Configuring Advanced BGP

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

This chapter includes the following sections:

Information About Advanced BGP

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

Peer Templates

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

Cisco NX-OS implements three types of peer templates:

- The **peer-session** template defines BGP peer session attributes, such as the transport details, remote autonomous system number of the peer, and session timers. A peer-session template can also inherit attributes from another peer-session template (with locally defined attributes that override the attributes from an inherited peer-session).

- A **peer-policy** template defines the address-family dependent policy aspects for a peer including the inbound and outbound policy, filter-lists, and prefix-lists. A peer-policy template can inherit from a set of peer-policy templates. Cisco NX-OS evaluates these peer-policy templates in the order specified by the preference value in the inherit configuration. The lowest number is preferred over higher numbers.

- The **peer** template can inherit the peer-session and peer-policy templates to allow for simplified peer definitions. It is not mandatory to use a peer template but it can simplify the BGP configuration by providing reusable blocks of configuration.

Authentication

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

Note The MD5 password must be identical between BGP peers.

Route Policies and Resetting BGP Sessions

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

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

- Hard reset—A hard reset tears down the specified peering sessions, including the TCP connection, and deletes routes coming from the specified peer. This option interrupts packet flow through the BGP network. Hard reset is disabled by default.

- Soft reconfiguration inbound—A soft reconfiguration inbound triggers routing updates for the specified peer without resetting the session. You can use this option if you change an inbound route policy. Soft reconfiguration inbound saves a copy of all routes received from the peer before processing the routes through the inbound route policy. If you change the inbound route policy, Cisco NX-OS passes these stored routes through the modified inbound route policy to update the route table without tearing down existing peering sessions. Soft reconfiguration inbound can use significant memory resources to store the unfiltered BGP routes. Soft reconfiguration inbound is disabled by default.

- Route Refresh—A route refresh updates the inbound routing tables dynamically by sending route refresh requests to supporting peers when you change an inbound route policy. The remote BGP peer responds with a new copy of its routes that the local BGP speaker processes with the modified route policy. Cisco NX-OS automatically sends an outbound route refresh of prefixes to the peer.

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

Note   BGP also uses route maps for route redistribution, route aggregation, route dampening, and other features. See Configuring Route Policy Manager, for more information on route maps.

eBGP

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

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

eBGP Next-Hop Unchanged

In an external BGP (eBGP) session, by default, the device changes the next-hop attribute of a BGP route (to its own address) when the device sends out a route. If the eBGP Next-Hop Unchanged feature is configured, BGP sends routes to an eBGP multihop peer without modifying the next-hop attribute. The next-hop attribute is unchanged. The BGP Next-hop Unchanged feature provides flexibility when designing and migrating networks. It can be used only between eBGP peers configured as multihop.

For example, consider a network with eBGP connection between Devices A, B, and C. Suppose Device A announces 100 prefixes to Device B. Device B is configured with an outbound route map to Device C and the match ip prefix list and set ip next-hop unchanged are configured on the route map. Device B propagates the unchanged next-hop address only for the routes that match the prefix list. For the other prefixes, it puts itself as the next-hop address.

iBGP

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

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

Figure 1: iBGP Network
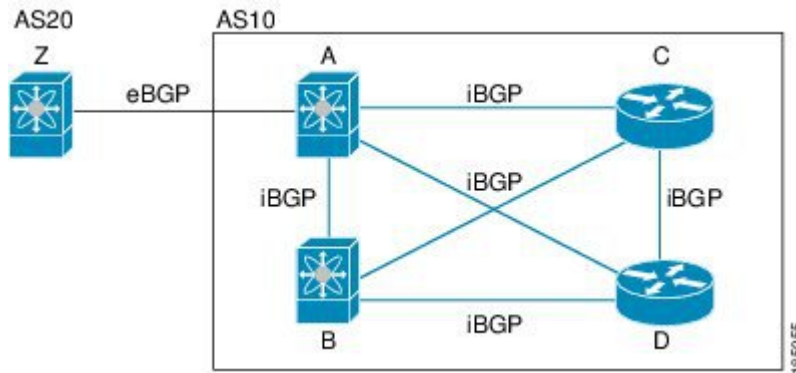
iBGP networks are fully meshed. Each iBGP peer has a direct connection to all other iBGP peers to prevent network loops.


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

AS Confederations

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

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

Figure 2: AS Confederation
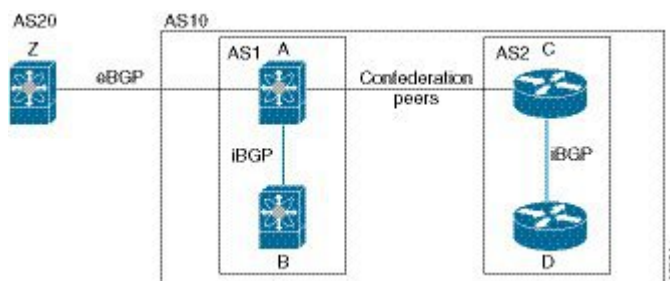
In this example, AS10 is split into two subautonomous systems, AS1 and AS2. Each subautonomous system is fully meshed, but there is only one link between the subautonomous systems. By using AS confederations, you can reduce the number of links compared to the fully meshed autonomous system in Figure **AS Confederation**.
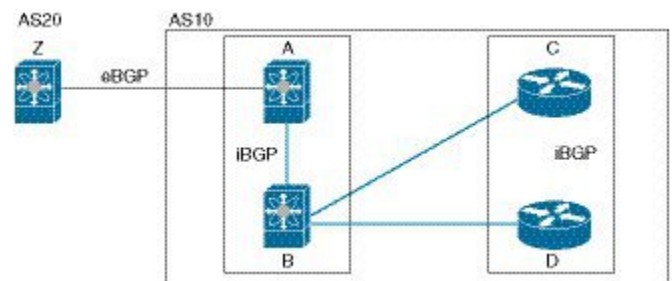
Route Reflector

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

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

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

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

Figure 3: Route Reflector

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

Capabilities Negotiation

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

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

Route Dampening

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

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

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

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

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

Load Sharing and Multipath

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

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

- Weight
- Local preference
- AS_path
- Origin code
- Multi-exit discriminator (MED)
- IGP cost to the BGP next hop

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

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

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

Route Aggregation

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

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

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

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

BGP Conditional Advertisement

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

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

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

BGP Next-Hop Address Tracking

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

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

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

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

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

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

- Noncritical events include only the IGP metric changes.

See the Configuring BGP Next-Hop Address Tracking section for more information.

Site of Origin

The site of origin prevents routing loops when you have a multihomed VPN site. Routes learned from the same site are tagged with the same site-of-origin value that is configured at the PE on all the PE-CE links to the same site. Routes with a particular site-of-origin value are never readvertised back to a CE with the same site-of-origin value configured at the PE-CE link. This process prevents a CE router from relearning routes that originated from the same site. BGP and EIGRP use site of origin to prevent loops.

You can override the autonomous system number (ASN) of a site with the ASN of the provider. This feature is often used with the site of origin to identify the site where a route originated and prevent routing loops between routers within a VPN.

Route Redistribution

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

Labeled and Unlabeled Unicast Routes

In release 7.0(3)I7(6), SAFI-1 (unlabeled unicast) and SAFI-4 (labeled unicast routing) are now supported for IPv4 BGP on a single session. For more information, see the *Cisco Nexus 9000 Series NX-OS Label Switching Configuration Guide, Release 7.x*.

BFD

This feature supports bidirectional forwarding detection (BFD). BFD is a detection protocol designed to provide fast forwarding-path failure detection times. BFD provides subsecond failure detection between two adjacent devices and can be less CPU-intensive than protocol hello messages because some of the BFD load can be distributed onto the data plane on supported modules.

BFD for BGP is supported on eBGP single-hop peers and iBGP single-hop peers. For iBGP single-hop peers using BFD, you must configure the update-source option in neighbor configuration mode. BFD is not supported on other iBGP peers or for multihop eBGP peers.

