

Border Gateway Protocol

Table 1: Feature History

Feature Name	Release Information	Description
Border Gateway Protocol (BGP)	2025.04.0	BGP feature enables dynamic, loop-free inter-domain routing between autonomous systems, allowing you to prioritize service IP addresses and ensure high availability. It works by using BGP speaker pods to advertise service IP addresses for incoming traffic and learn routes for outgoing traffic. BGP peers establish TCP connections and exchange routing information, with preference values determining traffic flow, especially in active-standby configurations. This feature also supports handling pod failovers and offers various commands for configuration and monitoring. Command introduced: **router bgp** *local_as_number*—Used to configure the autonomous systems (AS) and IP address for the BGP router. **Default Setting**: Disabled – Configuration Required

- Dynamic Routing, on page 2
- Call Flows, on page 5
- Configuring Dynamic Routing Using BGP, on page 8
- Configuring BGP Speaker, on page 11
- Monitoring and Troubleshooting, on page 12

Dynamic Routing

Border Gateway Protocol (BGP) allows you to create loop-free inter-domain routing between autonomous systems (AS). An AS is a set of routers under a single technical administration. The routers can use an Exterior Gateway Protocol to route packets outside the AS. The Dynamic Routing by Using BGP feature enables you to configure the next-hop attribute of a BGP router with alternate local addresses to service IP addresses with priority and routes. The App-Infra BGP speaker pods enable dynamic routing of traffic by using BGP to advertise pod routes to the service VIP.

Key functionality of Dynamic Routing using BGP:

- Advertising Specific Service IPs: BGP advertises individual service IP addresses using a */32* netmask. This enables dynamic routing for each service IP.
- Next-Hop for Ingress Traffic: When BGP advertises a */32* service IP address, it includes a next-hop address. This next-hop is the service interface's IP address. External networks use this address to forward incoming traffic to the specific service IP. This process ensures efficient and dynamic traffic steering.
- Prioritized Routing via MED: Initially, Master1 advertises the service VIP route with a lower MED value (1210) because the VIP is active on it. Master2 also advertises the route but uses a higher MED value (1220) since the VIP is not active there. External BGP peers select the route from Master1 because it has the lower MED, so incoming traffic goes to Master1. For iBGP, BGP speakers use local preference to prioritize routes.
- Dynamic Failover and Traffic Redirection using VIP Monitoring: If the GTP/Protocol pod on Master1 goes down, the VIP becomes unavailable on Master1 and moves to Master2. Both BGP monitors detect this change. Master1's BGP speaker advertises the VIP route with the higher MED (1220), and Master2's BGP speaker advertises it with the lower MED (1210). External nodes receive these updates and re-evaluate their paths. Incoming traffic then redirects seamlessly to Master2, enabling smooth failover.
- Dynamic Route Learning and Kernel Integration: BGP dynamically learns network prefixes and their next-hop IP addresses from other BGP peers. The learned routes are installed into the kernel routing table. This makes them available for forwarding outgoing traffic
- Policy-Driven Route Filtering for Control: BGP peers may advertise many routes, but not all are accepted or installed in the local kernel routing table. BGP implementation uses routing policies to filter incoming BGP advertisements. This selective acceptance allows BGP speaker to control which IP prefixes the network learns and uses.
- Automatic Route Re-installation on Interface Recovery: If a network interface goes down, the system removes all dependent routes from the kernel routing table. BGP speaker continuously monitors network interfaces. When a previously down interface returns to service, the BGP speaker automatically reinstalls the relevant routes, restoring network connectivity.
- Policy-Driven Static Route Configuration: BGP speaker uses specific policies to configure static routes manually. These static routes provide fixed, predefined paths to destinations.
- High Availability through Bonded Interfaces and Service IP: The BGP speaker's service IP is assigned to bonded interfaces for redundancy. Each bond includes multiple physical interfaces in an active/standby setup. Each physical interface connects to a BGP router, ensuring highly available network paths for the BGP speaker.

