

Dynamic Routing by Using BGP

- Feature Summary and Revision History, on page 1
- Feature Description, on page 2
- How it Works, on page 2
- Configuring Dynamic Routing Using BGP, on page 9
- Monitoring and Troubleshooting, on page 12

Feature Summary and Revision History

Summary Data

Table 1: Summary Data

Applicable Product(s) or Functional Area	cnSGW-C
Applicable Platform(s)	SMI
Default Setting	Disabled – Configuration required to enable
Related Changes in this Release	Not Applicable
Related Documentation	Not Applicable

Revision History

Table 2: Revision History

Revision Details	Release
First introduced.	2021.02.0

Feature Description

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 SMF BGP speaker pods enable dynamic routing of traffic by using BGP to advertise pod routes to the service VIP.

This feature supports the following functionality:

- Dynamic routing by using BGP to advertise service IP addresses for the incoming traffic.
- Learn route for outgoing traffic.
- Handling a BGP pod failover.
- Handling a protocol pod failover.
- Statistics and KPIs for the BGP speakers.
- Log messages for debugging the BGP speakers.
- Enable or disable the BGP speaker pods.
- New CLI commands to configure BGP.

How it Works

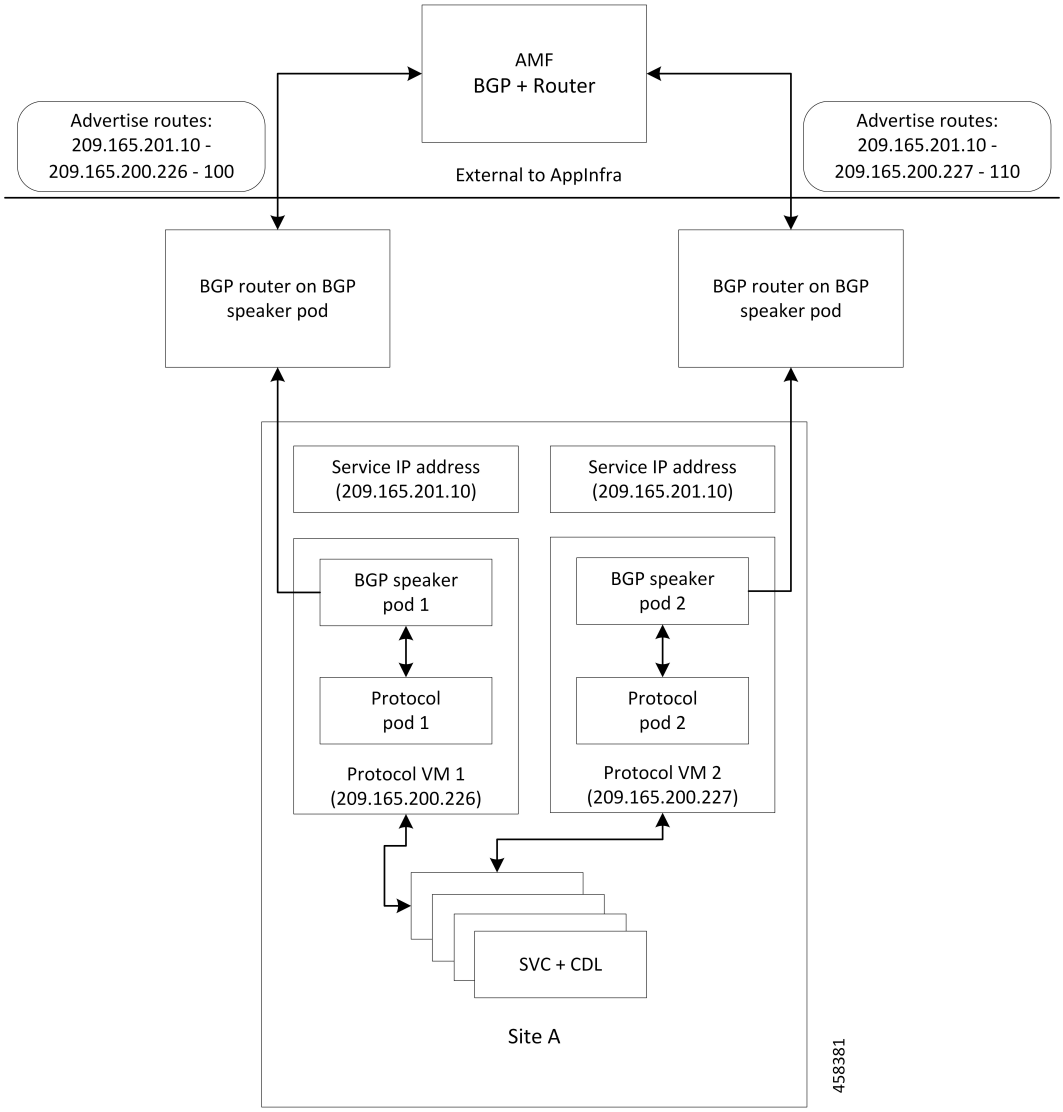
This section describes the operation of the Dynamic Routing feature.