BFD is supported for the following types of interfaces:

- Layer 3 physical and subinterface

- Layer 3 port channel and subinterface

- Switch virtual interface (SVI)

BFD for BGP does not support authentication or per-link BFD sessions on a port channel.

See Configuring Bidirectional Forwarding Detection for more information.

Tuning BGP

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

BGP Timers

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

Tuning the Best-Path Algorithm

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

Multiprotocol BGP

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

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

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

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

BGP Monitoring Protocol

The BGP Monitoring Protocol (BMP) monitors BGP updates and peer statistics. BMP is supported for all the Cisco Nexus 3000 Series switches.

Using this protocol, the BGP speaker connects to external BMP servers and sends them information regarding BGP events. A maximum of two BMP servers can be configured in a BGP speaker, and each BGP peer can be configured for monitoring by all or a subset of the BMP servers. The BGP speaker does not accept any information from the BMP server.

RFC 5549

Beginning with Cisco NX-OS Release 6.0(2)U4(1), BGP supports RFC 5549, which allows an IPv4 prefix to be carried over an IPv6 next hop. Because BGP is running on every hop, and all routers are capable of forwarding IPv4 and IPv6 traffic, there is no need to support IPv6 tunnels between any routers. BGP installs IPv4 over an IPv6 route to the Unicast Route Information Base (URIB)

Prerequisites for Advanced BGP

BGP has the following prerequisites:

- You must enable the BGP feature (see the Enabling the BGP Feature section).

- You should have a valid router ID configured on the system.

- You must have an AS number, either assigned by a Regional Internet Registry (RIR) or locally administered.

- You must have reachability (such as an interior gateway protocol (IGP), a static route, or a direct connection) to the peer that you are trying to make a neighbor relationship with.

- You must explicitly configure an address family under a neighbor for the BGP session establishment.

Guidelines and Limitations for Advanced BGP

BGP has the following configuration guidelines and limitations:

- Prefix peering operates only in passive TCP mode. It accepts incoming connections from remote peers if the peer address falls within the prefix.

- The dynamic 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 eBGP multihop sessions.

- Specify a BGP route map if you configure redistribution.

- Configure the BGP router ID within a VRF.

- If you decrease the keepalive and hold timer values, the network might experience session flaps.

- The following guidelines and limitations apply to the **remove-private-as** command:

- If the local AS number of the device is a private AS number, you cannot use the remove-private-as configuration command for any other neighbor on the same device. As a workaround, you can use the local-as command on each neighbor with a public local AS number.

- If the real AS number of the device is a private AS number and the remove-private-as all command is configured for a neighbor with a public local-as number, use local-as number [no-prepend [replace-as]] command to ensure that the real private AS number is not appended to the AS path.

- If the real AS number of the device is a public AS number and the remove-private-as all command is configured for a neighbor, you cannot configure a private local-as number for the same neighbor. As a workaround, you must remove the existing configuration to proceed further.

- The remove-private-as all command removes private AS numbers from the AS path even if the path contains both public and private AS numbers.

- The remove-private-as command removes private AS numbers even if the AS path contains only private AS numbers. There is no likelihood of a 0-length AS path because this command can be applied to eBGP peers only, in which case the AS number of the local device is appended to the AS path.

- The remove-private-as command removes private AS numbers even if the private AS numbers appear before the confederation segments in the AS path.

- When you remove private AS numbers from the AS path, the path length of the prefixes that are sent out will decrease. Because the AS path length is a key element of BGP best-path selection, it might be necessary to retain the path length. The replace-as keyword ensures that the path length is retained by replacing all removed AS numbers with the local router's AS number.

- On Cisco Nexus 3000 Series switches running Cisco NX-OS releases 7.0(3)I7(3) and 7.0(3)I7(4), a fragmented BGP Update packet that has MD5 authentication may not be properly reassembled upon reception. This situation can prevent the BGP session from coming up in a stable state. When this condition occurs, an error message similar to the following is displayed:

```
%KERN-3-SYSTEM_MSG: [322896.818155] TCP: MD5 Hash failed for (10.1.1.1,
179)->(10.2.2.2, 29862) - kernel
%KERN-3-SYSTEM_MSG: [322897.020543] TCP: MD5 Hash failed for (10.1.1.1,
179)->(10.2.2.2, 29862) - kernel
```

To prevent this situation, do either of the following:

- Disable BGP MD5 authentication.

- Avoid fragmentation of the BGP packets by setting the MTU value on the outbound interface to match the MTU of the path.

- The following guidelines and limitations apply to BGP Interface Peering via IPv6 Link-Local for IPv4 and IPv6 Address Families:

Default Settings for BGP

Table below lists the default settings for BGP parameters.

Table 1: Default BGP Parameters

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

Configuring Advanced BGP

✎

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

Enabling IP Forward on an Interface

To use RFC 5549, you must configure at least one IPv4 address. If you do not want to configure an IPv4 address, you must enable the ip forward feature to use RFC 5549.

SUMMARY STEPS

1. **configure terminal**
2. **interface** *type slot/port*
3. **ip forward**

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	**interface** *type slot/port*
	
	Example:
	
	`switch(config)# interface ethernet 1/2`
	`switch(config-if)#`
		Enters interface configuration mode.

Step 3	**ip forward**
	
	Example:
	
	`switch(config-if)# ip forward`
	`switch(config-if)#`
		Allows IPv4 traffic on the interface even when there is no IP address configuration on that interface.

Example

This example shows how to enable the ip forward feature on an interface:

```
switch# configure terminal
switch(config)# interface ethernet 1/2
switch(config-if)# ip forward
```

Configuring BGP Session Templates

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

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

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

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

Before you begin

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

When editing a template, you can use the **no** form of a command at either the peer or template level to explicitly override a setting in a template. You must use the **default** form of the command to reset that attribute to the default state.

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	**Purpose**

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

Step 2	**router bgp** *autonomous-system-number*
	
	Example:
	
	`switch(config)# router bgp 65536`
	`switch(config-router)#`
		Enables BGP and assigns the autonomous system number to the local BGP speaker.

Step 3	**template peer-session** *template-name*
	
	Example:
	
	`switch(config-router)# template peer-session`
	`BaseSession`
	`switch(config-router-stmp)#`
		Enters peer-session template configuration mode.

Step 4	(Optional) **password** *number password* **Example:** `switch(config-router-stmp)# password 0 test`	Adds the cleartext password *test* to the neighbor. The password is stored and displayed in type 3 encrypted form (3DES).
Step 5	(Optional) **timers** *keepalive hold* **Example:** `switch(config-router-stmp)# timers 30 90`	Adds the BGP keepalive and holdtimer values to the peer-session template. The default keepalive interval is 60. The default hold time is 180.

Step 6	**exit**
	
	Example:
	
	`switch(config-router-stmp)# exit`
	`switch(config-router)#`
		Exits peer-session template configuration mode.

Step 7	**neighbor** *ip-address* **remote-as** *as-number*
	
	Example:
	
	`switch(config-router)# neighbor 192.168.1.2`
	`remote-as 65536`
	`switch(config-router-neighbor)#`
		Places the router in the neighbor configuration mode for BGP routing and configures the neighbor IP address.

Step 8	**inherit peer-session** *template-name*
	
	Example:
	
	`switch(config-router-neighbor)# inherit`
	`peer-session BaseSession`
	`switch(config-router-neighbor)`
		Applies a peer-session template to the peer.

Step 9	(Optional) **description** *text* **Example:**	Adds a description for the neighbor.