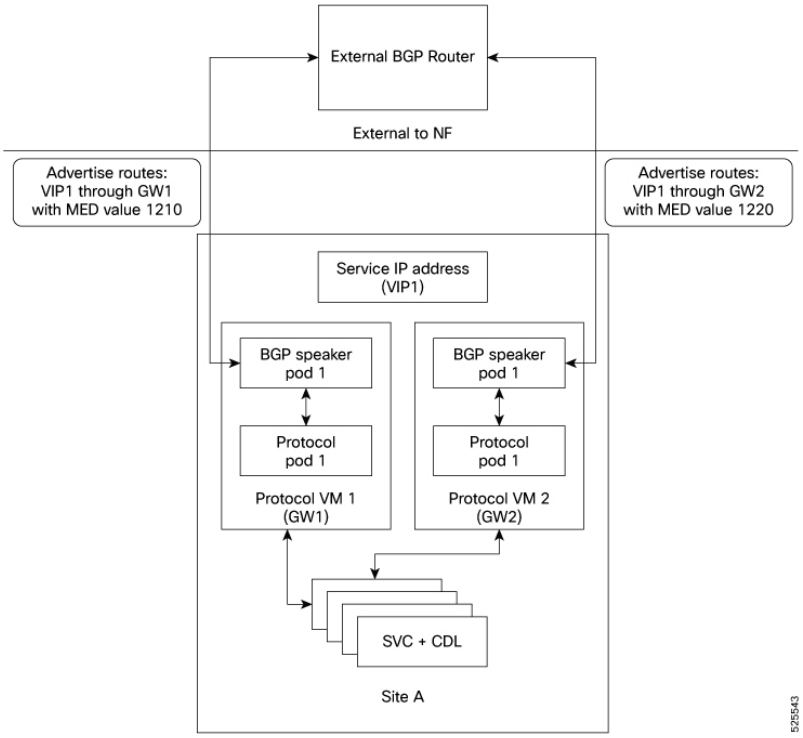
- BGP Peering-Triggered Bond Failover: If BGP peering sessions on the active interface go down, the BGP speaker initiates a bond switchover. This promotes a standby interface to active status. Traffic then redirects seamlessly, maintaining continuous BGP communication.
- Debugging and Troubleshooting through KPI & Log Messages: BGP speakers provide comprehensive statistics and Key Performance Indicators (KPIs) for monitoring. They also generate detailed log messages, which are crucial for debugging and troubleshooting BGP issues.
- BGP MED-Driven Geo-Active/Standby Deployments: In a multi-HA setup (for example, GEO Rack 1 is active and GEO Rack 2 is standby), BGP speakers on active Rack 1 advertise service IPs with lower MED values (1210 and 1220). Speakers on standby Rack 2 use higher MED values (2210 and 2220). BGP's preference for lower MED values directs all traffic to the active rack.
- Monitoring GEO Roles: BGP continuously monitors the active and standby roles of GEO racks. When a role changes, BGP dynamically re-advertises service IP routes with updated MED values. This ensures traffic always goes to the currently active BGP speaker.
- BGP Speaker Status Management: BGP speaker pods report their status to a central Geo pod via ETCD. The status depends on the health of their BGP peerings. If a BGP speaker pod loses all peerings, it is marked as "down" and isolated. Traffic then shifts automatically to the other High Availability (HA) BGP speaker pod. If both HA BGP speaker pods in a GEO rack lose all peerings, the entire rack is marked as "down" in ETCD and set to "standby" by the Geo pod.
- BGP Speaker Pod Architecture with BFD for Accelerated Failover: Each BGP speaker pod contains two containers: a BGP speaker container for routing protocol operations, and a Bidirectional Forwarding Detection (BFD) container. The BFD container quickly detects link or peer failures. This rapid detection allows BGP to respond almost instantly to connectivity issues, reducing failover times and improving network resilience.

Incoming Traffic

BGP uses TCP as the transport protocol, on port 179. Two BGP routers form a TCP connection between one another. These routers are peer routers. The peer routers exchange messages to open and confirm the connection parameters.

