then the RIB downloads both routes to the FIB.

When the S2-S4 link goes down, the FIB on S2 detects the link failure. It automatically switches from the primary path to the backup/alternate and points to the new next hop S3. Traffic is quickly rerouted due to the local fast re-convergence in FIB. After learning the link failure event, BGP on S2 recomputes the bestpath (which is the previous backup path), removing the next hop S4 from RIB and reinstalling S3 as the primary next hop into RIB. It also computes a new backup/alternate path, if any, and notifies RIB. With the support of the BGP PIC feature, the FIB can switch to the available backup route instantly upon detection of link failure on the primary route without waiting for BGP to select new bestpath and converge, and achieve a fast reroute.

BGP PIC Edge with Multipaths

In the presence of Equal Cost Multipath (ECMP), none of the multipaths can be selected as the backup path when BGP PIC Edge support is enabled.

Figure 2: BGP PIC Edge with Multipaths



In the above topology, there are six paths for a given prefix as follows:

- eBGP paths: e1, e2, e3
- iBGP paths: i1, i2, i3

The order of preference is e1 > e2 > e3 > i1 > i2 > i3.

The potential multipath situations are:

No multipaths configured

- bestpath = e1
- multipath-set = []
- backup path = e2
- PIC behavior: When e1 fails, e2 is activated.

Two-way eBGP multipaths configured:

- bestpath = e1
- multipath-set = [e1, e2]
- backup path = e3

- PIC behavior: Active multipaths are mutually backed up. When all multipaths fail, e3 is activated.

Three-way eBGP multipaths configured:

- bestpath = e1
- multipath-set = [e1, e2, e3]
- backup path = i1
- PIC behavior: Active multipaths are mutually backed up. When all multipaths fail, i1 is activated.

Four-way eiBGP multipaths configured:

- bestpath = e1
- multipath-set = [e1, e2, e3, i1]
- backup path = i2
- PIC behavior: Active multipaths are mutually backed up. When all multipaths fail, i2 is activated.

BGP Virtualization

BGP supports virtual routing and forwarding (VRF) instances. VRFs exist within virtual device contexts (VDCs). By default, Cisco NX-OS places you in the default VDC and default VRF unless you specifically configure another VDC and VRF. For more information, see the *Cisco Nexus 7000 Series NX-OS Virtual Device Context Configuration Guide*.

Prerequisites for BGP

BGP has the following prerequisites:

- You must enable BGP.
- 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:

- Cisco NX-OS does not support "fast-external-fallover" for the multi-hop eBGP peering. The BGP differentiates the single-hop (directly connected) and the multi-hop eBGP neighbors using the **ebgp-multihop** command. When you use the **ebgp-multihop 2** command for an eBGP peer, the BGP treats it as multi-hop session and does not trigger the "fast-external-fallover". This is a known behaviour.

- 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.
- If you decrease the keepalive and hold timer values, you might experience BGP session flaps.
- You can configure a minimum route advertisement interval (MRAI) between the sending of BGP routing updates by using the **advertisement-interval** command.
- The BGP Prefix-Independent Convergence (PIC) Edge feature only supports IPv4 address family.
- Only one repair path (backup path) is supported with the BGP PIC Edge feature.

Default Settings

Table 2: Default BGP Parameters

Parameters	Default
BGP feature	Disabled
Keep alive interval	60 seconds
Hold timer	180 seconds
BGP PIC core	Enabled
BGP PIC edge	Disabled
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.

This example shows how to enter router configuration mode:

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

BGP supports Virtual Routing and Forwarding (VRF). You can configure BGP within the appropriate VRF if you are using VRFs in your network.

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.

The following 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)#
```

The following 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.

The following 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)#
```

The following 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.

The following 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.

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.

Before you begin

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **feature bgp**	Enables BGP. Use the **no feature bgp** command to disable BGP and remove all associated configuration.
Step 3	(Optional) switch(config)# **show feature**	(Optional) Displays enabled and disabled features.

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

Creating a BGP Instance

You can create a BGP instance and assign a router ID to the BGP instance. Cisco NX-OS supports 2-byte or 4-byte autonomous system (AS) numbers in plain-text notation or as.dot notation.

Before you begin

- You must enable BGP.
- BGP must be able to obtain a router ID (for example, a configured loopback address).
- Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **router bgp***autonomous-system-number*	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. Use the **no** form of this command to disable this feature.
Step 3	switch(config-router)# **router-id** *ip-address*	(Optional) Configures the BGP router ID. This IP address identifies this BGP speaker.
Step 4	switch(config-router)# **address-family** {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**}{**unicast** | **multicast**}	Enters global address family configuration mode for the IP or VPN address family.
Step 5	switch(config-router-af)# **network ip-prefix** [**route-map** *map-name*]	(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	switch(config-router-af)# **show bgp all**	(Optional) Displays information about all BGP address families.

	Command or Action	Purpose
Step 7	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to 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:

Procedure

	Command or Action	Purpose
Step 1	switch(config)# **restart bgp** *instance-tag*	Restarts the BGP instance and resets or reestablishes all peering sessions.