	Command or Action	**Purpose**
| | ```
switch(config-router-neighbor)# description Peer
 Router A
switch(config-router-neighbor)
``` | |
| **Step 10** | (Optional) **show bgp peer-session** *template-name*<br><br>**Example:**<br><br>```
switch(config-router-neighbor)# show bgp
peer-session BaseSession
``` | Displays the peer-policy template. |
| **Step 11** | (Optional) copy running-config startup-config<br><br>**Example:**<br><br>```
switch(config-router-neighbor)# copy
running-config startup-config
``` | Saves this configuration change. |

Example

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

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

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

Configuring BGP Peer-Policy Templates

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

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

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

Before you begin

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

> ✎
>
> **Note** When editing a template, you can use the **no** form of a command at either the peer or template level to explicitly override a setting in a template. You must use the default form of the command to reset that attribute to the default state.

SUMMARY STEPS

1. **configure terminal**
2. **router bgp** *autonomous-system-number*
3. **template peer-policy** *template-name*
4. (Optional) **advertise-active-only**
5. (Optional) **maximum-prefix** *number*
6. **exit**
7. **neighbor** *ip-address* **remote-as** *as-number*
8. **address-family** { **ipv4** | **ipv6**} { **multicast** | **unicast** }
9. **inherit peer-policy** *template-name preference*
10. (Optional) **show bgp peer-policy** *template-name*
11. (Optional) copy running-config startup-config

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	**router bgp** *autonomous-system-number*
	
	Example:
	
	`switch(config)# router bgp 65536`
	`switch(config-router)#`
		Enables BGP and assigns the autonomous system number to the local BGP speaker.

Step 3	**template peer-policy** *template-name*
	
	Example:
	
	`switch(config-router)# template peer-policy`
	`BasePolicy`
	`switch(config-router-ptmp)#`
		Creates a peer-policy template.

Step 4	(Optional) **advertise-active-only** **Example:** `switch(config-router-ptmp)# advertise-active-only`	Advertises only active routes to the peer.
Step 5	(Optional) **maximum-prefix** *number* **Example:** `switch(config-router-ptmp)# maximum-prefix 20`	Sets the maximum number of prefixes allowed from this peer.

	Command or Action	Purpose

Step 6	**exit**
	
	Example:
	`switch(config-router-ptmp)# exit`
	`switch(config-router)#`
		Exits peer-policy template configuration mode.

Step 7	**neighbor** *ip-address* **remote-as** *as-number*
	
	Example:
	`switch(config-router)# neighbor 192.168.1.2`
	`remote-as 65536`
	`switch(config-router-neighbor)#`
		Places the router in neighbor configuration mode for BGP routing and configures the neighbor IP address.

Step 8	**address-family** { **ipv4** | **ipv6**} { **multicast** | **unicast** }
	
	Example:
	`switch(config-router-neighbor)# address-family`
	`ipv4 unicast`
	`switch(config-router-neighbor-af)#`
		Enters global address family configuration mode for the specified address family.

Step 9	**inherit peer-policy** *template-name preference*
	
	Example:
	`switch(config-router-neighbor-af)# inherit`
	`peer-policy BasePolicy 1`
		Applies a peer-policy template to the peer address family configuration and assigns the preference value for this peer policy.

Step 10	(Optional) **show bgp peer-policy** *template-name*
	
	Example:
	`switch(config-router-neighbor-af)# show bgp`
	`peer-policy BasePolicy`
		Displays the peer-policy template.

Step 11	(Optional) copy running-config startup-config
	
	Example:
	`switch(config-router-neighbor-af)# copy`
	`running-config startup-config`
		Saves this configuration change.

Example

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

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

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

Configuring BGP Peer Templates

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

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

Before you begin

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

Note When editing a template, you can use the **no** form of a command at either the peer or template level to explicitly override a setting in a template. You must use the default form of the command to reset that attribute to the default state.

SUMMARY STEPS

1. **configure terminal**
2. **router bgp** *autonomous-system-number*
3. **template peer** *template-name*
4. (Optional) **inherit peer-session** *template-name*
5. (Optional) **address-family** { **ipv4** | **ipv6**} { **multicast** | **unicast** }
6. (Optional) **inherit peer** *template-name*
7. **exit**
8. (Optional) **timers** *keepalive hold*
9. **exit**
10. **neighbor** *ip-address* **remote-as** *as-number*
11. **inherit peer** *template-name*
12. (Optional) **timers** *keepalive hold*
13. (Optional) **show bgp peer-template** *template-name*
14. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	**router bgp** *autonomous-system-number* **Example:** `switch(config)# router bgp 65536`	Enters BGP mode and assigns the autonomous system number to the local BGP speaker.

	Command or Action	Purpose

Step 3	**template peer** *template-name*
	
	Example:
	`switch(config-router)# template peer BasePeer`
	`switch(config-router-neighbor)#`
		Enters peer template configuration mode.

Step 4	(Optional) **inherit peer-session** *template-name*
	
	Example:
	`switch(config-router-neighbor)# inherit`
	`peer-session BaseSession`
		Inherits a peer-session template in the peer template.

Step 5	(Optional) **address-family** { **ipv4** | **ipv6**} { **multicast** | **unicast** }
	
	Example:
	`switch(config-router-neighbor)# address-family`
	`ipv4 unicast`
	`switch(config-router-neighbor-af)#`
		Configures the global address family configuration mode for the specified address family.

Step 6	(Optional) **inherit peer** *template-name*
	
	Example:
	`switch(config-router-neighbor-af)# inherit peer`
	`BasePolicy`
		Applies a peer template to the neighbor address family configuration.

Step 7	**exit**
	
	Example:
	`switch(config-router-neighbor-af)# exit`
	`switch(config-router-neighbor)#`
		Exits BGP neighbor address family configuration mode.

Step 8	(Optional) **timers** *keepalive hold* **Example:** `switch(config-router-neighbor)# timers 45 100`	Adds the BGP timer values to the peer. These values override the timer values in the peer-session template, BaseSession.

Step 9	**exit**
	
	Example:
	`switch(config-router-neighbor)# exit`
	`switch(config-router)#`
		Exits BGP peer template configuration mode.

Step 10	**neighbor** *ip-address* **remote-as** *as-number*
	
	Example:
	`switch(config-router)# neighbor 192.168.1.2`
	`remote-as 65536`
	`switch(config-router-neighbor)#`
		Places the router in neighbor configuration mode for BGP routing and configures the neighbor IP address.

Step 11	**inherit peer** *template-name*
	
	Example:
	`switch(config-router-neighbor)# inherit peer`
	`BasePeer`
		Inherits the peer template.

Step 12	(Optional) **timers** *keepalive hold*	Adds the BGP timer values to this neighbor.

	Command or Action	Purpose
	Example: `switch(config-router-neighbor)# timers 60 120`	These values override the timer values in the peer template and the peer-session template.
Step 13	(Optional) **show bgp peer-template** *template-name* **Example:** `switch(config-router-neighbor-af)# show bgp peer-template BasePeer`	Displays the peer template.
Step 14	(Optional) **copy running-config startup-config** **Example:** `switch(config-router-neighbor-af)# copy running-config startup-config`	Saves this configuration change.

Example

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

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

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

Configuring Prefix Peering

BGP supports the definition of a set of peers using a prefix for both IPv4 and IPv6. This feature allows you to not have to add each neighbor to the configuration.

When defining a prefix peering, you must specify the remote AS number with the prefix. BGP accepts any peer that connects from that prefix and autonomous system if the prefix peering does not exceed the configured maximum peers allowed.

When a BGP peer that is part of a prefix peering disconnects, Cisco NX-OS holds its peer structures for a defined prefix peer timeout value. An established peer can reset and reconnect without danger of being blocked because other peers have consumed all slots for that prefix peering.

To configure the BGP prefix peering timeout value, use the following command in router configuration mode:

Command	Purpose

timers prefix-peer-timeout *value*
 	
Example :
 	
`switch(config-router)#`
`timers prefix-peer-timeout`
`120`
	Configures the timeout value for prefix peering. The range is from 0 to 1200 seconds. The default value is 30.
	
	Note For prefix peers, set the prefix peer timeout to be greater than the configured graceful restart timer. If the prefix peer timeout is greater than the graceful restart timer, a peer's route is retained during its restart. If the prefix peer timeout is less than the graceful restart timer, the peer's route is purged by the prefix peer timeout, which may occur before the restart is complete.

timers prefix-peer-wait interval
 	
Example :
 	
`switch(config-router)#`
`timers prefix-peer-wait 50`
	Configures the BGP prefix peering wait timer on a per-VRF basis or on the default VRF. You can use the timers prefix-peer-wait command to disable the peer prefix wait time so that there is no delay before BGP prefixes are inserted into the routing information base (RIB).
	
	The range of the interval is from 0 to 1200 seconds. The default value is 90 seconds.
	
	Note The timer is only applicable for BGP dynamic neighbors. It is only set when BGP is restarted or is coming up for the first time for dynamic BGP neighbors.

To configure the maximum number of peers, use the following command in neighbor configuration mode:

Command	Purpose

maximum-peers *value*
 	
Example :
 	
`switch(config-router-neighbor)#`
`maximum-peers 120`
	Configures the maximum number of peers for this prefix peering. The range is from 1 to 1000.

This example shows how to configure a prefix peering that accepts up to 10 peers:

```
switch(config)# router bgp 65536
switch(config-router)# timers prefix-peer-timeout 120
switch(config-router)# neighbor 10.100.200.0/24 remote-as 65536
switch(config-router-neighbor)# maximum-peers 10
switch(config-router-neighbor)# address-family ipv4 unicast
switch(config-router-neighbor-af)#
```

This example shows how to disable the peer prefix wait time:

```
switch(config)# router bgp 100
switch(config-router)# timers prefix-peer-wait 50
switch(config-router)#
```

Use the **show ip bgp neighbors** command to show the details of the configuration for that prefix peering with a list of the currently accepted instances and the counts of active, maximum concurrent, and total accepted peers.

Configuring BGP Authentication

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

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

Command	Purpose

password [**0** | **3** | **7**] *string*
 	
Example :
 	
`switch(config-router-neighbor)# password`
`BGPpassword`
	Configures an MD5 password for BGP neighbor sessions.

Resetting a BGP Session

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

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

Command	Purpose

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

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

Command	Purpose
clear bgp ip { **unicast** | **multicast** } *ip-address* **soft** { **in** | **out** } **Example** : `switch# clear bgp ip unicast 192.0.2.1 soft in`	Resets the BGP session without tearing down the TCP session.

Modifying the Next-Hop Address

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

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

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

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

Command	Purpose

next-hop-self
Example :
`switch(config-router-neighbor-af)#`
`next-hop-self`
	Uses the local BGP speaker address as the next-hop address in route updates. This command triggers an automatic soft clear or refresh of BGP neighbor sessions.

next-hop-third-party
Example :
`switch(config-router-neighbor-af)#`
`next-hop-third-party`
	Sets the next-hop address as a third-party address. Use this command for single-hop EBGP peers that do not have **next-hop-self** configured.

BGP Next-Hop Address Tracking

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

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

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

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

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

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

See the Configuring BGP Next-Hop Address Tracking section for more information.

Configuring Next-Hop Filtering

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

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

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

Command	Purpose

nexthop route-map *name*
 	
Example :
 	
`switch(config-router-af)# nexthop`
`route-map nextHopLimits`
	Specifies a route map to match the BGP next-hop route to. The name can be any case-sensitive, alphanumeric string up to 63 characters.

Controlling Reflected Routes Through Next-Hop-Self

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

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

Shrinking Next-Hop Groups When A Session Goes Down

You can configure BGP to shrink ECMP groups in an accelerated way when a session goes down.

This feature applies to the following BGP path failure events:

- Any single or multiple Layer 3 link failures

- BFD failure detections for BGP neighbors

- Administrative shutdown of BGP neighbors (using the shutdown command)

The accelerated handling of Layer 3 link failures is enabled by default and does not require a configuration command to be enabled.

To configure the accelerated handling of the last two events, use the following command in the router configuration mode:

Command	Purpose

neighbor-down fib-accelerate
 	
Example :
 	
`switch(config-router)#`
`neighbor-down fib-accelerate`
	Withdraws the corresponding next hop from all next-hop groups (ECMP groups and single next-hop routes) whenever a BGP session goes down.
	
	Note This command applies to both IPv4 and IPv6 address-family routes and is supported only in a BGP-only environment where all non-direct routes are installed by BGP.

Disabling Capabilities Negotiation

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

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

Command	Purpose

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

Configuring eBGP

This section includes the following topics:

Configuring eBGP Next-Hop Unchanged

You can configure eBGP to send routes to an eBGP multihop peer without changing the next-hop address. By default, the device changes the next-hop address of a BGP route to its own address when the device sends out a route.

	Command	Purpose

Step 1	**disable-connected-check**
	
	Example :
	
	`switch(config-router-neighbor)#`
	`disable-connected-check`
		Disables checking whether or not a single-hop eBGP peer is directly connected. You must manually reset the BGP sessions after using this command.

Step 2	**configure terminal** Example: `switch# configure terminal`	Enters global configuration mode.

	Command	Purpose
Step 3	**route-map** *route-map name* Example: `switch(config)# route-map route`	Enters route map configuration mode.

Step 4	**set ip next-hop unchanged**
	
	Example:
	
	`switch(config-route-map)# set ip next-hop`
	` unchanged`
		Configures the device to send BGP updates to the specified eBGP peer without modifying the next-hop address.

Step 5	**exit** Example: `switch(config-route-map)# exit`	Exits route map configuration mode.

This example shows how to set eBGP next-hop unchanged to send routes without changing the next-hop address:

```
switch# configure terminal
switch(config)# route-map route
switch(config-route-map)# set ip next-hop unchanged
switch(config-route-map)# exit
switch(config)#
```

Disabling eBGP Single-Hop Checking

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

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

Command	Purpose

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

Configuring eBGP Multihop

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

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

Command	Purpose

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

Configuring eBGP Routes in the Same Autonomous System

You can configure eBGP learned routes from a remote autonomous system (AS) to advertise to another eBGP peer in the same AS.

Note When route updates are sent between peers within the same AS number, they are dropped unless you enter the **allowas-in** command.

To disable AS peer checking, use the following command in neighbor configuration mode:

	Command	Purpose
Step 1	**router bgp** *autonomous-system-number* **Example** : `switch(config)# router bgp 64496`	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 2	**neighbor ipv4 remote-as** *as-number*
	
	Example :
	
	`switch (config-router)# neighbor`
	`209.165.201.1 remote-as 64497`
		Configures the specified address type 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 3	**address family ipv4 unicast**
	
	Example :
	
	`switch(config-router)# address family`
	`ip4 unicast`
		Enters neighbor address family configuration mode for the unicast specified address family.

Step 4	**disable-peer-as-check**
	
	Example :
	
	`switch(config-router-neighbor-af)#`
	`disable-peer-as-check`
		Disables AS checking so that routes are updated between peers in the same AS.

Step 5	**show bgp neighbor**
	
	Example :
	
	`switch(config-router-neighbor-af)# show`
	` bgp ipv4 unicast neighbors`
		Displays information about BGP peers.

This example shows how to display BGP peer information:

```
switch(config)# show bgp neighbor 1.222.222.2
bgp neighbor is 1.222.222.2, remote as 2222, ebgp link, peer index 1
bgp version 4, remote router id1.100.1.2 ####output trunicated####
for address family:ipv4 unicast
bgp table version 54, neighbor version 54
3 accepted paths consume 108 bytes of memory
10 sent paths
peer asn check is disabled