The BGP speaker publishes routing information of the protocol pod for incoming traffic in the active standby mode. Use the following image as an example to understand the dynamic routing functionality. There are two protocol pods, pod1 and pod2. Pod1 is active and pod2 is in the standby mode. The service IP address vip1 is configured on both the nodes, on host IP1 and host IP2. BGP pod1 is running on host IP1 and BGP pod2 on host IP2. The host IP address exposes the pod services. BGP speaker publishes the route vip1 through host IP1 and host IP2. It also publishes the preference values, 110 and 100 to determine the priority of pods.

Figure 1: Dynamic Routing for Incoming Traffic in the Active-standby Topology



For high availability, each cluster has two BGP speaker pods with Active-standby topology. Kernel route modification is done at host network level where the protocol pod runs.

MED Value

The Local Preference is used only for IGP neighbours, whereas the MED Attribute is used only for EGP neighbours. A lower MED value is the preferred choice for BGP.

Table 2: MED Value

Bonding Interface Active	VIP Present	MED Value	Local Preference
Yes	Yes	1210	2220
Yes	No	1220	2210
No	Yes	1215	2215
No	No	1225	2205

Bootstrap of BGP Speaker Pods

The following sequence of steps set up the BGP speaker pods:

1. The BGP speaker pods use TCP as the transport protocol, on port 179. These pods use the AS number configured in the Ops Center CLI.
2. Register the Topology manager.

3. Select the Leader pod. The Active speaker pod is the default choice.
4. Establish connection to all the BGP peers provided by the Ops Center CLI.
5. Publish all existing routes from ETCD.
6. Configure import policies for routing by using CLI configuration.
7. Start gRPC stream server on both the speaker pods.
8. Similar to the cache pod, two BGP speaker pods must run on each Namespace.

Call Flows

This section describes the key call flows for Dyanmic Routing by Using BGP.

Publish Route for Incoming Traffic in an Active-Standby Mode

The following sections describe the Control Plane and Data Plane call flows in an active/standby mode.

Control Plane Call Flow

This section describes the Control Plane call flow.

Figure 2: Control Plane Call Flow

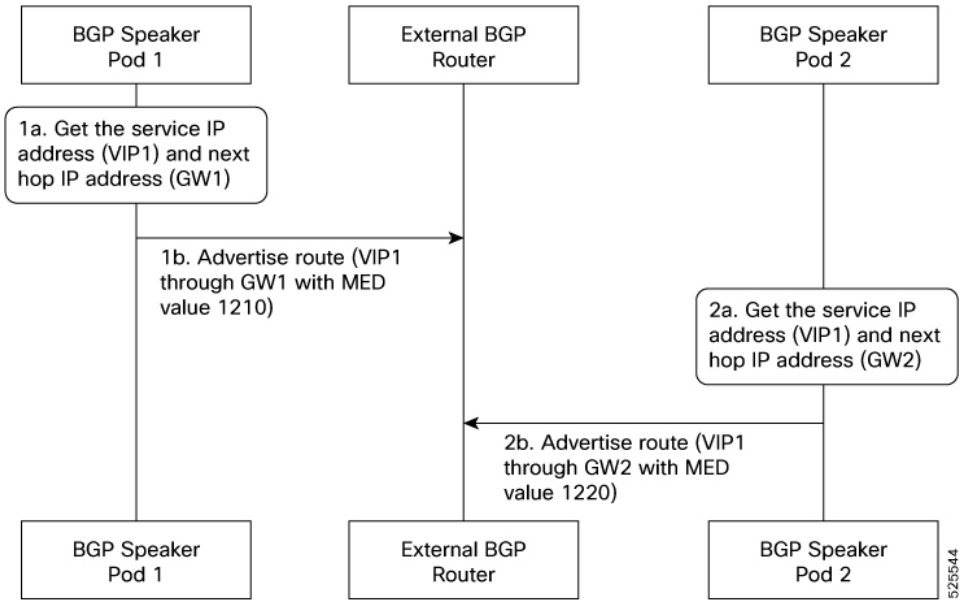


Table 3: Control Plane Call Flow Description

Step	Description
1	The BGP speaker pod starts and fetches the service IP address, next-hop IP address (host IP or loopbackEth), and the Instance ID for the BGP speaker pod. The pod service is exposed through host IP or configured loopbackEth. The NF Instance ID is used to find the route priority or preference.
2	The BGP speaker pod advertises routes by fetching vip-ip (service IP addresses) from the Ops Center.

