Configuring Advanced BGP

This chapter contains the following sections:

Finding Feature Information

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

Information About 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 multicast routes and multiple Layer 3 protocol address families. BGP uses TCP as a reliable transport protocol to create TCP sessions with other BGP-enabled devices 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 configuration 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.

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

eBGP

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

You should use loopback interfaces for establishing eBGP peering sessions because loopback interfaces are less susceptible to interface flapping. An interface flap occurs when the interface is administratively brought up or down because of a failure or maintenance issue.

BGP Next Hop Unchanged

In an eBGP session, by default, the router changes the next-hop attribute of a BGP route to its own address when the router sends out a route. The BGP next-hop unchanged feature allows BGP to send an update to an eBGP multihop peer with the next-hop attribute unchanged.

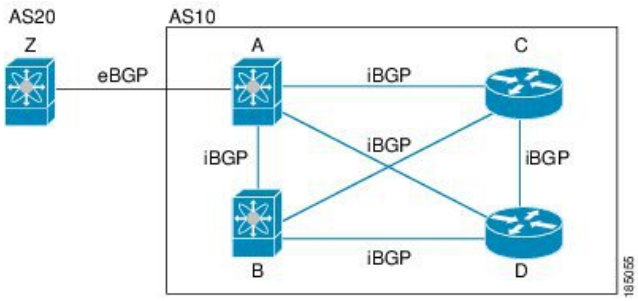
By default, BGP puts itself as the next hop when announcing to an eBGP peer. When you enter the **set ip next-hop unchanged** command for an outbound route map that is configured for an eBGP peer, it propagates the received next hop to the eBGP peer.

iBGP

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

The 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.

For single-hop iBGP peers with update-source configured under neighbor configuration mode, the peer supports fast external fall-over.

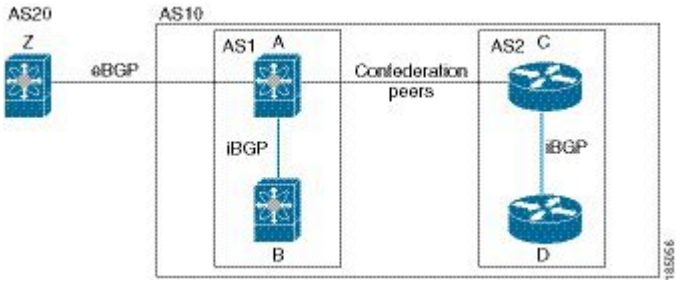
Note You should configure a separate interior gateway protocol in the iBGP network.

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 figure shows the BGP network, 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.

Route Reflector

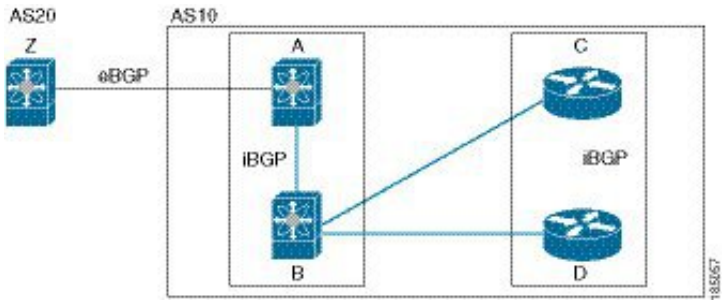
You can alternately reduce the iBGP mesh by using a route reflector configuration where route reflectors pass learned routes to neighbors so that all iBGP peers do not need to be fully meshed.

The figure below shows a simple iBGP configuration with four meshed iBGP speakers (routers A,B,C, and D.) Without these route reflectors, when router A receives a route from an external neighbor, it advertise 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 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 that are 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 attempts a new session to the peer without capabilities negotiation if you have configured the address family as IPv4. Any other multiprotocol configuration (such as IPv6) requires capabilities negotiation.

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

In Cisco NX-OS releases prior to 6.1, BGP selects only one of these multiple paths as the best path and advertises the path to the BGP peers. Beginning with Cisco NX-OS Release 6.1, BGP supports sending and receiving multiple paths per prefix and advertising such paths.

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

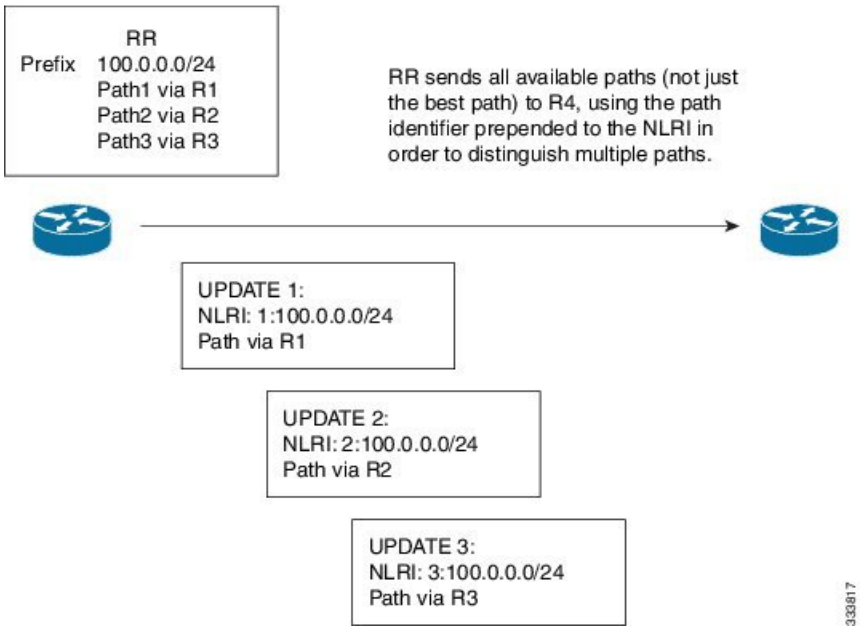
Note When you configure a route reflector for iBGP multipath, and the route reflector advertises the selected best path to its peers, the next hop for the path is not modified.

BGP Additional Paths

In Cisco NX-OS releases prior to 6.1, only one BGP best path is advertised, and the BGP speaker accepts only one path for a given prefix from a given peer. If a BGP speaker receives multiple paths for the same prefix within the same session, it uses the most recent advertisement.

Beginning with Cisco NX-OS Release 6.1, BGP supports the additional paths feature, which allows the BGP speaker to propagate and accept multiple paths for the same prefix without the new paths replacing any previous ones. This feature allows BGP speaker peers to negotiate whether they support advertising and receiving multiple paths per prefix and advertising such paths. A special 4-byte path ID is added to the network layer reachability information (NLRI) to differentiate multiple paths for the same prefix sent across a peer session. The following figure illustrates the BGP additional paths capability.

Figure 4: BGP Route Advertisement with the Additional Paths Capability



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.

Summary entry is created in the BGP table when **aggregate-address** command is configured, though it will not be eligible for advertisement until a subset of the aggregate is found in the table.

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.

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 Routing Information Base (RIB) that may affect BGP next-hop reachability.

BGP receives notifications from the RIB when the 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 Interior Gateway Protocol (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 next-hop reachability, such as the loss of next hops resulting in a switchover to a different path. A change in the IGP metric for a next hop resulting in a switchover to a different path can also be considered a critical event.
- Non-critical events are related to next hops being added without affecting the best path or changing the IGP metric to a single next hop.

Note Critical and non-critical events can be configured individually on a per address family basis. For more information on address families, see the "Configuring MPLS Layer 3 VPNs" chapter in the *Cisco Nexus 7000 Series NX-OS MPLS Configuration Guide*.

Route Redistribution

You can configure BGP to redistribute static routes or routes from other protocols. You must configure a route map with the redistribution to control which routes are passed into BGP. A route map allows you to filter routes based on attributes such as the destination, origination protocol, route type, route tag, and so on.