#####output omitted#####
```

Disabling a Fast External Failover

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

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

Command	Purpose
no fast-external-fallover **Example** : `switch(config-router)# no fast-external-fallover`	Disables a fast external failover for eBGP peers. This command is enabled by default.

Limiting the AS-path Attribute

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

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

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

Configuring Local AS Support

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

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

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

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

Configuring AS Confederations

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

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

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

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

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

Configuring an Autonomous System Path Containing Your Own Autonomous System

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

Before you begin

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

SUMMARY STEPS

1. **configure terminal**
2. **router bgp** *as-number*
3. **neighbor** *ip-address* **remote-as** *as-number*
4. **address-family** { **ipv4** | **ipv6**} { **multicast** | **unicast** }
5. [**no** | **default**] **allowas-in** [*allowas-in-cnt*]
6. **end**
7. (Optional) **show running-config bgp**
8. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**router bgp** *as-number*	Enters BGP mode and assigns the autonomous system number to the local BGP speaker. The *as-number* value range is from 1 to 65535.
Step 3	**neighbor** *ip-address* **remote-as** *as-number*	Enters neighbor configuration mode for BGP routing and configures the neighbor IP address.
Step 4	**address-family** { **ipv4**	**ipv6**} { **multicast**	**unicast** }	Enters router address family configuration for the specified address family.
Step 5	[**no**	**default**] **allowas-in** [*allowas-in-cnt*]	Enables the allowas-in feature for BGP and configures the number of occurrences of the AS number. For **allowas-in-cnt**, enter an integer between 1 and 10. By default, the number of occurrences of the AS number is set to 3.
Step 6	**end**	Exits router address family configuration mode.
Step 7	(Optional) **show running-config bgp**	Displays the BGP configuration.
Step 8	**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 the allowas-in feature for BGP and configure it for a unicast IPv4 address family:

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

Configuring Route Reflector

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

Before you begin

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

SUMMARY STEPS

1. **configure terminal**
2. **router bgp** *as-number*
3. **cluster-id** *cluster-id*
4. **address-family** {**ipv4** | **ipv6**} { **unicast** | **multicast**}
5. (Optional) **client-to-client reflection**
6. **exit**
7. **neighbor** *ip-address* **remote-as** *as-number*
8. **address-family** { **ipv4** | **ipv6** }{ **unicast** | **multicast** }
9. **route-reflector-client**
10. (Optional) **show bgp ip** { **unicast** | **multicast** } **neighbors**
11. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	**router bgp** *as-number* **Example:**	Enters BGP mode and assigns the autonomous system number to the local BGP speaker.

	Command or Action	Purpose
	switch(config)# router bgp 65536 switch(config-router)#	
Step 3	**cluster-id** *cluster-id* **Example:** switch(config-router)# cluster-id 192.0.2.1	Configures the local router as one of the route reflectors that serve the cluster. You specify a cluster ID to identify the cluster. This command triggers an automatic soft clear or refresh of BGP neighbor sessions.
Step 4	**address-family** {**ipv4** | **ipv6**} { **unicast** | **multicast**} **Example:** switch(config-router)# address-family ipv4 unicast switch(config-router-af)#	Enters router address family configuration mode for the specified address family.
Step 5	(Optional) **client-to-client reflection** **Example:** switch(config-router-af)# client-to-client reflection	Configures client-to-client route reflection. This feature is enabled by default. This command triggers an automatic soft clear or refresh of BGP neighbor sessions.
Step 6	**exit** **Example:** switch(config-router-neighbor)# exit switch(config-router)#	Exits router address configuration mode.
Step 7	**neighbor** *ip-address* **remote-as** *as-number* **Example:** switch(config-router)# neighbor 192.0.2.10 remote-as 65536 switch(config-router-neighbor)#	Configures the IP address and AS number for a remote BGP peer.
Step 8	**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 specified address family.
Step 9	**route-reflector-client** **Example:** switch(config-router-neighbor-af)# route-reflector-client	Configures the switch as a BGP route reflector and configures the neighbor as its client. This command triggers an automatic notification and session reset for the BGP neighbor sessions.
Step 10	(Optional) **show bgp ip** { **unicast** | **multicast** } **neighbors** **Example:** switch(config-router-neighbor-af)# show bgp ip unicast neighbors	Displays the BGP peers.
Step 11	(Optional) **copy running-config startup-config** **Example:** switch(config-router-neighbor-af)# copy running-config startup-config	Saves this configuration change.

Example

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