Data Plane Call Flow

This section describes the data plane call flow.

Figure 3: Data Plane Call Flow

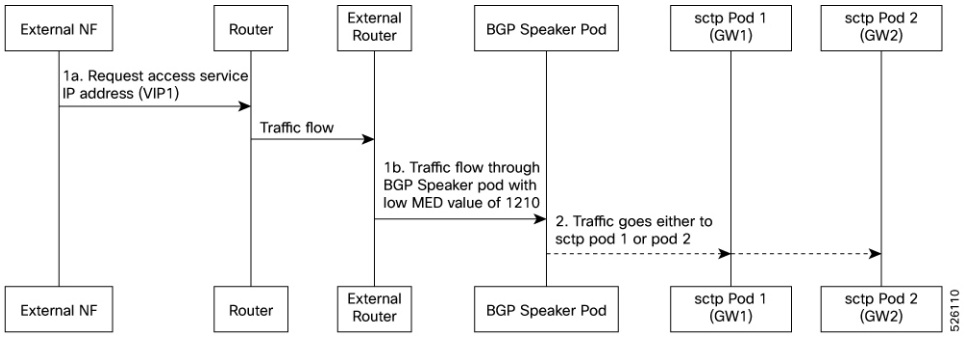


Table 4: Data Plane Call Flow Description

Step	Description
1	External NF requests for service IP address. The request is sent to the nearest connected router through multiple external routers. Then, the router sends the request to the BGP speaker pod with highest priority.
2	The BGP router sets the data plane flow based on the preference value. In the preceding call flow example, the router routes the service request through the host IP1 to pod 1 due to its higher preference value. From host IP1, traffic is forwarded to either sctp pod 1 or pod 2.

Single Protocol Pod Failure Call Flow

The following section describes the Single Protocol Pod Failure call flow.