Prior to Cisco NX-OS Release 5.2(1), when you redistribute BGP to IGP, iBGP is redistributed as well. To override this behavior, you must insert an additional deny statement into the route map. Beginning with Cisco NX-OS Release 5.2(1), redistribution varies as follows:

- In a non-MPLS VPN scenario, iBGP is not redistributed to IGP by default.
- In an MPLS VPN scenario (route distinguisher configured under a VRF), iBGP is redistributed to IGP by default.

You can use route maps to override the default behavior in both scenarios, but be careful when doing so as incorrect use of route maps can result in network loops. The following examples show how to use route maps to change the default behavior.

You can change the default behavior for scenario 1 by modifying the route map as follows:

```
route-map foo permit 10
   match route-type internal
router ospf 1
   redistribute bgp 100 route-map foo
```

Similarly, you can change the default behavior for scenario 2 by modifying the route map as follows:

```
route-map foo deny 10
  match route-type internal
router ospf 1
   vrf bar
     redistribute bgp 100 route-map foo
```

The default route should be redistributed into BGP or advertised to peers only when **default-information originate** is configured for an Address Family where the command is supported.

BGP should withdraw the default route on removal of default-information originate if it was already advertised. Also, the redistributed path should be removed for the default route.

You can delete the redistributed path for default route using the following command:

no default-information originate

BGP Support for Importing Routes from Default VRF

You can import IP prefixes from the global routing table (the default VRF) into any other VRF by using an import policy. The VRF import policy uses a route map to specify the prefixes to be imported into a VRF. The policy can import IPv4 and IPv6 unicast prefixes.

You can configure the maximum number of prefixes that can be imported from the default VRF.

Note Routes in the BGP default VRF can be imported directly. Any other routes in the global routing table should be redistributed into BGP first.

BGP Support for Exporting Routes to Default VRF

You can export IP prefixes to the default VRF (global routing table) from any other VRF using an export policy. The VRF export policy leaks a VRF route into default VRF BGP table, which will then be installed in the IPv4/IPv6 routing table. The VRF export policy uses a route map to specify the prefixes to be exported to the default VRF. The policy can export IPv4 and IPv6 unicast prefixes.

You can configure the maximum number of prefixes that can be exported to the default VRF to prevent the routing table from being overloaded.

BFD

This feature supports bidirectional forwarding detection (BFD) for IPv4 only. 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 peers and iBGP single-hop peers. Configure the update-source option in neighbor configuration mode for iBGP single-hop peers using BFD.

Note BFD is not supported on other iBGP peers or for multihop eBGP peers.

See the *Cisco Nexus 7000 Series NX-OS Interfaces Configuration Guide* 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 (MED) attribute and the router ID.

Multiprotocol BGP

BGP on Cisco NX-OS supports multiple address families. Multiprotocol BGP (MP-BGP) carries different sets of routes depending on the address family. For example, BGP can carry one set of routes for IPv4 unicast routing, one set of routes for IPv4 multicast routing, and one set of routes for IPv6 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.

Note Beginning with Cisco NX-OS Release 6.2(8), 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).

Graceful Restart and High Availability

Cisco NX-OS supports nonstop forwarding and graceful restart for BGP.

You can use nonstop forwarding (NSF) for BGP to forward data packets along known routes in the Forward Information Base (FIB) while the BGP routing protocol information is being restored following a failover. With NSF, BGP peers do not experience routing flaps. During a failover, the data traffic is forwarded through intelligent modules while the standby supervisor becomes active.

If a Cisco NX-OS router experiences a cold reboot, the network does not forward traffic to the router and removes the router from the network topology. In this scenario, BGP experiences a nongraceful restart and removes all routes. When Cisco NX-OS applies the startup configuration, BGP reestablishes peering sessions and relearns the routes.

A Cisco NX-OS router that has dual supervisors can experience a stateful supervisor switchover. During the switchover, BGP uses nonstop forwarding to forward traffic based on the information in the FIB, and the system is not removed from the network topology. A router whose neighbor is restarting is referred to as a "helper." After the switchover, a graceful restart operation begins. When it is in progress, both routers reestablish their neighbor relationship and exchange their BGP routes. The helper continues to forward prefixes pointing to the restarting peer, and the restarting router continues to forward traffic to peers even though those neighbor relationships are restarting. When the restarting router has all route updates from all BGP peers that are graceful restart capable, the graceful restart is complete, and BGP informs the neighbors that it is operational again.

When a router detects that a graceful restart operation is in progress, both routers exchange their topology tables. When the router has route updates from all BGP peers, it removes all the stale routes and runs the best-path algorithm on the updated routes.

After the switchover, Cisco NX-OS applies the running configuration, and BGP informs the neighbors that it is operational again.

For single-hop iBGP peers with update-source configured under neighbor configuration mode, the peer supports fast external fall-over.

With the additional BGP paths feature, if the number of paths advertised for a given prefix is the same before and after restart, the choice of path ID guarantees the final state and removal of stale paths. If fewer paths are advertised for a given prefix after a restart, stale paths can occur on the graceful restart helper peer.

Low Memory Handling

BGP reacts to low memory for the following conditions:

- Minor alert—BGP does not establish any new eBGP peers. BGP continues to establish new iBGP peers and confederate peers. Established peers remain, but reset peers are not re-established.
- Severe alert—BGP shuts down select established eBGP peers every two minutes until the memory alert becomes minor. For each eBGP peer, BGP calculates the ratio of total number of paths received to the number of paths selected as best paths. The peers with the highest ratio are selected to be shut down to reduce memory usage. You must clear a shutdown eBGP peer before you can bring the eBGP peer back up to avoid oscillation.

Note You can exempt important eBGP peers from this selection process.

- Critical alert—BGP gracefully shuts down all the established peers. You must clear a shutdown BGP peer before you can bring the BGP peer back up.

ISSU

Cisco NX-OS supports in-service software upgrades (ISSU). ISSU allows you to upgrade software without impacting forwarding.

The following conditions are required to support ISSU:

- Graceful restart must be enabled (default)
- Keepalive and hold timers must not be smaller than their default values

If either of these requirements is not met, Cisco NX-OS issues a warning. You can proceed with the upgrade or downgrade, but service might be disrupted.

Note Cisco NX-OS cannot guarantee ISSU for non-default timer values if the negotiated hold time between BGP peers is less than the system switchover time.

Virtualization Support

Cisco NX-OS supports multiple instances of BGP that run on the same system. BGP supports virtual routing and forwarding (VRF) instances that exist within virtual device contexts (VDCs). You can configure one BGP instance in a VDC, but you can have multiple VDCs on the system.

By default, Cisco NX-OS places you in the default VDC and default VRF unless you specifically configure another VDC and VRF.

Prerequisites for Advanced BGP

Advanced BGP has the following prerequisites:

- You must enable BGP.
- You should have a valid router ID configured on the system.
- You must have an AS number, either assigned by a Regional Internet Registry (RIR) or locally administered.
- You must 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

Advanced BGP has the following configuration guidelines and limitations:

- The dynamic AS number prefix peer configuration overrides the individual AS number configuration inherited from a BGP template.
- If you configure a dynamic AS number for prefix peers in an AS confederation, BGP establishes sessions with only the AS numbers in the local confederation.
- Command **ttl-security hops** is visible but not supported for Nexus 7K platform, it is supported only for Nexus 9K platform.
- 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 a redistribution.
- Configure the BGP router ID within a VRF.
- If you decrease the keepalive and hold timer values, the network might experience session flaps.
- When you redistribute BGP to IGP, iBGP is redistributed as well. To override this behavior, you must insert an additional deny statement into the route map.
- Cisco NX-OS does not support multi-hop BFD. BFD for BGP has the following limitations:
  - BFD is supported only for BGP IPv4.
  - BFD is supported only for eBGP peers and iBGP single-hop peers.
  - To enable BFD for iBGP single-hop peers, you must configure the update-source option on the physical interface.

    - BFD is not supported for multi-hop iBGP peers and multi-hop eBGP peers.