```
switch(config)# router bgp 65536
switch(config-router)# neighbor 192.0.2.10 remote-as 65536
switch(config-router-neighbor)# address-family ip unicast
switch(config-router-neighbor-af)# route-reflector-client
switch(config-router-neighbor-af)# copy running-config startup-config
```

Configuring Route Dampening

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

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

Command	Purpose
dampening [{ *half-life reuse-limit suppress-limit max-suppress-time* | **route-map** *map-name* }] **Example** : `switch(config-router-af)# dampening route-map bgpDamp`	Disables capabilities negotiation. The parameter values are as follows: • half-life—The range is from 1 to 45. • reuse-limit—The range is from 1 to 20000. • suppress-limit—The range is from 1 to 20000. • max-suppress-time—The range is from 1 to 255.

Configuring Route Deletion Delay

When routes are deleted from the hardware faster than the BGP neighbor can remove them from the next-hop list, the traffic that is received for those routes is lost. You can delay route deletion from the hardware by configuring a route deletion interval.

This configuration is not applied when a subset of next hops is being removed from the route or when a backup path exists for the route. It applies only to a route that is completely getting deleted.

Command	Purpose

route delete dampen interval [*seconds*]
 	
Example :
 	
`switch(config)# route delete dampen interval 20`
`switch(config)#`
	Delays route deletion from the hardware.
	
	The interval can range from 1 second to 30 seconds.
	
	The default value is 0 seconds.

Configuring Load Sharing and ECMP

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

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

Command	Purpose
maximum-paths [**ibgp**] *maxpaths* **Example** : switch(config-router-af)# maximum-paths 12	Configures the maximum number of equal-cost paths for load sharing. The range is from 1 to 16. The default is 1.

Configuring Maximum Prefixes

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

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

Command	Purpose
maximum-prefix *maximum* [*threshold*] [**restart** *time* | **warming-only**] **Example** : switch(config-router-neighbor-af)# maximum-prefix 12	Configures the maximum number of prefixes from a peer. The parameter ranges are as follows: • *maximum* —The range is from 1 to 300000. • *Threshold* —The range is from 1 to 100 percent. The default is 75 percent. • *time* —The range is from 1 to 65535 minutes. This command triggers an automatic notification and session reset for the BGP neighbor sessions if the prefix limit is exceeded.

Configuring Dynamic Capability

You can configure dynamic capability for a BGP peer.

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

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

Configuring Aggregate Addresses

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

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

Command	Purpose
aggregate-address *ip-prefix/length* [**as-set**] [**summary-only**] [**advertise-map** *map-name*] [**attribute-map** *map-name*] [**suppress-map** *map-name*] **Example** : `switch(config-router-af)# aggregate-address 192.0.2.0/8 as-set`	Creates an aggregate address. The path advertised for this route is an autonomous system set that consists of all elements contained in all paths that are being summarized: • The **as-set** keyword generates autonomous system set path information and community information from contributing paths. • The **summary-only** keyword filters all more specific routes from updates. • The **advertise-map** keyword and argument specify the route map used to select attribute information from selected routes. • The **attribute-map** keyword and argument specify the route map used to select attribute information from the aggregate. • The **suppress-map** keyword and argument conditionally filters more specific routes.

Suppressing BGP Routes

You can configure Cisco NX-OS to advertise newly learned BGP routes only after these routes are confirmed by the Forwarding Information Base (FIB) and programmed in the hardware. After the routes are programmed, subsequent changes to these routes do not require this hardware-programming check. BGP route suppression is not enabled by default.

Note When you enable fib-suppression on the switch for routes that are not programmed locally in the hardware because of hardware table exhaustion, BGP advertises these failed routes even though they are not programmed locally in the hardware.

To suppress BGP routes, use the following command in the router configuration mode:

Command	Purpose
suppress-fib-pending **Example** : `switch(config-router)# suppress-fib-pending`	Suppresses newly learned BGP routes (IPv4 or IPv6) from being advertised to downstream BGP neighbors until the routes have been programmed in the hardware.

Configuring BGP Conditional Advertisement

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

- Advertise map—Specifies the conditions that the route must match before BGP considers the conditional advertisement. This route map can contain any appropriate match statements.

- Exist map or nonexist map—Defines the prefix that must exist in the BGP table before BGP propagates a route that matches the advertise map. The nonexist map defines the prefix that must not exist in the BGP table before BGP propagates a route that matches the advertise map. BGP processes only the permit statements in the prefix list match statements in these route maps.

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

Before you begin

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

SUMMARY STEPS

1. **configure terminal**
2. **router bgp** *as-number*
3. **neighbor** *ip-address* **remote-as** *as-number*
4. **address-family** { **ipv4** | **ipv6** } { **unicast** | **multicast** }
5. **advertise-map** *adv-map* { **exist-map** *exist-rmap* | **non-exist-map** *nonexist-rmap* }
6. (Optional) **show ip bgp neighbor**
7. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	**router bgp** *as-number*
	
	Example:
	
	`switch(config)# router bgp 65536`
	`switch(config-router)#`
		Enters BGP mode and assigns the autonomous system number to the local BGP speaker.

Step 3	**neighbor** *ip-address* **remote-as** *as-number*
	
	Example:
	
	`switch(config-router)# neighbor 192.168.1.2`
	`remote-as 65537`
	`switch(config-router-neighbor)#`
		Places the router in neighbor configuration mode for BGP routing and configures the neighbor IP address.

Step 4	**address-family** { **ipv4**	**ipv6** } { **unicast**	**multicast** }
			
			Example:
			
			`switch(config-router-neighbor)# address-family ipv4`
			`  multicast`
			`switch(config-router-neighbor-af)#`
				Enters address family configuration mode.

	Command or Action	Purpose
Step 5	**advertise-map** *adv-map* { **exist-map** *exist-rmap* | **non-exist-map** *nonexist-rmap* } **Example:** `switch(config-router-neighbor-af)# advertise-map advertise exist-map exist`	Configures BGP to conditionally advertise routes based on the two configured route maps: • *adv-map*—Specifies a route map with **match** statements that the route must pass before BGP passes the route to the next route map. The adv-map is a case-sensitive, alphanumeric string up to 63 characters. • *exist-rmap*—Specifies a route map with **match** statements for a prefix list. A prefix in the BGP table must match a prefix in the prefix list before BGP will advertise the route. The exist-rmap is a case-sensitive, alphanumeric string up to 63 characters. • *nonexist-rmap*—Specifies a route map with **match** statements for a prefix list. A prefix in the BGP table must not match a prefix in the prefix list before BGP will advertise the route. The nonexist-rmap is a case-sensitive, alphanumeric string up to 63 characters.
Step 6	(Optional) **show ip bgp neighbor** **Example:** `switch(config-router-neighbor-af)# show ip bgp neighbor`	Displays information about BGP and the configured conditional advertisement route maps.
Step 7	(Optional) **copy running-config startup-config** **Example:** `switch(config-router-neighbor-af)# copy running-config startup-config`	Saves this configuration change.

Example

This example shows how to configure BGP conditional advertisement:

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

Configuring Route Redistribution

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

Before you begin

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

SUMMARY STEPS

1. **configure terminal**
2. **router bgp** *as-number*
3. **address-family** { **ipv4** | **ipv6** } { **unicast** | **multicast** }
4. **redistribute** { **direct** | { **eigrp** | **ospf** | **ospfv3** | **rip** } *instance-tag* | **static** } **route-map** *map-name*
5. (Optional) **default-metric** *value*
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	**router bgp** *as-number*
	Example:
	`switch(config)# router bgp 65536`
	`switch(config-router)#`
		Enters BGP mode and assigns the autonomous system number to the local BGP speaker.

Step 3	**address-family** { **ipv4**	**ipv6** } { **unicast**	**multicast** }
			Example:
			`switch(config-router)# address-family ipv4 unicast`
			`switch(config-router-af)#`
				Enters address family configuration mode.

Step 4	**redistribute** { **direct**	{ **eigrp**	**ospf**	**ospfv3**	**rip** } *instance-tag*	**static** } **route-map** *map-name*
						Example:
						`switch(config-router-af)# redistribute eigrp 201`
						`route-map Eigrpmap`
							Redistributes routes from other protocols into BGP. See the Configuring Route Maps section for more information about route maps.

Step 5	(Optional) **default-metric** *value* **Example:** `switch(config-router-af)# default-metric 33`	Generates a default route into BGP.
Step 6	(Optional) **copy running-config startup-config** **Example:**	Saves this configuration change.

Command or Action	Purpose
switch(config-router-af)# copy running-config startup-config	

Example

This example shows how to redistribute EIGRP into BGP:

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

Disabling BGP Dampening with Redistribution

When an IGP metric of routes redistributed into BGP changes, BGP has internal dampening that prevents an immediate route update to the BGP peers. It affects how BGP handles IGP metric changes reported for redistributed routes. BGP dampens these changes through a batch process with a 10-minute delay. This command enables you to adjust that delay or remove it altogether for a quicker response to these changes.

SUMMARY STEPS

1. **configure terminal**
2. **router bgp** *as-number*
3. **address-family** { **ipv4** | **ipv6** } { **unicast** | **multicast** }
4. **dampen-igp-metric** *seconds*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	**router bgp** *as-number* **Example:** switch(config)# router bgp 65536 switch(config-router)#	Enters BGP mode and assigns the autonomous system number to the local BGP speaker.
Step 3	**address-family** { **ipv4**	**ipv6** } { **unicast**	**multicast** } **Example:** switch(config-router)# address-family ipv4 unicast switch(config-router-af)#	Enters address family configuration mode.
Step 4	**dampen-igp-metric** *seconds* **Example:**	Configures dampening of IGP metric-related changes for redistributed routes.

Command or Action	Purpose
`switch(config-router-af)# dampen-igp-metric 100`	

Example

This example shows how to configure BGP dampening for redistributed routes:

```
switch# configure terminal
switch(config)# router bgp 100
switch(config-router)# address-family ipv4 unicast
switch(config-router-af)# dampen-igp-metric 100
switch(config-router-af)#
```

Configuring Multiprotocol BGP

You can configure MP-BGP to support multiple address families, including IPv4 unicast and multicast routes.

Before you begin

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

SUMMARY STEPS

1. **configure terminal**
2. **router bgp** *as-number*
3. **neighbor** *ip-address* **remote-as** *as-number*
4. **address-family** { **ipv4** | **ipv6** } { **unicast** | **multicast** }
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	**router bgp** *as-number*
	Example:
	`switch(config)# router bgp 65536`
	`switch(config-router)#`
		Enters BGP mode and assigns the autonomous system number to the local BGP speaker

Step 3	**neighbor** *ip-address* **remote-as** *as-number*
	Example:
	`switch(config-router)# neighbor 192.168.1.2`
	`remote-as 65537`
	`switch(config-router-neighbor)#`
		Places the router in neighbor configuration mode for BGP routing and configures the neighbor IP address.

	Command or Action	Purpose
Step 4	**address-family** { **ipv4**	**ipv6** } { **unicast**	**multicast** } **Example:** `switch(config-router-neighbor)# address-family ipv4 multicast` `switch(config-router-neighbor-af)#`	Enters address family configuration mode.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch(config-router-neighbor-af)# copy running-config startup-config`	Saves this configuration change.

Example

This example shows how to enable advertising and receiving IPv4 routes for multicast RPF for a neighbor:

```
switch# configure terminal
switch(config)# interface ethernet 1/2
switch(config-if)# ipv6 address 2001:0DB8::1
switch(config-if)# router bgp 65536
switch(config-router)# neighbor 192.168.1.2 remote-as 35537
switch(config-router-neighbor)# address-family ipv4 multicast
switch(config-router-neighbor-af)# exit
switch(config-router-neighbor)# address-family ipv6 multicast
switch(config-router-neighbor-af)# copy running-config startup-config
```

Configuring BGP Extended Community Site of Origin

To configure BGP extended community site of origin, use the following commands

Command	Purpose
router bgp *as-number* **Example** : `switch(config)# router bgp 1` `switch(config-router)#`	Configures a BGP routing process and enters router configuration mode.
vrf *vrf-name* **Example:** `switch(config-router)# vrf 450` `switch(config-router-vrf)#`	Enters the router VRF configuration mode and associates this BGP instance with a VRF.
neighbor ip-address remote-as *as-number* **Example:** `switch(config-router-vrf)# neighbor 1::1 remote-as 2` `switch(config-router-vrf-neighbor)#`	Configures the IP address and AS number for a remote BGP peer.

Command	Purpose

address-family { **ipv4** | **ipv6** } { **multicast** | **unicast** }
 	
Example :
 	
`switch(config-router-vrf-neighbor)# address-family ipv6 unicast`
`switch(config-router-vrf-neighbor-af)#`
	Enters global address family configuration mode for the specified address family.

soo *value* Example: `switch(config-router-vrf-neighbor-af)# soo 22:14`	Configures the site of origin BGP extended community value. The value is in one of the following formats: • asn:number • IP address:number The number range is from 0 to 65535 for a 2-byte ASN or from 0 to 4294967295 for a 4-byte ASN.

Tuning BGP

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

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

Command	Purpose

bestpath [**always-compare-med** | **compare-routerid** | **igp-metric ignore** | **med** { **missing-as-worst** | **non-deterministic** } | **as-path multipath-relax**]
 	
Example :
 	
`switch(config-router)# bestpath always-compare-med`
`switch(config-router)# bestpath as-path multipath-relax`
	Modifies the best-path algorithm. The optional parameters are as follows:
	
	• always-compare-med —Compares MED on paths from different autonomous systems.
	
	• compare-routerid—Compares the router IDs for identical eBGP paths.
	
	• igp-metric ignore—Ignores the Interior Gateway Protocol (IGP) metric for next hop during best-path selection. This option is supported beginning with Cisco NX-OS Release 9.2(2).
	
	• med missing-as-worst—Handles a missing MED as the highest MED.
	
	• med non-deterministic —Does not always select the best MED path from among the paths from the same autonomous system.
	
	• as-path multipath-relax —Allows the switch to handle the paths received from different AS numbers for multipath, if their AS-path lengths are the same and other multipath conditions are met.

Command	Purpose
enforce-first-as **Example:** `switch(config-router)# enforce-first-as`	Enforces the neighbor autonomous system to be the first AS number listed in the AS_path attribute for eBGP.
log-neighbor-changes **Example** : `switch(config-router)# log-neighbor-changes`	Generates a system message when a neighbor changes state. **Note** To suppress neighbor status change messages for a specific neighbor, you can use the log-neighbor-changes disable command in neighbor configuration mode.
router-id *id* **Example** : `switch(config-router)# router-id 209.165.20.1`	Manually configures the router ID for this BGP speaker.
timers [**bestpath-delay** *delay* | **bgp** *keepalive holdtime* | **prefix-peer-timeout** *timeout*] **Example** : `switch(config-router)# timers bgp 90 270`	Sets the BGP timer values. The optional parameters are as follows: • *delay* —Initial best-path timeout value after a restart. The range is from 0 to 3600 seconds. The default value is 300. • *keepalive* —BGP session keepalive time. The range is from 0 to 3600 seconds. The default value is 60. • *holdtime* —BGP session hold time.The range is from 0 to 3600 seconds. The default value is 180. • *timeout* —Prefix peer timeout value. The range is from 0 to 1200 seconds. The default value is 30. You must manually reset the BGP sessions after configuring this command.

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

Command	Purpose
distance *ebgp-distance ibgp distance local-distance* **Example** : `switch(config-router-af)# distance 20 100 200`	Sets the administrative distance for BGP. The range is from 1 to 255. The defaults are as follows: • eBGP distance—20. • iBGP distance—200. • local distance—220. Local distance is the administrative distance used for aggregate discard routes when they are installed in the RIB.

Command	Purpose

bestpath all-paths-ecmp
 	
Example :
 	
`switch(config-router-af)# bestpath`
`all-paths-ecmp`
	Treats all paths as ECMP during best path calculation

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

Command	Purpose

description *string*
 	
Example :
 	
`switch(config-router-neighbor)#`
`description main site`
	Sets a descriptive string for this BGP peer. The string can be up to 80 alphanumeric characters.

low-memory exempt
 	
Example :
 	
`switch(config-router-neighbor)#`
`low-memory exempt`
	Exempts this BGP neighbor from a possible shutdown due to a low memory condition.

transport connection-mode passive
 	
Example:
 	
`switch(config-router-neighbor)#`
`transport connection-mode passive`
	Allows a passive connection setup only. This BGP speaker does not initiate a TCP connection to a BGP peer. You must manually reset the BGP sessions after configuring this command.

remove-private-as
 	
Example :
 	
`switch(config-router-neighbor)#`
`remove-private-as`
	Removes private AS numbers from outbound route updates to an eBGP peer. This command triggers an automatic soft clear or refresh of BGP neighbor sessions.
	
	Note See the Guidelines and Limitations for Advanced BGP section for additional information on this command.

update-source *interface-type number*
 	
Example :
 	
`switch(config-router-neighbor)#`
`update-source ethernet 1/2`
	Configures the BGP speaker to use the source IP address of the configured interface for BGP sessions to the peer. This command triggers an automatic notification and session reset for the BGP neighbor sessions.

log-neighbor-changes [**disable**]
 	
Example :
 	
`switch(config-router-neighbor)#`
`log-neighbor-changes disable`
	Generates a system message when this specific neighbor changes state.
	
	The disable option suppresses neighbor status change messages for this specific neighbor.

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

Command	Purpose
suppress-inactive **Example** : `switch(config-router-neighbor-af)# suppress-inactive`	Advertises the best (active) routes only to the BGP peer. This command triggers an automatic soft clear or refresh of BGP neighbor sessions.
default-originate [**route-map** *map-name*] **Example** : `switch(config-router-neighbor-af)# default-originate`	Generates a default route to the BGP peer.
filter-list *list-name* { **in** | **out** } **Example:** `switch(config-router-neighbor-af)# filter-list BGPFilter in`	Applies an AS_path filter list to this BGP peer for inbound or outbound route updates. This command triggers an automatic soft clear or refresh of BGP neighbor sessions.
prefix-list *list-name* { **in** | **out** } **Example:** `switch(config-router-neighbor-af)# prefix-list PrefixFilter in`	Applies a prefix list to this BGP peer for inbound or outbound route updates. This command triggers an automatic soft clear or refresh of BGP neighbor sessions.
send-community **Example** : `switch(config-router-neighbor-af)# send-community`	Sends the community attribute to this BGP peer. This command triggers an automatic soft clear or refresh of BGP neighbor sessions.
send-extcommunity **Example** : `switch(config-router-neighbor-af)# send-extcommunity`	Sends the extended community attribute to this BGP peer. This command triggers an automatic soft clear or refresh of BGP neighbor sessions.
[no | default] as-override **Example:** `switch(config-router-neighbor-af)# as-override`	**no** - (Optional) Disables the command. **default** - (Optional) Moves the command to its default mode. **as-override** - While sending updates to eBGP peer, replaces in the *path* attribute all occurrences of the peer's AS number with the local AS number.

Configuring Virtualization

Before you begin

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

SUMMARY STEPS

1. **configure terminal**

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** switch# configure terminal switch(config)#	Enters configuration mode.