Figure 4: Single Protocol Pod Failure Call Flow

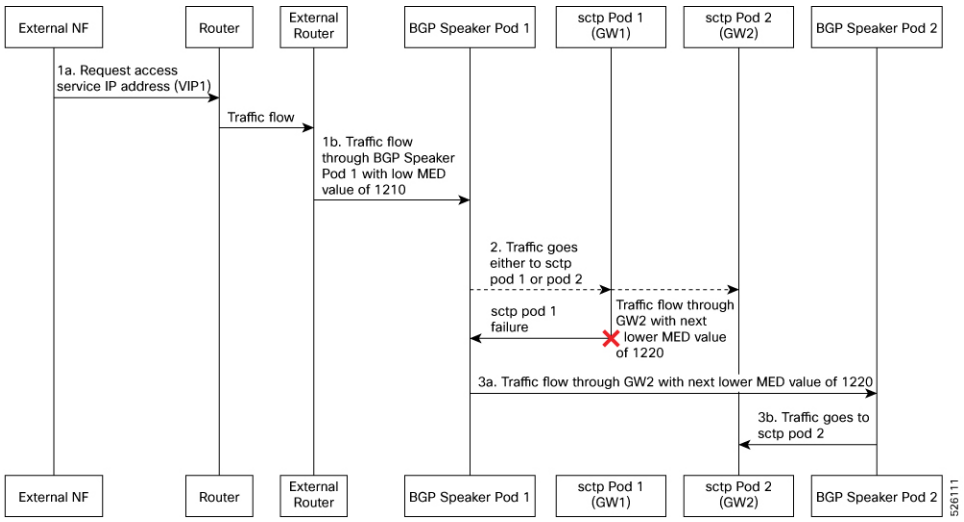


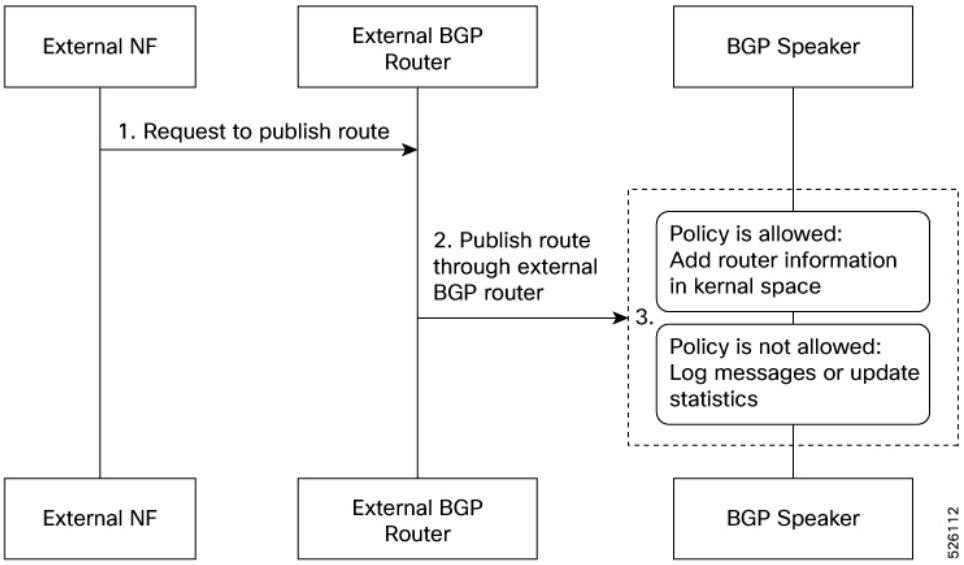
Table 5: Single Protocol Pod Failure Call Flow Description

Step	Description
1	External NF requests for service IP address. The request is sent to the nearest connected BGP router through multiple external routers based on the next highest preference value.
2	The BGP router sets the data plane flow based on the preference value. If the pod with the highest preference value is not available, then the request is routed to the pod with the next highest preference value. In the example shown in the preceding call flow figure, BGP pod 2 with the host IP2 address serves the request due to its higher preference value.

Learn Route for Outgoing Traffic Call Flow

This section describes the Learn route for outgoing traffic call flow.

Figure 5: Learn Route for Outgoing Traffic Call Flow



External NF or other systems advertise route to the external BGP route. In turn, the external BGP router advertises routes for its service through BGP.

Table 6: Learn Route for Outgoing Traffic Call Flow Description

Step	Description
1	The BGP speakers receive the routing information.
2	Learn the route by using the BGP protocol.
3	Based on the configure policy, the system either checks the routing information or ignores it.
4	If the policy is not allowed, then the system logs the messages and updates the statistics.
5	The protocol pods configures the route in Kernel space on host through the netlink go APIs.

Configuring Dynamic Routing Using BGP

This section describes how to configure the dynamic routing using BGP.

Configuring AS and BGP Router IP Address

To configure the AS and IP address for the BGP router, use the following commands:

```
config
   router bgp local_as_number
   exit
exit
```

NOTES:

- **router bgp** *local_as_number*—Specify the identification number for the AS for the BGP router.

Configure Router ID (Optional)

By default, the BGP speaker uses the IPv4 address of the BGP server running on an interface as the Router ID. However, you can configure a custom Router ID if desired.

To configure a custom Router ID, assign an IP address to the loopback (lo) interface on the node where the BGP Speaker pod is running. This assigned IP address will be used as the Router ID.



Note Configuring a Router ID is mandatory if you plan to run the BGP server exclusively with IPv6 addresses (i.e., without any IPv4 address on the interface where the BGP server is running).