- For single-hop iBGP peers with update-source configured under neighbor configuration mode, the peer supports fast external fall-over.
- The following guidelines and limitations apply to the **remove-private-as** command:
    - It applies only to eBGP peers.
    - It can be configured only in neighbor configuration mode and not in neighbor-address-family mode.
    - If the AS-path includes both private and public AS numbers, the private AS numbers are not removed.
    - If the AS-path contains the AS number of the eBGP neighbor, the private AS numbers are not removed.
    - Private AS numbers are removed only if all AS numbers in that AS-path belong to a private AS number range. Private AS numbers are not removed if a peer's AS number or a non-private AS number is found in the AS-path segment.
- BGP conditional route injection is available only for IPv4 and IPv6 unicast address families in all VRF instances.
- The **match interface** command is only supported for **redistribute** command **route-maps**.
- When sending a route advertisement to an iBGP peer, NXOS sets the interface IP address through which the announced network is reachable for the peer as the next hop instead of preserving the original next hop of the non locally originated route.

  This occurs with the 'network' statement and route 'redistribution' configurations in BGP.

  The knobs 'set ip next-hop redist-unchanged' or 'set ipv6 next-hop redist-unchanged' available under route-map configuration mode helps to resolve this issue. These knobs are available from Cisco NX-OS Release 6.2(12) onwards.

Default Settings

Parameters	Default
BGP feature	Disabled
BGP additional paths	Disabled
Hold timer	180 seconds
Keep alive interval	60 seconds
Dynamic capability	Enabled

Configuring Advanced BGP

Configuring BGP Session Templates

You can use BGP session templates to simplify the 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.

Note Use the **show bgp neighbor** command to see the template applied. See the *Cisco Nexus 7000 Series NX-OS Unicast Routing Command Reference*, for details on all commands available in the template.

Before you begin

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

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.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **router bgp** *autonomous-system-number*	Enables BGP and assigns the autonomous system number to the local BGP speaker.
Step 3	switch(config-router)# **template peer-session** *template-name*	Enters peer-session template configuration mode.
Step 4	switch(config-router-stmp)# **password** *number password*	(Optional) Adds the clear text password test to the neighbor. The password is stored and displayed in type 3 encrypted form (3DES).

	Command or Action	Purpose
Step 5	switch(config-router-stmp)# **timers** *keepalive hold*	(Optional) 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	switch(config-router-stmp)# **exit**	Exits peer-session template configuration mode.
Step 7	switch(config-router)# **neighbor** *ip-address* **remote-as** *as-number*	Places the router in the neighbor configuration mode for BGP routing and configures the neighbor IP address.
Step 8	switch(config-router-neighbor)# **inherit peer-session** *template-name*	Applies a peer-session template to the peer.
Step 9	switch(config-router-neighbor)# **description** *text*	(Optional) Adds a description for the neighbor.
Step 10	switch(config-router-neighbor)# **show bgp peer-session** *template-name*	(Optional) Displays the peer-policy template.
Step 11	switch(config-router-neighbor)# **copy running-config startup-config**	(Optional) Saves this configuration change.

Example

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 65535
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 65535
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.

Note Use the **show bgp neighbor** command to see the template applied. See the *Cisco Nexus 7000 Series NX-OS Unicast Routing Command Reference*, for details on all commands available in the template.

Before you begin

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

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.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **router bgp** *autonomous-system-number*	Enables BGP and assigns the autonomous system number to the local BGP speaker.
Step 3	switch(config-router)# **template peer-policy** *template-name*	Creates a peer-policy template.
Step 4	switch(config-router-ptmp)# **advertise-active-only**	(Optional) Advertises only active routes to the peer.
Step 5	switch(config-router-ptmp)# **maximum-prefix** *number*	(Optional) Sets the maximum number of prefixes allowed from this peer.
Step 6	switch(config-router-ptmp)# **exit**	Exits peer-policy template configuration mode.
Step 7	switch(config-router)# **neighbor** *ip-address* **remote-as** *as-number*	Places the router in the neighbor configuration mode for BGP routing and configures the neighbor IP address.
Step 8	switch(config-router-neighbor)# **address-family**{**ipv4** | **ipv6** |**vpnv4** | **vpnv6**} {**multicast** | **unicast**}	Enters global address family configuration mode.
Step 9	switch(config-router-neighbor-af)# **inherit peer-policy** *template-name preference*	Applies a peer-policy template to the peer address family configuration and assigns the preference value for this peer policy.
Step 10	switch(config-router-neighbor-af)# **show bgp peer-policy** *template-name*	(Optional) Displays the peer-policy template.

	Command or Action	Purpose
Step 11	switch(config-router-neighbor-af)# **copy running-config startup-config**	(Optional) Saves this configuration change.

Example

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

```
switch# configure terminal
switch(config)# router bgp 65535
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 65535
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.

Note Use the **show bgp neighbor** command to see the template applied. See the *Cisco Nexus 7000 Series NX-OS Unicast Routing Command Reference*, for details on all commands available in the template.

Before you begin

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

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.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.

	Command or Action	Purpose
Step 2	switch(config)# **router bgp** *autonomous-system-number*	Enables BGP and assigns the autonomous system number to the local BGP speaker.
Step 3	switch(config-router)# **template peer** *template-name*	Enter peer template configuration mode.
Step 4	switch(config-router-neighbor)# **inherit peer-session** *template-name*	(Optional) Inherits a peer-session template in the peer template.
Step 5	switch(config-router-neighbor)# **address-family** {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} {**multicast** | **unicast**}	(Optional) Configures the global address family configuration mode.
Step 6	switch(config-router-neighbor-af)# **inherit peer** *template-name*	(Optional) Applies a peer template to the neighbor address family configuration.
Step 7	switch(config-router-neighbor-af)# **exit**	Exits BGP neighbor address family configuration mode.
Step 8	switch(config-router-neighbor)# **timers** *keepalive hold*	(Optional) Adds the BGP timer values to the peer. These values override the timer values in the peer-session template, BaseSession.
Step 9	switch(config-router-neighbor)# **exit**	Exits BGP peer template configuration mode.
Step 10	switch(config-router)# **neighbor** *ip-address* **remote-as** *as-number*	Places the router in neighbor configuration mode for BGP routing and configures the neighbor IP address.
Step 11	switch(config-router-neighbor)# **inherit peer** *template-name*	Inherits the peer template.
Step 12	switch(config-router-neighbor)# **timers** *keepalive hold*	(Optional) Adds the BGP timer values to this neighbor. These values override the timer values in the peer template and the peer-session template.
Step 13	switch(config-router-neighbor-af)# **show bgp peer-template** *template-name*	(Optional) Displays the peer template.
Step 14	switch(config-router-neighbor-af)# **copy running-config startup-config**	(Optional) Saves this configuration change.

Example

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

```
switch# configure terminal
switch(config)# router bgp 65535
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 65535
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.

Note Use the **show ip bgp neighbor** 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.

Use the **show bgp covergence private** command to display details of the prefix peer wait timer.

Procedure

	Command or Action	Purpose
Step 1	(Optional) switch(config-router-neighbor)# **timers prefix-peer-timeout** *interval*	Configures the BGP prefix peering timeout value. When a BGP peer that is part of a prefix peering disconnects, the peer structures are held for a defined prefix peer timeout value which enables the peer to reset and reconnect without danger of being blocked. The timeout range is from 0 to 1200 seconds. The default value is 30.
Step 2	(Optional) switch(config-router-neighbor)# **timers prefix-peer-wait** *interval*	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. **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 the dynamic BGP neighbors.

	Command or Action	Purpose
Step 3	(Optional) switch(config-router-neighbor)# **maximum-peers** *value*	Configures the maximum number of peers for this prefix peering in neighbor configuration mode. The range is from 1 to 1000.

Example

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

```
switch(config)# router bgp 65535
switch(config-router)# timers prefix-peer-timeout 120
switch(config-router)# neighbor 10.100.200.0/24 remote-as 65535
switch(config-router-neighbor)# maximum-peers 10
switch(config-router-neighbor)# address-family ipv4 unicast
switch(config-router-neighbor-af)#
```

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:

Procedure

	Command or Action	Purpose
Step 1	switch(config-router-neighbor)# **password** {**0** | **3** | **7**} *string*	Configures an MD5 password (for authentication) for BGP neighbor sessions in neighbor configuration mode.