Shutting Down BGP

You can shut down the BGP protocol and gracefully disable BGP and retain the configuration.

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

Procedure

	Command or Action	Purpose
Step 1	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.
- Ensure that you are in the correct VDC (or use the **switchto vdc**command).

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **router bgp** *autonomous-system-number*	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	switch(config-router)# **neighbor** {*ip-address* | *ipv6-address*} **remote-as** *as-number*	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	switch(config-router-neighbor)# **description** *text*	(Optional) Adds a description for the neighbor. The description is an alphanumeric string up to 80 characters.
Step 5	switch(config-router-neighbor)# **timers** *keepalive-time hold-time*	(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	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	switch(config-router-neighbor)# **address-family** {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} {**unicast** | **multicast**}	Enters neighbor address family configuration mode for the unicast IPv4 address family.
Step 8	switch(config-router-neighbor)# **weight** *value*	(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

	Command or Action	Purpose
		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	(Optional) switch(config-router-neighbor)# **show bgp** {**ipv4**|**ipv6**|**vpnv4**|**vpnv6**} {**unicast**|**multicast**} **neighbors**	(Optional) Displays information about BGP peers.
Step 10	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

The following 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)# weight 100
switch(config-router-neighbor-af)# copy running-config startup-config
```

Configuring AS-4 Dot Notation

You can configure 4-byte autonomous system (AS) numbers in asdot notation. The default value is asplain.

Before you begin

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **as-format asdot**	Configures the ASN notation to asdot.
Step 3	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

The following example configures AS numbers in asdot notation.

```
switch # configure terminal
switch (config) # as-format asdot
switch (config) # 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.
- Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **router bgp** *autonomous-system-number*	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	switch(config-router)# **neighbor** *prefix* **remote-as route-map** *map-name*	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	switch(config-router-neighbor-af)# **show bgp** {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} {**unicast** | **multicast**} **neighbors**	(Optional) Displays information about BGP peers.

	Command or Action	Purpose
Step 5	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to configure 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
```

Configuring BGP PIC Edge

Note The BGP PIC Edge feature only supports IPv4 address families.

Before you begin

- You must enable BGP.
- Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Procedure

Step 1 Enter configuration mode:

switch#**configure terminal**

Step 2 Enable BGP and assign the autonomous system number to the local BGP speaker:

switch(config)# **router bgp** *autonomous-system-number*

Step 3 Enter router address family configuration mode for the IPv4 unicast address family:

switch(config-router)# **address-family ipv4 unicast**

Step 4 Enable BGP to install the backup path to the routing table:

switch(config-router-af)# **additional-paths install backup**

Step 5 Exit router address family configuration mode:

switch(config-router-af)# **exit**

Example

This example shows how to configure the device to support BGP PIC Edge in IPv4 network:

```
interface Ethernet2/2
  ip address 1.1.1.5/24
  no shutdown

interface Ethernet2/3
  ip address 2.2.2.5/24
  no shutdown

router bgp 100
  address-family ipv4 unicast
additional-paths install backup
  neighbor 1.1.1.6 remote-as 200
    address-family ipv4 unicast
  neighbor 2.2.2.6 remote-as 100
    address-family ipv4 unicast
```

If BGP receives the same prefix (for example, 99.0.0.0/24) from the two neighbors 1.1.1.6 and 2.2.2.6, both paths will be installed in the URIB—one as the primary path and the other as the backup path.

BGP output:

```
switch(config)# show ip bgp 99.0.0.0/24
BGP routing table information for VRF default, address family IPv4 Unicast
BGP routing table entry for 99.0.0.0/24, version 4
Paths: (2 available, best #2)
Flags: (0x00001a) on xmit-list, is in urib, is best urib route

  Path type: internal, path is valid, not best reason: Internal path, backup path
  AS-Path: 200 , path sourced external to AS
    2.2.2.6 (metric 0) from 2.2.2.6 (2.2.2.6)
      Origin IGP, MED not set, localpref 100, weight 0

  Advertised path-id 1
  Path type: external, path is valid, is best path
  AS-Path: 200 , path sourced external to AS
    1.1.1.6 (metric 0) from 1.1.1.6 (99.0.0.1)
      Origin IGP, MED not set, localpref 100, weight 0

  Path-id 1 advertised to peers:
    2.2.2.6
```

URIB output:

```
URIB output:
switch(config)# show ip route 99.0.0.0/24
IP Route Table for VRF "default"
'*' denotes best ucast next-hop
'**' denotes best mcast next-hop
'[x/y]' denotes [preference/metric]
'%<string>' in via output denotes VRF <string>

99.0.0.0/24, ubest/mbest: 1/0
    *via 1.1.1.6, [20/0], 14:34:51, bgp-100, external, tag 200
```

```
     via 2.2.2.6, [200/0], 14:34:51, bgp-100, internal, tag 200 (backup)
```

UFIB output:

```
switch# show forwarding route 123.1.1.0 detail module 8

Prefix 123.1.1.0/24, No of paths: 1, Update time: Fri Feb  7 19:00:12 2014
   Vobj id: 141   orig_as: 65002   peer_as: 65100  rnh: 10.3.0.2
   10.4.0.2           Ethernet8/4         DMAC: 0018.bad8.4dfd
    packets: 2             bytes: 3484   Repair path    10.3.0.2           Ethernet8/3
     DMAC: 0018.bad8.4dfd
    packets: 0             bytes: 1
```

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.

Command	Purpose
clear bgp {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} {**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** | **vpnv4** | **vpnv6**} {**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.
clear bgp {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} {*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*—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 {**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-numbe*— 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 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 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*]	• *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 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.

Command	Purpose
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** | **vpnv4** | **vpnv6**} {**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** | **vpnv4** | **vpnv6**} {**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** | **vpnv4** | **vpnv6**} {**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** | **vpnv4** | **vpnv6**} {**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.
show bgp {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} {**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** | **vpnv4** | **vpnv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix* **history-paths** [**regexp** *expression*] [**vrf** *vrf-name*]	Displays the BGP route history paths.

Command	Purpose
show bgp {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} {**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** | **vpnv4** | **vpnv6**} {**unicast** | **multicast**} [*ip-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** | **vpnv4** | **vpnv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **neighbors** [*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** | **vpnv4** | **vpnv6**} {**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** | **vpnv4** | **vpnv6**} {**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** | **vpnv4** | **vpnv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **received-paths** [**vrf** *vrf-name*]	Displays the BGP paths stored for soft reconfiguration.
show bgp {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} {**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** | **vpnv4** | **vpnv6**} {**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 {**ipv** | **ipv6**} **bgp options**	Displays the BGP status and configuration information. This command has multiple options. See the *Cisco Nexus 7000 Series NX-OS Unicast Routing Command Reference*, for more information.

Command	Purpose
show {**ipv** | **ipv6**} **mbgp options**	Displays the BGP status and configuration information. This command has multiple options. See the *Cisco Nexus 7000 Series NX-OS Unicast Routing Command Reference*, for more 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** | **vpnv4** | **vpnv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **flap-statistics** [**vrf** *vrf-name*]	Displays the BGP route flap statistics. Use the **clear bgp flap-statistics** command to clear these statistics.
show bgp sessions [**vrf** *vrf-name*]	Displays the BGP sessions for all peers. Use the **clear bgp sessions** command to clear these statistics.
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:ODB8:0:1::55 remote-as 64496
switch (config-router) # address-family ipv6 unicast
switch (config-router-af) # next-hop-self
```

This example shows a basic BGP configuration:

```
switch (config) # address-family
switch (config) # router bgp 64496
switch (config-router) # address-family ipv4 unicast
switch (config-router) # network 1.1.10 mask 255.255.255.0
switch (config-router) # neighbor 10.1.1.1 remote-as 64496
switch (config-router) # address-family ipv4 unicast
```

Related Documents for Basic BGP

Related Topics	Document Title
BGP CLI commands	*Cisco Nexus 7000 Series NX-OS Unicast Routing Command Reference*

Related Topics	Document Title
MPLS configuration	*Cisco Nexus 7000 Series NX-OS MPLS Configuration Guide*
VDCs and VRFs	*Cisco Nexus 7000 Series NX-OS Virtual Device Context Configuration Guide, Release 5.x*

MIBs

MIBs	MIBs Link
BGP4-MIB CISCO-BGP4-MIB CISCO-BGP-MIBv2	To locate and download MIBs, go to the following URL:https://cfnng.cisco.com/mibs.

Feature History for BGP

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

Table 3: Feature History for BGP

Feature Name	Releases	Feature Information
BGP PIC Edge	6.2(8)	Introduced this feature.
BGP	6.2(8)	Added support for CISCO-BGP-MIBv2
4-byte AS number	6.2(2)	Added the ability to configure 4-byte AS numbers in asdot notation.
BGP	6.1(1)	Added support for additional BGP paths.
BGP	6.1(1)	Added the ability to set the default weigh for routes from a neighbor using the **weight** command in the neighbor address family configuration mode.
BGP	5.2(1)	Added support for the BGP PIC core feature.
VPN address families	5.2(1)	Added support for VPN address families.

Feature Name	Releases	Feature Information
BFD	5.0(2)	Added support for BFD. See the *Cisco Nexus 7000 Series NX-OS Interfaces Configuration Guide, Release 6.x* for more information.
ISSU	4.2(3)	Lowered BGP minimum hold-time check to eight seconds.
IPv6	4.2(1)	Added support for IPv6.
4-Byte AS numbers	4.2(1)	Added support for 4-byte AS numbers in plaintext notation.
Conditional advertisement	4.2(1)	Added support for conditionally advertising BGP routes based on the existence of other routes in the BGP table.
Dynamic AS number for prefix peers	4.1(2)	Added support for a range of AS numbers for BGP prefix peer configuration.
BGP	4.0(1)	This feature was introduced.