Configuring BGP Service Listening IP Address

To configure the BGP service listening IP address, use the following commands:

```
config
   router bgp local_as_number
      interface interface_name
   exit
exit
```

NOTES:

- **router bgp** *local_as_number*—Specify the identification number for the AS for the BGP router.
- **interface** *interface_name*—Specify the name of the interface.

Configuring BGP Neighbors

To configure the BGP neighbors, use the following commands:

```
config
   router bgp local_as_number
      interface interface_name
         neighbor neighbor_ip_address remote-as as_number
         exit
      exit
   exit
```

NOTES:

- **router bgp** *local_as_number*—Specify the identification number for the AS for the BGP router.
- **interface** *interface_name*—Specify the name of the interface.
- **neighbor** *neighbor_ip_address*—Specify the IP address of the neighbor BGP router.
- **remote-as** *as_number*—Specify the identification number for the AS.

Configuring Bonding Interface

To configure the bonding interface related to the interfaces, use the following commands:

```
config
   router bgp local_as_number
      interface interface_name
         bondingInterface interface_name
         exit
      exit
   exit
```

NOTES:

- **router bgp** *local_as_number*—Specify the identification number for the AS for the BGP router.
- **interface** *interface_name*—Specify the name of the interface.
- **bondingInterface** *interface_name*—Specify the related bonding interface for an interface. If the bonding interface is active, then the BGP gives a higher preference to the interface-service by providing a lower MED value.

Configuring Learn Default Route

If the user configures specific routes on their system and they need to support all routes, then they must set the **learnDefaultRoute** as **true**.



Note This configuration is optional.

To configure the Learn Default Route, use the following commands:

```
config
   router bgp local_as_number
      learnDefaultRoute true/false
   exit
exit
```

NOTES:

- **router bgp** *local_as_number*—Specify the identification number for the AS for the BGP router.
- **learnDefaultRoute** *true/false*—Specify the option to enable or disable the **learnDefaultRoute** parameter. When set to true, BGP learns default route and adds it in the kernel space. By default, it is false.

Configuring BGP Port

To configure the Port number for a BGP service, use the following commands:

```
config
   router bgp local_as_number
      loopbackPort port_number
   exit
exit
```

NOTES:

- **router bgp** *local_as_number*—Specify the identification number for the AS for the BGP router.
- **loopbackPort** *port_number*—Specify the port number for the BGP service. The default value is 179.

Policy Addition

The BGP speaker pods learns many route information from its neighbors. However, only a few of them are used for supporting the outgoing traffic. This is required for egress traffic handling only, when is sending information outside to external NF. Routes are filtered by configuring import policies on the BGP speakers and is used to send learned routes to the protocol pods.

A sample CLI code for policy addition and the corresponding descriptions for the parameters are shown below.

```
$bgp policy <policy_Name> ip-prefix 209.165.200.225 subnet 16 masklength-range 21..24
as-path-set "^65100"
```

Table 7: Import Policies Parameters

Element	Description	Example	Optional
as-path-set	AS path value	"^65100"	Yes
ip-prefix	Prefix value	"209.165.200.225/16"	Yes
masklength-range	Range of length	"21..24"	Yes
interface	Interface to set as source IP (default is VM IP)	eth0	Yes
gateWay	Change gateway of incoming route	209.165.201.30	Yes
modifySourceIp	Modify source ip of incoming route Default value is False.	true	Yes
isStaticRoute	Flag to add static IP address into kernel route Default value is False.	true	Yes

Configuring BGP Speaker

This configuration controls the number of BGP speaker pods in deployment. BGP speaker advertises service IP information for incoming traffic from both the racks.

Note

- Use non-bonded interface in BGP speaker pods for BGP peering.
- BGP peering per Proto node is supported with only two BGP routers/leafs. Considering two Proto nodes, there can be maximum of four BGP neighborships.

```
instance instance-id  instance_id endpoint bgpspeaker interface { bgp | bfd
 } internal base-port start base_port_number
```

```
config
instance instance-id instance_id
 endpoint bgpspeaker
   replicas replica_id
   nodes node_id
   interface bgp
     internal base-port start base_port_number
   exit
   interface bfd
     internal base-port start base_port_number
  exit
exit
```

NOTES:

- **instance instance-id** *instance_id*—Specify the GR instance ID.
- *base_port_number*—Specify the port range only if logical NF is configured. This range depends on your deployment.