Step 2	**vrf context** *vrf-name* **Example:** switch(config)# vrf context RemoteOfficeVRF switch(config-vrf)#	Creates a new VRF and enters VRF configuration mode.
Step 3	**exit** **Example:** switch(config-vrf)# exit switch(config)#	Exits VRF configuration mode.
Step 4	**router bgp** *as-number* **Example:** switch(config)# router bgp 65536 switch(config-router)#	Creates a new BGP process with the configured autonomous system number.
Step 5	**vrf** *vrf-name* **Example:** switch(config-router)# vrf RemoteOfficeVRF switch(config-router-vrf)#	Enters the router VRF configuration mode and associates this BGP instance with a VRF.
Step 6	**neighbor** *ip-address* **remote-as** *as-number* **Example:** switch(config-router-vrf)# neighbor 209.165.201.1 remote-as 65536 switch(config-router--vrf-neighbor)#	Configures the IP address and AS number for a remote BGP peer.
Step 7	(Optional) **bestpath as-path multipath-relax** **Example:** switch(config-router-vrf)# bestpath as-path multipath-relax	Allows the switch to treat paths received from different autonomous systems for multipath, if their autonomous path lengths are the same and other multipath conditions are met.

	Command or Action	**Purpose**
Step 8	(Optional) **copy running-config startup-config** **Example:** `switch(config-router-neighbor)# copy running-config startup-config`	Saves this configuration change.

Example

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

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

Configuring BMP

You can configure BMP on the device.

Before you begin

You must enable BGP.

SUMMARY STEPS

1. **configure terminal**
2. **router bgp** *as-number*
3. **bmp server server-number**
4. **address** *ip-address* **port-number** *port-number*
5. **description** *string*
6. **initial-refresh** { **skip** | **delay** *time* }
7. **initial-delay** *time*
8. **stats-reporting-period** *time*
9. **shutdown**
10. **neighbor** *ip-address* **remote-as** *as-number*
11. **bmp-activate-server** *server-number*
12. (Optional) **show bgp bmp server** [*server-number*][*detail*]
13. (Optional) **copy running-config startup-config**

DETAILED STEPS

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

	Command or Action	Purpose
	`switch# configure terminal` `switch(config)#`	

Step 2	**router bgp** *as-number*
	Example:
	`switch(config)# router bgp 200`
	`switch(config-router)#`
		Enters BGP mode and assigns the autonomous system number to the local BGP speaker.

Step 3	**bmp server** *server-number*
	Example:
	`switch(config-router)# bmp server 1`
	`switch(config-router)#`
		Configures the BMP server to which BGP should send information. The server number is used as a key.
		Note You can configure up to two BMP servers

Step 4	**address** *ip-address* **port-number** *port-number*
	Example:
	`switch(config-router)# address 10.1.1.1`
	`port-number 2000`
		Configures the IPv4 or IPv6 address of the host and the port number on which the BMP speaker connects to the BMP server.

Step 5	**description** *string* **Example:** `switch(config-router)# description BMPserver1`	Configures the BMP server description. You can enter up to 256 alphanumeric characters.
Step 6	**initial-refresh** { **skip**	**delay** *time* } **Example:** `switch(config-router)# initial-refresh delay 100`	Configures the option to send a route refresh when BGP is converged and the BMP server connection is established later. The skip option specifies to not send a route refresh if the BMP server connection comes up later. The delay option specifies the time in seconds after which the route refresh should be sent. The range is from 30 to 720 seconds, and the default value is 30 seconds.
Step 7	**initial-delay** *time* **Example:** `switch(config-router)# initial-delay 120`	Configures the delay after which a connection is attempted to the BMP server. The range is from 30 to 720 seconds, and the default value is 45 seconds.

Step 8	**stats-reporting-period** *time*
	Example:
	`switch(config-router)#`
	`stats-reporting-period 50`
		Configures the time interval in which the BMP server receives the statistics report from BGP neighbors. The range is from 30 to 720 seconds, and the default is disabled.

Step 9	**shutdown** **Example:** `switch(config-router)# shutdown`	Disables the connection to the BMP server.
Step 10	**neighbor** *ip-address* **remote-as** *as-number* **Example:**	Enters neighbor configuration mode for BGP routing and configures the neighbor IP address.

	Command or Action	Purpose
	switch(config-router)# neighbor 192.168.1.2 remote-as 65535 switch(config-router-neighbor)#	
Step 11	**bmp-activate-server** *server-number* **Example:** switch(config-router-neighbor)# bmp-activate-server 1	Configures the BMP server to which a neighbor's information should be sent.
Step 12	(Optional) **show bgp bmp server** [*server-number*][*detail*] **Example:** switch(config-router-neighbor)# show bgp bmp server	Displays BMP server information.
Step 13	(Optional) **copy running-config startup-config** **Example:** switch(config-router-neighbor)# copy running-config startup-config	Saves this configuration change.

Verifying the Advanced BGP Configuration

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

Command	Purpose
show bgp all [**summary**] [**vrf** *vrf-name*]	Displays the BGP information for all address families.
show bgp convergence [**vrf** *vrf-name*]	Displays the BGP information for all address families.
show bgp ip {**unicast**} [*ip-address*] **community** {**regexp** *expression* | [**community**] [**no-advertise**] [**no-export**] [**no-export-subconfed**]} [**vrf** *vrf-name*]	Displays the BGP routes that match a BGP community.
show bgp [**vrf** *vrf-name*] **ip** {**unicast**} [*ip-address*] **community-list** list-name [**vrf** *vrf-name*]	Displays the BGP routes that match a BGP community list.
show bgp ip {**unicast**} [*ip-address*] **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 ip {**unicast**} [ip-address] **extcommunity-list** *list-name* [**exact-match**] [**vrf** *vrf-name*]	Displays the BGP routes that match a BGP extended community list.
show bgp ip {**unicast**} [*ip-address*] {**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.

Command	Purpose
show bgp ip {**unicast**} [*ip-address*] **history-paths** [**regexp** *expression*] [**vrf** *vrf-name*]	Displays the BGP route history paths.
show bgp ip {**unicast**} [*ip-address*] **filter-list** *list-name* [**vrf** *vrf-name*]	Displays the information for the BGP filter list.
show bgp ip {**unicast**} [*ip-address*] **neighbors** [*ip-address*] [**vrf** *vrf-name*]	Displays the information for BGP peers. Use the **clear bgp neighbors** command to clear these neighbors.
show bgp ip {**unicast**} [*ip-address*] {**nexthop** | **nexthop-database**} [**vrf** *vrf-name*]	Displays the information for the BGP route next hop.
show bgp paths	Displays the BGP path information.
show bgp ip {**unicast**} [*ip-address*] **policy name** [**vrf** *vrf-name*]	Displays the BGP policy information. Use the **clear bgp policy** command to clear the policy information.
show bgp ip {**unicast**} [*ip-address*] **prefix-list** *list-name* [**vrf** *vrf-name*]	Displays the BGP routes that match the prefix list.
show bgp ip {**unicast**} [*ip-address*] **received-paths** [**vrf** *vrf-name*]	Displays the BGP paths stored for soft reconfiguration.
show bgp ip {**unicast**} [*ip-address*] **regexp** *expression* [**vrf** *vrf-name*]	Displays the BGP routes that match the AS_path regular expression.
show bgp ip {**unicast**} [*ip-address*] **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 bgp *options*	Displays the BGP status and configuration information. This command has multiple options. See the Cisco Nexus 3000 Series Command Reference for more information.
show ip mbgp *options*	Displays the BGP status and configuration information. This command has multiple options. See the Cisco Nexus 3000 Series Command Reference for more information.
show running-configuration bgp	Displays the current running BGP configuration.

Displaying BGP Statistics

To display BGP statistics, use the following commands:

Command	Purpose
show bgp ip {**unicast**} [*ip-address*] **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 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.

Related Topics

The following topics can give more information on BGP:

- Configuring Advanced BGP
- Configuring Route Policy Manager

Additional References

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

Related Documents

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

MIBs

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