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.

Procedure

	Command or Action	Purpose
Step 1	switch(config-router-neighbor-af)# **soft-reconfiguration inbound**	This command in neighbor address-family configuration mode, enables soft reconfiguration to store the inbound BGP route updates. This command triggers an automatic soft clear or refresh of BGP neighbor sessions.
Step 2	switch# **clear bgp** {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} {**unicast** | **multicast** *ip-address* **soft** {**in** | **out**}	This command in any mode 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 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 commands in address-family configuration mode:

Procedure

	Command or Action	Purpose
Step 1	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.
Step 2	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.

Configuring BGP Next-Hop Address Tracking

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

You can modify the delay interval between RIB checks to increase the performance of BGP next-hop tracking.

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

Procedure

	Command or Action	Purpose
Step 1	switch(config-router-af)# **nexthop trigger-delay** {**critical** | **non-critical**} *milliseconds*	Specifies the next-hop address tracking delay timer for critical next-hop reachability routes and for noncritical routes. The range is from 1 to 4294967295 milliseconds. The critical timer default is 3000. The noncritical timer default is 10000.
Step 2	switch(config-router-af)# **nexthop route-map** *name*	Specifies a route map to match the BGP next-hop addresses to. The name can be any case-sensitive, alphanumeric string up to 63 characters.

Configuring Next-Hop Filtering

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

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

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

Procedure

	Command or Action	Purpose
Step 1	switch(config-router-af)# **nexthop route-map** *name*	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.

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:

Procedure

	Command or Action	Purpose
Step 1	switch(config-router-neighbor)# **dont-capability-negotiate**	Disables capabilities negotiation. You must manually reset the BGP sessions after configuring this command.

Configuring BGP Additional Paths

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

Advertising the Capability of Sending and Receiving Additional Paths

You can configure BGP to advertise the capability of sending and receiving additional paths to and from the BGP peers.

Procedure

	Command or Action	Purpose
Step 1	switch(config-router-neighbor-af)# [**no**]**capability additional paths send** [**disable**]	Advertises the capability to send additional paths to the BGP peer. The disable option

	Command or Action	Purpose
		disables the advertising capability of sending additional paths. The no form of this command disables the capability of sending additional paths.
Step 2	switch (config-router-neighbor-af)# [**no**]**capability additional paths receive** [**disable**]	Advertises the capability to send additional paths to the BGP peer. The disable option disables the advertising capability of sending additional paths. The no form of this command disables the capability of sending additional paths.
Step 3	switch(config-router-neighbor-af)# **show bgp neighbor**	Displays whether the local peer has advertised the additional paths send or receive capability to the remote peer.
Step 4	switch(config-router-neighbor-af)# **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 BGP to advertise the capability to send and receive additional paths to the BGP peer:

```
switch(config)# router bgp 100
switch(config-router)# neighbor 10.131.31.2 remote-as 100
switch(config-router-neighbor)# address-family ipv4 unicast
switch(config-router-neighbor-af)# capability additional-paths send
switch(config-router-neighbor-af)#capability additional-paths receive
switch(config-router-neighbor-af)# show bgp neighbor
switch(config-router-neighbor-af)# copy running-config startup-config
```

Configuring the Sending and Receiving of Additional Paths

You can configure the capability of sending and receiving additional paths to and from the BGP peers.

Procedure

	Command or Action	Purpose
Step 1	switch(config-router-neighbor-af)# [**no**]**additional-paths send**	Enables the send capability of additional paths for all of the neighbors under this address family for which the capability has not been disabled. The no form of this command disables the send capability.
Step 2	switch (config-router-neighbor-af)# [**no**]**additional-paths receive** [**disable**]	Enables the receive capability of additional paths for all of the neighbors under this address

	Command or Action	Purpose
		family for which the capability has not been disabled. The no form of this command disables the capability of sending additional paths.
Step 3	switch(config-router-neighbor-af)# **show bgp neighbor**	Displays whether the local peer has advertised the additional paths send or receive capability to the remote peer.
Step 4	switch(config-router-neighbor-af)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to enable the additional paths send and receive capability for neighbors under the specified address family for which this capability has not been disabled.:

```
switch(config)# router bgp 100
switch(config-router)# neighbor 10.131.31.2 remote-as 100
switch(config-router-neighbor)# address-family ipv4 unicast
switch(config-router-neighbor-af)# additional-paths send
switch(config-router-neighbor-af)#additional-paths receive
switch(config-router-neighbor-af)# show bgp neighbor
switch(config-router-neighbor-af)# copy running-config startup-config
```

Configuring Advertised Paths

You can specify the paths that are advertised for BGP.

Procedure

	Command or Action	Purpose
Step 1	switch(config-route-map)# [**no**]**set path-selection all advertise**	Specifies that all paths be advertised for a given prefix. The no form of this command specifies that only the best path be advertised.
Step 2	switch(config-route-map)# **show bgp neighbor**{**ipv4** | **ipv6**} **unicast***ip-address* | *ipv6-prefix* [**vrf***vrf-name*]	Displays whether the local peer has advertised the additional paths send or receive capability to the remote peer.
Step 3	switch(config-route-map)# **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 specify that all paths be advertised for the specified prefix:

```
switch(config)# route-map PATH_SELECTION_RMAP
switch(config-route-map)#match ip address prefeix-list pl
switch(config-route-map)# show bgp ip4 unicast
switch(config-route-map)# copy running-config startup-config
```

Configuring Additional Path Selection

You can configure the capability of selecting additional paths for a prefix.

Procedure

	Command or Action	Purpose
Step 1	switch(config-router-af)# [**no**]**additional-paths selection route-map***map-name*	Specifies that all paths be advertised for a given prefix. The no form of this command specifies that only the best path be advertised.
Step 2	switch(config-router-af)# **show bgp** {**ipv4** | **ipv6**} **unicast***ip-address* | *ipv6-prefix* [**vrf***vrf-name*]	Displays whether the local peer has advertised the additional paths send or receive capability to the remote peer.
Step 3	(Optional) switch(config-router-af)# **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 specify that all paths be advertised for the specified prefix:

```
switch(config)# router bgp 100
switch(config-router)# address-family ipv4 unicast
switch(config-router-af)#additional-paths selection route-map PATH_SELECTION_RMAP
switch(config-router-af)# copy running-config startup-config
```

Configuring eBGP

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:

Procedure

	Command or Action	Purpose
Step 1	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:

Procedure

	Command or Action	Purpose
Step 1	switch(config-router-neighbor)# **ebgp-multihop** *ttl-value*	Configures the eBGP TTL value for eBGP multihop. The range is from 2 to 255. You must manually reset the BGP sessions after using this command.

Disabling a Fast External Fallover

By default, the Cisco Nexus 7000 Series device supports fast external fallover for neighbors in all VRFs and address-families (IPv4 or IPv6).

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

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

Procedure

	Command or Action	Purpose
Step 1	switch(config-router)# **no fast-external-fallover**	Disables a fast external fallover 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:

Procedure

	Command or Action	Purpose
Step 1	switch(config-router)# **maxas-limit** *number*	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.

This feature can only be used for true eBGP peers. You cannot use this feature for two peers that are members of different confederation sub-autonomous systems.

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

Procedure

	Command or Action	Purpose
Step 1	switch(config-router-neighbor)# **local-as** *number* [**no-prepend** [**replace-as** [**dual-as**]]]	Configures eBGP to prepend the local AS number to the AS_PATH attribute. 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.

Example

The local-AS feature under VRF configuration mode is supported for both IBGP and EBGP neighbor relationships.