Example

The following is a configuration example:

```
instance instance-id 1
endpoint bgpspeaker
   replicas 1
   nodes    2
   interface bgp
     internal base-port start {24000}
   exit
   interface bfd
     internal base-port start {25000}
 exit
```

Monitoring and Troubleshooting

This section describes the show commands that are supported by the Dynamic Routing by Using BGP feature.

show bgp-kernel-route

Use the **show bgp-kernel-route** command to view all the kernel level routes for a BGP router.

The following configuration is a sample output of the **show bgp-kernel-route** command:

```
kernel-route

-----bgpspeaker-pod-1 ----

 DestinationIP       SourceIP        Gateway

 209.165.200.235     209.165.200.239   209.165.200.239

-----bgpspeaker-pod-2 ----
```

```
 DestinationIP       SourceIP        Gateway

 209.165.200.235     209.165.200.229   209.165.200.244
```

show bgp-global

Use the **show bgp-global** command to view all BGP global configurations.

The following configuration is a sample output of the **show bgp-global** command:

```
global-details

-----bgpspeaker-pod-1 ----
AS:        65000
Router-ID: 209.165.200.239
Listening Port: 179, Addresses: 209.165.200.239
AS:        65000
Router-ID: 209.165.200.232
Listening Port: 179, Addresses: 209.165.200.232

-----bgpspeaker-pod-2 ----
AS:        65000
Router-ID: 209.165.200.235
Listening Port: 179, Addresses: 209.165.200.235
AS:        65000
Router-ID: 209.165.200.246
Listening Port: 179, Addresses: 209.165.200.246
```

show bgp-neighbors

Use the **show bgp-neighbors** command to view all BGP neighbors for a BGP router.

The following configuration is a sample output of the **show bgp-neighbors** command:

```
neighbor-details

-----bgpspeaker-pod-2 ----
Peer              AS  Up/Down State       |#Received  Accepted
209.165.200.244 60000 00:34:20 Establ      |       10        10
Peer           AS  Up/Down State       |#Received  Accepted
209.165.200.250 60000 00:34:16 Establ      |        3         3

-----bgpspeaker-pod-1 ----
Peer              AS  Up/Down State       |#Received  Accepted
209.165.200.244 60000 00:33:53 Establ      |       10        10
Peer           AS  Up/Down State       |#Received  Accepted
209.165.200.250 60000 00:33:53 Establ      |        3         3
```

show bgp-neighbors *ip*

Use the **show bgp-neighbors** *ip* command to view details of a neighbor for a BGP router.

The following configuration is a sample output of the **show bgp-neighbors** *ip* command:

```
neighbor-details

-----bgpspeaker-pod-1 ----
BGP neighbor is 209.165.200.244, remote AS 60000
  BGP version 4, remote router ID 209.165.200.244
  BGP state = ESTABLISHED, up for 00:34:50
  BGP OutQ = 0, Flops = 0
  Hold time is 90, keepalive interval is 30 seconds
  Configured hold time is 90, keepalive interval is 30 seconds
```

```
  Neighbor capabilities:
    multiprotocol:
        ipv4-unicast:   advertised and received
    route-refresh:      advertised and received
    extended-nexthop:   advertised
        Local:  nlri: ipv4-unicast, nexthop: ipv6
    4-octet-as: advertised and received
  Message statistics:
                         Sent       Rcvd
    Opens:                  1          1
    Notifications:          0          0
    Updates:                1          2
    Keepalives:            70         70
    Route Refresh:          0          0
    Discarded:              0          0
    Total:                 72         73
  Route statistics:
    Advertised:             0
    Received:              10
    Accepted:              10

-----bgpspeaker-pod-2 ----
BGP neighbor is 209.165.200.244, remote AS 60000
  BGP version 4, remote router ID 209.165.200.244
  BGP state = ESTABLISHED, up for 00:35:17
  BGP OutQ = 0, Flops = 0
  Hold time is 90, keepalive interval is 30 seconds
  Configured hold time is 90, keepalive interval is 30 seconds

  Neighbor capabilities:
    multiprotocol:
        ipv4-unicast:   advertised and received
    route-refresh:      advertised and received
    extended-nexthop:   advertised
        Local:  nlri: ipv4-unicast, nexthop: ipv6
    4-octet-as: advertised and received
  Message statistics:
                         Sent       Rcvd
    Opens:                  1          1
    Notifications:          0          0
    Updates:                1          2
    Keepalives:            71         71
    Route Refresh:          0          0
    Discarded:              0          0
    Total:                 73         74
  Route statistics:
    Advertised:             0
    Received:              10
    Accepted:              10
```

