Enabling and Setting up BGP

This chapter provides clear, step-by-step guidance for enabling and configuring BGP on your network devices. It covers essential setup tasks, best practices for managing BGP neighbors and route policies, and introduces multi-instance and multi-AS features to help you scale and optimize your network.

- Enable BGP routing, on page 1
- BGP multi-instance and multi-AS support, on page 3

Enable BGP routing

Enable BGP routing and establish BGP neighbor relationships to control route exchange in your network.

Configure BGP on your router to participate in inter-domain routing. You must define route policies and associate address families with each neighbor to control which routes are exchanged. This is essential for both internal BGP (iBGP) and eBGP peering.

Before you begin

- Ensure your BGP router has a router identifier, such as a configured loopback address.
- Configure at least one address family within the BGP router configuration, and explicitly configure the same address family under each BGP neighbor configuration.
- For an external BGP (eBGP) neighbor:
  - Configure both inbound and outbound route policies on the neighbor using the**route-policy** command, or
  - Allow route exchange for all eBGP neighbors by using the **bgp unsafe-ebgp-policy** command.
- If peering with eBGP neighbors that are not directly connected:
  - Note that, by default, BGP checks for direct connectivity and does not establish a connection if peers are not directly connected.
  - Use the **ignore-connected-check** command to enable peering between loopback interfaces of non-directly connected routers. Set the TTL value to 1 when using this command.
  - Configure the **egp-multihop** command for eBGP peers separated by intermediate routers, setting the TTL value to at least the number of hops between peers (for example, use TTL 3 for two intermediate hops).

Procedure

Step 1 Define a route policy to control routing behavior.

Example:

```
Router# configure
Router(config)# route-policy drop-as-1234
Router(config-rpl)# if as-path passes-through '1234' then
Router(config-rpl)# apply check-communities
Router(config-rpl)# else
Router(config-rpl)# pass
Router(config-rpl)# endif
Router(config-rpl)# end-policy
```

Step 2 Enter BGP configuration mode. Configure the router ID, and specify the address family to define the type of routes BGP will handle, such as IPv4 or IPv6 unicast.

Example:

```
Router# configure
Router(config)# router bgp 120
Router(config-bgp)# bgp router-id 192.168.70.24
Router(config-bgp)# address-family ipv4 unicast
Router(config-bgp-af)# exit
```

Step 3 Configure a BGP neighbor with an IP address, assign a remote AS number, and associate an address family. Optionally, configure route policies to control which routes are exchanged.

Example:

```
Router(config-bgp)# neighbor 172.168.40.24
Router(config-bgp-nbr)# remote-as 2002
Router(config-bgp-nbr)# address-family ipv4 unicast
Routing(config-bgp-nbr-af)# route-policy drop-as-1234 in
Routing(config-bgp-nbr-af)# commit
```

BGP routing is enabled. The router can now establish BGP neighbor relationships and exchange routing information as configured.

Example

Sample configuration to enable BGP:

```
route-policy pass-all
  pass
end-policy

route-policy set_next_hop_agg_v4
  set next-hop 10.0.0.1
end-policy

route-policy set_next_hop_static_v4
  if (destination in static) then
    set next-hop 10.1.0.1
  else
    drop
```

```
  endif
end-policy

route-policy set_next_hop_agg_v6
  set next-hop 2001:DB8::121
end-policy

route-policy set_next_hop_static_v6
  if (destination in static) then
    set next-hop 2001:DB8::201 else
  drop
  endif
end-policy

router bgp 65000
bgp fast-external-fallover disable
bgp confederation peers 65001 65002
bgp confederation identifier 1
bgp router-id 10.1.1.1
address-family ipv4 unicast
  aggregate-address 10.2.0.0/24 route-policy
set_next_hop_agg_v4
  aggregate-address 10.3.0.0/24
  redistribute static route-policy
set_next_hop_static_v4
  address-family ipv6 unicast
  aggregate-address 2001:DB8:2::/32 route-policy
set_next_hop_agg_v6
  aggregate-address 2001:DB8:3::/32
  redistribute static route-policy
set_next_hop_static_v6
  neighbor 10.0.101.60
    remote-as 65000
    address-family ipv4 unicast
  neighbor 10.0.101.61
    remote-as 65000
    address-family ipv4 unicast
  neighbor 10.0.101.62
    remote-as 3
    address-family ipv4 unicast
      route-policy pass-all in
      route-policy pass-all out
  neighbor 10.0.101.64
    remote-as 5
    update-source Loopback0
    address-family ipv4 unicast
      route-policy pass-all in
      route-policy pass-all out
```