This following example shows how to configure the feature for the IBGP neighbor 10.1.2.1:

```
router bgp 65001
  vrf BGP1
    local-as 65002
    address-family ipv4 unicast
    neighbor 10.1.2.1 remote-as 65002
```

The **local-as** command must be configured in the neighbor configuration mode for eBGP or a warning message is displayed stating that the local AS cannot be same as the remote AS. The following example shows how to configure the local-AS feature for eGBP:

```
router bgp 65001
  vrf BGP1
    neighbor 20.1.2.1 remote-as 65003
      local-as 65001
```

Configuring AS Confederations

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

Procedure

	Command or Action	Purpose
Step 1	switch(config-router)# **confederation identifier** *as-number*	In router configuration mode, this command configures a BGP confederation identifier. The command triggers an automatic notification and session reset for the BGP neighbor sessions.
Step 2	switch(config-router)# **bgp confederation peers** *as-number* [*as-number2...*]	In router configuration mode, this command configures the autonomous systems that belong to the AS confederation. The command specifies a list of autonomous systems that belong to the confederation and it triggers an automatic notification and session reset for the BGP neighbor sessions.

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

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

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **router bgp** *as-number*	Enters BGP mode and assigns the autonomous system number to the local BGP speaker.
Step 3	switch(config-router)# **cluster-id** *cluster-id*	Configures the local router as one of the route reflectors that serve the cluster. You specify a

	Command or Action	Purpose
		cluster ID to identify the cluster. This command triggers an automatic soft clear or refresh of BGP neighbor sessions.
Step 4	switch(config-router)# **address-family** {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} {**unicast** | **multicast**}	Enters router address family configuration mode for the specified address family.
Step 5	switch(config-router-af)# **client-to-client reflection**	(Optional) 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	switch(config-router-neighbor)# **exit**	Exits router address configuration mode.
Step 7	switch(config-router)# **neighbor** *ip-address* **remote-as** *as-number*	Configures the IP address and AS number for a remote BGP peer.
Step 8	switch(config-router-neighbor)# **address-family** {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} {**unicast** | **multicast**}	Enters neighbor address family configuration mode for the unicast IPv4 address family.
Step 9	switch(config-router-neighbor-af)# **route-reflector-client**	Configures the device 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) switch(config-router-neighbor-af)# **show bgp** {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} {**unicast** | **multicast**} **neighbors**	Displays the BGP peers.
Step 11	(Optional) 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 65535
switch(config-router)# neighbor 192.0.2.10 remote-as 65535
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 Next-Hops on Reflected Routes Using an Outbound Route-Map

You can change the next-hop on reflected routes on a BGP route reflector using an outbound route-map. You can configure the outbound route-map to specify the peer's local address as the next-hop address.

Note The **next-hop-self** command does not enable this functionality for routes being reflected to clients by a route reflector. This functionality can only be enabled using an outbound route-map.

Before you begin

You must enable BGP.

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

You must enter the **set next-hop** command to configure an address family specific next-hop address. For example, for the IPv6 address family, you must enter the **set ipv6 next-hop peer-address** command.

- When setting IPv4 next-hops using route-maps—If **set ip next-hop peer-address** matches the route-map, the next-hop is set to the peer's local address. If no next-hop is set in the route-map, the next-hop is set to the one stored in the path.
- When setting IPv6 next-hops using route-maps—If **set ipv6 next-hop peer-address** matches the route-map, the next-hop is set as follows:
  - For IPv6 peers, the next-hop is set to the peer's local IPv6 address.
  - For IPv4 peers, if **update-source** is configured, the next-hop is set to the source interface's IPv6 address, if any. If no IPv6 address is configured, no next-hop is set
  - For IPv4 peers, if **update-source** is not configured, the next-hop is set to the outgoing interface's IPv6 address, if any. If no IPv6 address is configured, no next-hop is set.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **router bgp** *as-number*	Enters BGP mode and assigns the autonomous system number to the local BGP speaker.
Step 3	switch(config-router)# **neighbor** *ip-address* **remote-as** *as-number*	Configures the IP address and AS number for a remote BGP peer.
Step 4	switch(config-router-neighbor)# **update-source** *interface number*	(Optional) Specifies and updates the source of the BGP session.
Step 5	switch(config-router-neighbor)# **address-family** {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} {**unicast** | **multicast**}	Enters router address family configuration mode for the specified address family.
Step 6	switch(config-router-neighbor-af)# **route-reflector-client**	Configures the device 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 7	switch(config-router-neighbor-af)# **route-map** *map-name* **out**	Applies the configured BGP policy to outgoing routes.

	Command or Action	Purpose
Step 8	(Optional) switch(config-router-neighbor-af)# **show bgp** {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} {**unicast** | **multicast**} [**ip-address** | **ipv6-prefix**] **route-map** *map-name* [**vrf** *vrf-name*]	Displays the BGP routes that match the route map.
Step 9	(Optional) switch(config-router-neighbor-af)# **copy running-config startup-config**	Saves this configuration change.

Example

This example shows how to configure the next-hop on reflected routes on a BGP route reflector using an outbound route-map:

```
switch(config)# interface loopback 300
switch(config-if)# ip address 192.0.2.11/32
switch(config-if)# ipv6 address 2001::a0c:1a65/64
switch(config-if)# ip router ospf 1 area 0.0.0.0
switch(config-if)# exit
switch(config)# route-map setrrnh permit 10
switch(config-route-map)# set ip next-hop peer-address
switch(config-route-map)# exit
switch(config)# route-map setrrnhv6 permit 10
switch(config-route-map)# set ipv6 next-hop peer-address
switch(config-route-map)# exit
switch(config)# router bgp 200
switch(config-router)# neighbor 192.0.2.12 remote-as 200
switch(config-router-neighbor)# update-source loopback 300
switch(config-router-neighbor)# address-family ipv4 unicast
switch(config-router-neighbor-af)# route-reflector-client
switch(config-router-neighbor-af)# route-map setrrnh out
switch(config-router-neighbor-af)# exit
switch(config-router-neighbor)# address-family ipv6 unicast
switch(config-router-neighbor-af)# route-reflector-client
switch(config-router-neighbor-af)# route-map setrrnhv6 out
```

Configuring Route Dampening

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

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

Procedure

	Command or Action	Purpose
Step 1	switch (config-router-af)# **dampening** [*half-life reuse-limit suppress-limit max-suppress-time* | **route-map** *map-name*}]	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.

	Command or Action	Purpose
		• suppress-limit—The range is from 1 to 20000. • max-suppress-time—The range is from 1 to 255

Configuring Load Sharing and ECMP

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

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

Command	Purpose
switch(config-router-af)# **maximum-paths** [**ibgp**] *maxpaths*	Configures the maximum number of equal-cost paths for load sharing. The range is from 1 to 16. The default is 1. Starting from Cisco NX-OS Release 8.4(1), the range is from 1 to 64 on M3- and F3-Series I/O modules. Starting from Cisco NX-OS Release 8.4(2), the range is from 1 to 64 on F4-Series I/O modules.

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:

Procedure

	Command or Action	Purpose
Step 1	switch(config-router-neighbor-af)# **maximum-prefix** *maximum* [*threshold*] [**restart***time* | **warning-only**]	Configure 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
switch(config-router-neighbor)# **dynamic-capability**	Enables dynamic capability. This command triggers an automatic notification and session reset for the BGP neighbor sessions.

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*]	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** *map-name* keyword and argument specify the route map used to select attribute information from selected routes. • The **attribute-map** *map-name* keyword and argument specify the route map used to select attribute information from the aggregate. • The **suppress-map** *map-name* keyword and argument conditionally filter more specific routes.

Unsuppressing the Advertisement of Aggregated Routes

You can configure BGP to advertise routes that are suppressed by the **aggregate-address** command.

To unsuppress the advertising of aggregated routes, use the following command in router neighbor address-family configuration mode:

Procedure

	Command or Action	Purpose
Step 1	switch(config-route-neighbor-af)# **unsuppress-map** *map-name*	Advertises selective routes that are suppressed by the **aggregate-address** command.