show bgp-route-summary

Use the **show bgp-route-summary** command to view all the route details of a BGP router.

The following configuration is a sample output of the **show bgp-route-summary** command:

```
route-details

-----bgpspeaker-pod-1 ----
Table afi:AFI_IP safi:SAFI_UNICAST
Destination: 5, Path: 5

-----bgpspeaker-pod-2 ----
```

```
Table afi:AFI_IP safi:SAFI_UNICAST
Destination: 5, Path: 5
```

show bgp-routes

Use the **show bgp-routes** command to view all the routes for a BGP router.

The following configuration is a sample output of the **show bgp-routes** command:

```
bgp-route

-----bgpspeaker-pod-1 ----
   Network              Next Hop             AS_PATH       Age        Attrs
*> 209.165.200.235/24   209.165.200.250      60000         00:36:39   [{Origin: i} {Med:
0}]
*> 209.165.200.227/32   209.165.200.232                 00:36:44   [{Origin: e} {LocalPref:
 220} {Med: 3220}]
*> 209.165.200.247/24   209.165.200.250      60000         00:36:39   [{Origin: i} {Med:
0}]
*> 209.165.200.251/24   209.165.200.250      60000         00:36:39   [{Origin: i} {Med:
0}]
*> 209.165.200.252/32   209.165.200.232                 00:36:44   [{Origin: e} {LocalPref:
 220} {Med: 3220}]

-----bgpspeaker-pod-2 ----
   Network              Next Hop             AS_PATH        Age        Attrs
*> 209.165.200.235/24   209.165.200.250      60000          00:37:02   [{Origin: i} {Med:
0}]
*> 209.165.200.227/32   209.165.200.246                     00:37:11   [{Origin: e}
{LocalPref: 220} {Med: 3220}]
*> 209.165.200.228/24   209.165.200.234      60000          00:37:02   [{Origin: i} {Med:
0}]
*> 209.165.200.229/24   209.165.200.234      60000          00:37:02   [{Origin: i} {Med:
0}]
*> 209.165.200.230/32   209.165.200.246                     00:37:11   [{Origin: e}
{LocalPref: 220} {Med: 3220}]
```

KPIs

The following KPIs are supported for this feature:

Table 8: Statistics for Dynamic Routing by Using BGP

KPI Name	Type	Description/Formula	Label
bgp_outgoing_route request_total	Counter	Total number of outgoing routes.	local_pref, med, next_hope, service_IP
bgp_outgoing_failedroute request_total	Counter	Total number of failed outgoing routes.	local_pref, med, next_hope, service_IP
bgp_incoming_route request_total	Counter	Total number of incoming routes.	interface, next_hope, service_IP
bgp_incoming_failedroute request_total	Counter	Total number of failed incoming routes.	interface, next_hope, service_IP

KPI Name	Type	Description/Formula	Label
bgp_peers_total	Counter	Total number of peers added.	peer_ip, as_path
bgp_failed_peerstotal	Counter	Total number of failed peers.	peer_ip, as_path, error