What to do next

Verify BGP neighbor relationships and route exchange using the **show bgp summary** command.

BGP multi-instance and multi-AS support

BGP multi-instance and multi-AS are mechanisms that

- combine services from multiple routers into a single IOS XR router

- achieve address family (AF) isolation by configuring different address families in separate BGP instances, and
- increase session scale by distributing peering sessions across multiple BGP instances.

This feature enables each BGP instance on a router to operate with a unique Autonomous System (AS) number. This provides enhanced flexibility and scalability for complex network environments.

Benefits of BGP multi-instance and multi-AS

These are the benefits of the BGP multi-instance and multi-AS feature:

- **Simplified network management**: You can consolidate services from multiple routers into a single IOS XR router. This reduces overall complexity and decreases the effort needed to manage the network.
- **Enhanced control and isolation**: By isolating different address families (AFs) in separate BGP instances, you gain better traffic management and policy enforcement.
- **Scalability for large networks**: By distributing the load across multiple BGP instances, this feature supports a higher number of peering sessions and larger prefix tables.
- **Improved network stability**: By providing faster and more reliable BGP convergence in certain scenarios, this feature increases your network uptime and responsiveness.
- **High availability**: With full support for BGP features, including Nonstop Routing (NSR), your network runs continuously and remains resilient.

Requirements for configuring BGP multi-instance and multi-AS

Follow these requirements when configuring BGP multi-instance and multi-autonomous system (multi-AS) features:

Instance limits and identification

- Limit the router to a maximum of 4 BGP instances.
- Assign a unique router ID to each BGP instance.

Address family configuration

- Configure only one address family under each BGP instance. VPNv4, VPNv6, and RT-Constrain address families can be configured under multiple BGP instances.
- Place IPv4 or IPv6 unicast within the same BGP instance where you configure IPv4 or IPv6 labeled-unicast.
- Place IPv4 or IPv6 multicast within the same BGP instance where you configure IPv4 or IPv6 unicast.

Configuration management

- Commit all configuration changes for one BGP instance at the same time.
- Commit configuration changes for only one BGP instance at a time.

Operational recommendations

Use a unique BGP update-source in the default VRF for each BGP instance when you peer with the same remote router.

Configure multiple BGP instances for a specific autonomous system

Configure several BGP instances within a single autonomous system, assigning a unique router ID to each instance.

Before you begin

- Ensure you have administrative access to the router.
- Collect autonomous system (AS) numbers and router ID values you want to use.

Follow these steps to configure multiple BGP instances in a specific autonomous system:

Procedure

Step 1 Enter BGP instance configuration mode for the specific autonomous system and specify a unique name for each instance. Assign a unique router ID value to the BGP instance.

Example:

```
Router# configure
Router(config)# router bgp 100 instance inst1
Router(config-bgp)# bgp router-id 10.0.0.0
Router(config-bgp)# commit
```

Ensure that the router ID you assign is unique for each BGP instance in the autonomous system.

Restriction
Commit all configuration changes for each BGP instance separately. You cannot commit changes for multiple instances at the same time.

Step 2 Repeat step 1 for each additional BGP instance you want to configure within the same autonomous system, specifying a unique instance name and router ID each time.

Each BGP instance in the specified autonomous system has a unique router ID and its configuration is committed individually.