Configuring BGP Conditional Route Injection

You can configure BGP conditional route injection to inject specific routes based on the administrative policy or traffic engineering information and control the packets being forwarded to these specific routes, which are injected into the BGP routing table only if the configured conditions are met. This feature allows you to improve the accuracy of common route aggregation by conditionally injecting or replacing less specific prefixes with more specific prefixes. Only prefixes that are equal to or more specific than the original prefix can be injected.

Note The injected prefixes inherit the attributes of the aggregated route.

Before you begin

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

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **router bgp** *as-number*	Enters BGP configuration mode and assigns the autonomous system number to the local BGP speaker.
Step 3	switch(config-router)# **address-family** {**ipv4** | **ipv6**} **unicast**	Enters address family configuration mode.
Step 4	switch(config-router-af)# **inject-map** *inject-map-name* **exist-map** *exist-map-name* [**copy-attributes**]	Specifies the inject-map and exist-map routes for conditional route injection. These maps install one or more prefixes intoa BGP routing table. The *exist-map* route map specifies the prefixes that BGP tracks, and the *inject-map* route map defines the prefixes that are created and installed into the local BGP table. Use the **copy-attributes** keyword to specify that the injected route inherits the attributes of the aggregate route.
Step 5	switch(config-router-af)# **exit**	Exits address family configuration mode.
Step 6	switch(config-router)# **exit**	Exits BGP configuration mode.
Step 7	switch(config)# **ip prefix-list** *list-name* **seq** *sequence-number* **permit***network-length*	Configures a prefix list. Repeat this step for every prefix list to be created.
Step 8	switch(config)# **route-map** *map-name* **permit** *sequence-number*	Configures a route-map and enters route-map configuration mode.

	Command or Action	Purpose
Step 9	switch(config-route-map)# **match ip address prefix-list** *prefix-list-name*	Specifies the aggregate route to which a more specific route will be injected.
Step 10	switch(config-route-map)# **match ip route-source prefix-list** *prefix-list-name*	Specifies the match conditions for the source fo the route.
Step 11	switch(config-route-map)# **exit**	Exits route-map configuration mode.
Step 12	switch(config)# **ip prefix-list** *list-name* **seq** *sequence-number* **permit** *network-length*	Configures a prefix list. Repeat this step for every prefix list to be created.
Step 13	switch(config)# **route-map** *map-name* **permit** *sequence-number*	Configures a route map and enters route-map configuration mode.
Step 14	switch(config-route-map)# **set ip address prefix-list**	Specifies the routes to be injected.
Step 15	(Optional) switch(config-route-map)# **show bgp** {**ipv4** | **ipv6**} **unicast injected-routes**	Displays injected routes in the routing table.
Step 16	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

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

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

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.

	Command or Action	Purpose
Step 2	switch(config)# **router bgp** *as-number*	Enters BGP mode and assigns the autonomous system number to the local BGP speaker.
Step 3	switch(config-router)#**neighbor** *ip-address* **remote-as** *as-number*	Places the router in neighbor configuration mode for BGP routing and configures the neighbor IP address.
Step 4	switch(config-router-neighbor)# **address-family** {**ipv4**|**ipv6**|**vpnv4**|**vpnv6**} {**unicast**|**multicast**}	Enters address family configuration mode.
Step 5	switch(config-router-neighbor-af)# **advertise-map** *adv-map* {**exist-map** *exist-rmap*|**non-exist-map** *nonexist-rmap*}	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 advertises 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 advertises the route. The nonexist-rmap is a case-sensitive, alphanumeric string up to 63 characters.
Step 6	(Optional) switch(config-router-neighbor-af)# **show ip bgp neighbor**	Displays information about BGP and the configured conditional advertisement route maps.
Step 7	(Optional) 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 65535
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 metric for redistributed routes.

Before you begin

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

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **router bgp** *as-number*	Enters BGP mode and assigns the autonomous system number to the local BGP speaker.
Step 3	switch(config-router)# **address-family** {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} {**unicast** | **multicast**}	Enters address family configuration mode.
Step 4	switch(config-router-af)# **redistribute** {**direct** | {**eigrp** | **isis** | **ospf** | **ospfv3** | **rip**} *instance-tag* | **static**} **route-map** *map-name*	Redistributes routes from other protocols into BGP.
Step 5	(Optional) switch(config-router-af)# **default-metric** *value*	Generates a default metric into BGP.
Step 6	(Optional) switch(config-router-neighbor-af)# **copy running-config startup-config**	Saves this configuration change.

Example

This example shows how to redistribute EIGRP into BGP:

```
switch# configure terminal
switch(config)# router bgp 65535
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
```

Advertising the Default Route

You can configure BGP to advertise the default route (network 0.0.0.0).

Before you begin

You must enable BGP.

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

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **route-map allow permit**	Enters router map configuration mode and defines the conditions for redistributing routes
Step 3	switch(config-route-map)# **exit**	Exits router map configuration mode.
Step 4	switch(config)# **ip route** *ip-address network-mask* **null** *null-interface-number*	Configures the IP address.
Step 5	switch(config)# **router bgp** *as-number*	Enters BGP mode and assigns the AS number to the local BGP speaker.
Step 6	switch(config-router)# **address-family** {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} **unicast**	Enters address family configuration mode.
Step 7	switch(config-router-af)# **default-information originate**	Advertises the default route.
Step 8	switch(config-router-af)# **redistribute static route-map allow**	Redistributes the default route.
Step 9	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Configuring Route Import from Default VRF to any other VRF

Perform the following steps to import routes from default VRF to any other non-default VRF.

Before you begin

- Enable BGP.
- Ensure that you are in the correct VDC.

Procedure

Step 1 Enter the global configuration mode:

switch#**configure terminal**

Step 2 Enable BGP:

switch(config)#**feature bgp**

Step 3 Create a new VRF and enter VRF configuration mode:

switch(config)#**vrf context** *vrf-name*

Step 4 Enter the IPv4 / IPv6 unicast address family configuration mode:

switch(config-vrf)# **address-family** {**ipv4** | **ipv6**} unicast

Step 5 Configure an import policy for a VRF to import prefixes from the default VRF:

switch(config-vrf-af)# **import vrf default** [*prefix-limit*] **map** *route-map*

prefix-limit limits the number of routes that can be imported. Default value is 1000.

route-map specifies the route-map to be imported and can be case-sensitive, alphanumeric string up to 63 characters.

Configuring Route Export from BGP VRF to Default VRF

Perform the following steps to export routes from non-default VRF to Default VRF.

Before you begin

- Enable BGP.
- Ensure that you are in the correct VDC.

Procedure

Step 1 Enter the global configuration mode:

switch#**configure terminal**

Step 2 Enable BGP:

switch(config)#**feature bgp**

Step 3 Create a new VRF and enter VRF configuration mode:

switch(config)#**vrf context** *vrf-name*

Step 4 Enter the IPv4 / IPv6 unicast address family configuration mode:

switch(config-vrf)# **address-family** {**ipv4** | **ipv6**} unicast

Step 5 Export IPv4 or IPv6 prefixes from non-default VRF to default VRF, filtered by *route-map.*:

switch(config-vrf-af)# **export vrf default** [*prefix-limit*] **map** *route-map*

prefix-limit limits the number of routes that can be exported, in order to avoid the global table being overloaded. Default value is 1000.

route-map can be case-sensitive, alphanumeric string up to 63 characters. It specifies the route-map.

If the route map does not exist, the command will be accepted but processed at a later time when the route map is created.

Example

The following example shows how to export the route map, BgpMap, to default VRF, and verify the configuration.

```
switch# configure terminal
switch(config)# feature bgp
switch(config)#vrf context vpn1
switch(config-vrf)# address-family ipv4 unicast
switch(config-vrf-af)# export vrf default 3 map BgpMap
switch(config-vrf-af)# exit
switch(config)#show bgp process vrf vpn1

Information regarding configured VRFs:

BGP Information for VRF vpn1
VRF Id                          : 3
VRF state                       : UP
Router-ID                       : 20.0.0.1
Configured Router-ID            : 0.0.0.0
Confed-ID                       : 0
Cluster-ID                      : 0.0.0.0
No. of configured peers         : 2
No. of pending config peers     : 0
No. of established peers        : 2
VRF RD                          : 100:1

    Information for address family IPv4 Unicast in VRF vpn1
    Table Id                    : 3
    Table state                 : UP
    Peers      Active-peers    Routes     Paths      Networks   Aggregates
    1          1               6          6          0          0

    Redistribution
        static, route-map allow

    Export RT list:
        100:1
        1000:1
    Import RT list:
        100:1
    Label mode: per-prefix
    Aggregate label: 492287
    Import default limit       : 1000
    Import default prefix count : 2
    Import default map         : allow
    Export default limit       : 1000
```

```
Export default prefix count : 3
Export default map          : allow
```

Configuring Multiprotocol BGP

You can configure MP-BGP to support multiple address families, including IPv4 and IPv6 unicast and multicast routes.

Before you begin

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

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **router bgp** *as-number*	Enters BGP mode and assigns the autonomous system number to the local BGP speaker.
Step 3	switch(config-router)#**neighbor** *ip-address* **remote-as** *as-number*	Places the router in neighbor configuration mode for BGP routing and configures the neighbor IP address.
Step 4	switch(config-router-neighbor)# **address-family** {**ipv4**|**ipv6**|**vpnv4**|**vpnv6**} {**unicast**|**multicast**}	Enters address family configuration mode.
Step 5	(Optional) 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 and IPv6 routes for multicast RPF for a neighbor:

```
switch# configure terminal
switch(config)# interface ethernet 2/1
switch(config-if)# ipv6 address 2001:0DB8::1
switch(config-if)# router bgp 65535
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 Policy-Based Administrative Distance

You can configure a distance for external BGP (eBGP) and internal BGP (iBGP) routes that match a policy described in the configured route map. The distance configured in the route map is downloaded to the unicast

RIB along with the matching routes. BGP uses the best path to determine the administrative distance when downloading next hops in the unicast RIB table. If there is no match or a deny clause in the policy, BGP uses the distance configured in the distance command or the default distance for routes.

The policy-based administrative distance feature is useful when there are two or more different routes to the same destination from two different routing protocols.

Before you begin

You must enable BGP.

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

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **ip prefix-list** *name* **seq** *number* **permit** *prefix-length*	Creates a prefix list to match IP packets or routes with the permit keyword.
Step 3	switch(config)# **route-map** *map-tag* **permit** *sequence-number*	Creates a route map and enters route-map configuration mode with the permit keyword. If the match criteria for the route is met in the policy, the packet is policy routed.
Step 4	switch(config-route-map)# **match ip address prefix-list** *prefix-list-name*	Matches IPv4 network routes based on a prefix list. The prefix-list name can be any alphanumeric string up to 63 characters.
Step 5	switch(config-route-map)# **set distance** *<value1> <value2> <value3>*	Specifies the administrative distance for interior BGP (iBGP) or exterior BGP (eBGP) routes and BGP routes originated in the local autonomous system. The range is from 1 to 255. After you enter the value for the external administrative distance, you must enter the value for the administrative distance for the internal routes or/and the value for the administrative distance for the local routes depending on your requirement; so that the internal/local routes are also considered in the route administration.
Step 6	switch(config-route-map)# **exit**	Exits route-map configuration mode.
Step 7	switch(config)# **router bgp** *as-number*	Enters BGP mode and assigns the AS number to the local BGP speaker.
Step 8	switch(config-router)# **address-family** {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} **unicast**	Enters address family configuration mode.
Step 9	switch(config-router-af)# **table-map** *map-name*	Configures the selective administrative distance for a route map for BGP routes before

	Command or Action	Purpose
		forwarding them to the RIB table. The table-map name can be any alphanumeric string up to 63 characters. **Note** You can also configure the **table-map** command under the VRF address-family configuration mode.
Step 10	(Optional) switch(config-router-af)# **show forwarding distribution**	Displays forwarding information distribution.
Step 11	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Tuning BGP

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

Procedure

	Command or Action	Purpose
Step 1	Required: switch(config-router)# **bestpath** [**always-compare-med** | **as-path multipath-relax** | **compare-routerid** | **cost-community ignore** | **med** {**confed** | **missing-as-worst** | **non-deterministic**}]	Modifies the best-path algorithm. The optional parameters are as follows: • **always-compare-med**—Compares MED on paths from different autonomous systems. • **as-path multipath-relax**—Allows load sharing across the providers with different (but equal-length) AS paths. Without this option, the AS paths must be identical for load sharing. • **compare-routerid**—Compares the router IDs for identical eBGP paths. • **cost-community ignore**—Ignores the cost community for BGP best-path calculations. For more information on the BGP cost community, see the "Configuring MPLS Layer 3 VPN Load Balancing" chapter of the Cisco Nexus 7000 Series NX-OS MPLS Configuration Guide. • **med confed**—Forces bestpath to do a MED comparison only between paths originated within a confederation.

	Command or Action	Purpose
		• **med missing-as-worst**—Treats a missing MED as the highest MED. • **med non-deterministic**—Does not always pick the best MED path from among the paths from the same autonomous system.
Step 2	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.
Step 3	switch(config-router)# **log-neighbor-changes**	Generates a system message when a neighbor changes state.
Step 4	switch(config-router)# **router-id** *id*	Manually configures the router ID for this BGP speaker.
Step 5	switch(config-router)# **timers** [**bestpath-delay** *delay* | **bgp** *keepalive holdtime* | **prefix-peer-timeout** *timeout*]	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.
Step 6	switch(config-router-af)# **distance** *ebgp-distance ibgp-distance local-distance*	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.
Step 7	switch(config-router-neighbor)# **description** *string*	Sets a descriptive string for this BGP peer. The string can be up to 80 alphanumeric characters.

	Command or Action	Purpose
Step 8	switch(config-router-neighbor)# **low-memory exempt**	Exempts this BGP neighbor from a possible shutdown due to a low memory condition.
Step 9	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.
Step 10	**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 more information on this command.
Step 11	switch(config-router-neighbor)# **update-source** *interface-type number*	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. Single-hop iBGP peers support fast external failover when **update-source** is configured.
Step 12	switch(config-router-neighbor)# **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.
Step 13	switch(config-router-neighbor)# **default-originate** [**route-map** *map-name*]	Generates a default route to the BGP peer.
Step 14	switch(config-router-neighbor)# **filter-list** *list-name* {**in**|**out**}	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.
Step 15	switch(config-router-neighbor)# **prefix-list** *list-name* {**in**|**out**}	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.
Step 16	switch(config-router-neighbor)# **send-community**	Sends the community attribute to this BGP peer. This command triggers an automatic soft clear or refresh of BGP neighbor sessions.
Step 17	switch(config-router-neighbor)# **send-community extended**	Sends the extended community attribute to this BGP peer. This command triggers an automatic soft clear or refresh of BGP neighbor sessions.

Configuring a Graceful Restart

You can configure a graceful restart and enable the graceful restart helper feature for BGP.

Before you begin

You must enable BGP.

Create the VDCs and VRFs.

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

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **router bgp** *as-number*	Creates a new BGP process with the configured autonomous system number.
Step 3	switch(config-router)# **graceful-restart**	Enables a graceful restart and the graceful restart helper functionality. This command is enabled by default. This command triggers an automatic notification and session reset for the BGP neighbor sessions.
Step 4	switch(config-router)# **graceful-restart** {**restart-time** *time*|**stalepath-time** *time*}	Configures the graceful restart timers. The optional parameters are as follows: • **restart-time**—Maximum time for a restart sent to the BGP peer. The range is from 1 to 3600 seconds. The default is 120. • **stalepath-time**—Maximum time that BGP keeps the stale routes from the restarting BGP peer. The range is from 1 to 3600 seconds. The default is 300. This command triggers an automatic notification and session reset for the BGP neighbor sessions.
Step 5	switch(config-router)# **graceful-restart-helper**	Enables the graceful restart helper functionality. Use this command if you have disabled graceful restart but you still want to enable graceful restart helper functionality. This command triggers an automatic notification and session reset for the BGP neighbor sessions.
Step 6	(Optional) switch(config-router)# **show running-config bgp**	Displays the BGP configuration.

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

Example

This example shows how to enable a graceful restart:

```
switch# configure terminal
switch(config)# router bgp 65535
switch(config-router)# graceful-restart
switch(config-router)# copy running-config startup-config
```

Configuring Virtualization

You can configure one BGP process in each VDC. You can create multiple VRFs within each VDC and use the same BGP process in each VRF.

Before you begin

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

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **vrf context** *vrf-name*	Creates a new VRF and enters VRF configuration mode.
Step 3	switch(config-vrf)# **exit**	Exits VRF configuration mode.
Step 4	switch(config)# **router bgp** *as-number*	Creates a new BGP process with the configured autonomous system number.
Step 5	switch(config-router)# **vrf** *vrf-name*	Enters the router VRF configuration mode and associates this BGP instance with a VRF.
Step 6	switch(config-router-vrf)# **neighbor** *ip-address* **remote-as** *as-number*	Configures the IP address and AS number for a remote BGP peer.
Step 7	(Optional) switch(config-router-neighbor-af)# **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 65535
switch(config-router)# vrf NewVRF
switch(config-router-vrf)# neighbor 209.165.201.1 remote-as 65535
switch(config-router-vrf-neighbor)# copy running-config startup-config
```

Verifying the Advanced BGP Configuration

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

Command	Purpose
show bgp all [**summary**] [**vrf** *vrf-name*]	Displays the BGP information for all address families.
show bgp convergence vrf *vrf-name*	Displays the BGP information for all address families.
show bgp {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **community** {**regexp** *expression* | [**community**] [**no-advertise**] [**no-export**] [**no-export-subconfed**]} [**vrf** vrf-name]	Displays the BGP routes that match a BGP community.
show bgp [**vrf** *vrf-name*] {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **community-list** *list-name* [**vrf** *vrf-name*]	Displays the BGP routes that match a BGP community list.
show bgp {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **extcommunity** {**regexp** *expression* | **generic** [**non-transitive** | **transitive**] *aa4:nn* [**exact-match**]} [**vrf** *vrf-name*]	Displays the BGP routes that match a BGP extended community.
show bgp {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **extcommunity-list** *list-name* [**exact-match**]} [**vrf** vrf-name]	Displays the BGP routes that match a BGP extended community list.
show bgp {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] {**dampening dampened-paths** [**regexp** *expression*]} [**vrf** vrf-name]	Displays the information for BGP route dampening. Use the **clear bgp dampening** command to clear the route flap dampening information.
show bgp {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **history-paths** [**regexp** *expression*] [**vrf** *vrf-name*]	Displays the BGP route history paths.
show bgp {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **filter-list** *list-name* [**vrf** *vrf-name*]	Displays the information for the BGP filter list.
show bgp {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **neighbors** [*ip-address* | *ipv6-prefix*] [**vrf** *vrf-name*]	Displays the information for BGP peers. Use the **clear bgp neighbors** command to clear these neighbors.
show bgp {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] {**nexthop** | **nexthop-database**} [**vrf** *vrf-name*]	show bgp {ipv4 | ipv6 | vpnv4 | vpnv6} {unicast | multicast} [ip-address | ipv6-prefix] {nexthop | nexthop-database} [vrf vrf-name]

show bgp paths	Displays the BGP path information.
show bgp {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **policy** *name* [**vrf** *vrf-name*]	Displays the BGP policy information. Use the **clear bgp policy** command to clear the policy information.
show bgp {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **prefix-list** *list-name* [**vrf** *vrf-name*]	Displays the BGP routes that match the prefix list.
show bgp {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **received-paths** [**vrf** *vrf-name*]	Displays the BGP paths stored for soft reconfiguration.
show bgp {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **regexp** *expression* [**vrf** *vrf-name*]	Displays the BGP routes that match the AS_path regular expression.
show bgp {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **route-map** *map-name* [**vrf** *vrf-name*]	Displays the BGP routes that match the route map.
show bgp peer-policy *name* [**vrf** *vrf-name*]	Displays the information about BGP peer policies.
show bgp peer-session *name* [**vrf** *vrf-name*]	Displays the information about BGP peer sessions.
show bgp peer-template *name* [**vrf** *vrf-name*]	Displays the information about BGP peer templates. Use the **clear bgp peer-template** command to clear all neighbors in a peer template.
show bgp process	Displays the BGP process information.
show {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} **bgp** *options*	Displays the BGP status and configuration information. This command has multiple options. See the *Cisco Nexus 7000 Series NX-OS Unicast Routing Command Reference*, for more information.
show {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} **mbgp** *options*	Displays the BGP status and configuration information. This command has multiple options. See the *Cisco Nexus 7000 Series NX-OS Unicast Routing Command Reference*, for more information.
show running-configuration bgp	Displays the current running BGP configuration.

Displaying Advanced BGP Statistics

To display advanced BGP statistics, use the following commands:

Command	Purpose
show bgp {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **flap-statistics** [**vrf** *vrf-name*]	Displays the BGP route flap statistics. Use the **clear bgp flap-statistics** command to clear these statistics.
show bgp sessions [**vrf** *vrf-name*]	Displays the BGP sessions for all peers. Use the clear bgp sessions command to clear these statistics.
show bgp statistics	Displays the BGP statistics.

Related Documents

Related Topic	Document Title
BGP CLI commands	**Cisco Nexus 7000 Series NX-OS Unicast Routing Command Reference**
VDCs and VRFs	**Cisco Nexus 7000 Series NX-OS Virtual Device Context Configuration Guide**

RFCs

RFC	Title
RFC 2918	Route Refresh Capability for BGP-4 http://www.faqs.org/rfcs/rfc2918.html

MIBs

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

Feature History for Advanced BGP

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

Table 1: Feature History for Advanced BGP

Feature Name	Release	Feature Information
BGP	7.3(0)D1(1)	Added support for exporting routes to Default VRF
BGP	6.2(8)	Added support for CISCO-BGP-MIBv2
BGP	6.2(8)	Added support for RFC 5549
BGP Next Hop Unchanged	6.2(8)	Introduced this feature.
BGP	6.2(2)	Added BFD support for the IPv6 address family.
BGP	6.2(2)	Added the ability to configure BGP to advertise the default route and introduced the **default-information originate** command.
BGP	6.2(2)	Added the ability to advertise routes that are suppressed by the **aggregate-address** command.
Policy-based administrative distance	6.2(2)	Introduced this feature.
BGP conditional route injection	6.2(2)	Introduced this feature.
BGP AS-path multipath relax	6.0(1)	Added the **as-path multipath-relax** option to the **bestpath** command.
BGP outbound route-maps	6.0(1)	Added support for setting next-hops on reflected routes using an outbound route-map.
BGP cost community ignore	5.2(1)	Added the **cost-community ignore** option to the **bestpath** command.
VPN address families	5.2(1)	Added support for VPN address families.
BGP	5.1(1)	No change from Release 5.0.
BFD	5.0(2)	Added support for BFD. See the *Cisco Nexus 7000 Series NX-OS Interfaces Configuration Guide* for more information.
ISSU	4.2(3)	Lowered the BGP minimum hold-time check to eight seconds.

Feature Name	Release	Feature Information
Next-hop addressing	4.2(1)	Added support for the BGP next-hop address tracking and filtering.
4-Byte AS numbers	4.2(1)	Added support for 4-byte AS numbers in plaintext notation.
Conditional advertisement	4.2(1)	Added support for conditionally advertising BGP routes based on the existence of other routes in the BGP table.
Dynamic AS number for prefix peers	4.1(2)	Added support for a range of AS numbers for the BGP prefix peer configuration.
BGP	4.0(1)	This feature was introduced.