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, 209.165.201.10 is configured on both the nodes, 209.165.200.226 and 209.165.200.227. Pod1 is running on host 209.165.200.226 and pod2 on host 209.165.200.227. The host IP address exposes the pod services. BGP speaker publishes the route 209.165.201.10 through 209.165.200.226 and 209.165.200.227. 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 neighbors, whereas the MED Attribute is used only for EGP neighbors. A lower MED value is the preferred choice for BGP.

Table 3: MED Value

Bonding Interface Active	VIP Present	MED Value	Local Preference
Yes	Yes	1210	2220
Yes	No	1220	2210

Bonding Interface Active	VIP Present	MED Value	Local Preference
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 that is 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.

External Network Failure

The NF instance start-up causes the BGP Speaker K8s pod to configure the next-hop attribute of the BGP router with alternate local addresses to service IP addresses with priority and routes.

After the Geo HA is triggered, the path selection is based on the destination service IP address, path connectivity and the priority value.

Note The subscriber sessions are not impacted because of the transparent migration between pods.

Geo Switchover

The achieves geo switchover by transparently migrating service IP address to mated peer K8s cluster, rack collocated, or geo-located. During the NF start-up, all the K8s cluster Namespaces register with the next-hop BGP router to advertise its service IP address and local IP address along with the priority and route modifier values.

Each logical NF exposes separate NF instance toward NRF or DNS, separate configuration, and separate LCM for a Namespace.

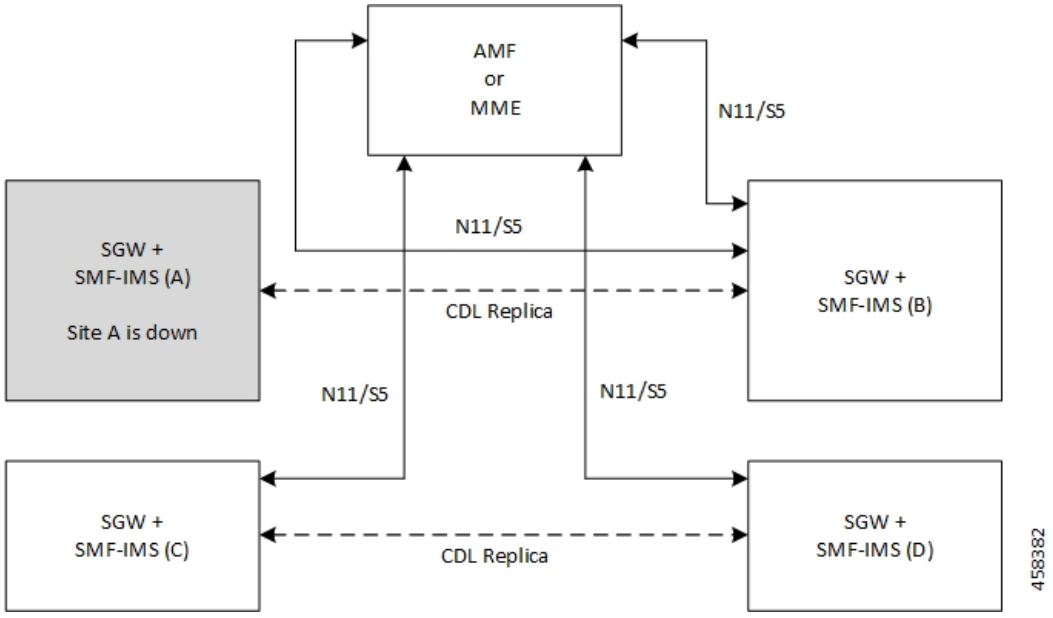
Internal Network Failure

If a functioning K8s cluster has an internal network failure due to a disrupted server communication with the master node, BFD failure, or a K8s pod networking issue, Geo HA is triggered due to K8s dependency checks that are based on the K8s liveliness failure.

In the example shown in the following figure, the AMF or MME transparently starts using the alternate rack server. The N11/S11/S5 and N4/Sxa service addresses are migrated to site B rack B. The system continues signalling from rack B for rack A. At rack B, the session continues without any impact to existing subscriber sessions.

Note Few in-transit calls might fail depending on the state where it is terminated before the UE re-attaches.

Figure 2: Geo HA for Internal Network Failure



Local Switchover

The achieves geo switchover by transparently migrating service IP address to mated peer K8s cluster or rack collocated within the same data center. During the NF start-up, all the K8s cluster Namespaces register with the next-hop BGP router to advertise its service IP address and local IP address along with the priority and route modifier values. Each logical NF exposes separate NF instance toward NRF or DNS, separate configuration, and separate LCM for a Namespace.

Recovery and Failback

For a seamless failover and failback, the UE sessions and the corresponding service IP addresses are grouped together.

The following scenarios describe the seamless failover and failback mechanism for the UE sessions:

- **Normal** - The UE sessions set is created, updated, or deleted from first rack and replicated to second rack.
- **Failure** - The UE sessions set is created, updated, or deleted from second rack and is not replicated to first rack due to its unavailability.
- **Recovery** - The CDL for first rack performs an auto-sync with the CDL for second rack to recover all the UE session data. During the recovery, the second rack continues to handle traffic from the sessions set.

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 3: Control Plane Call Flow

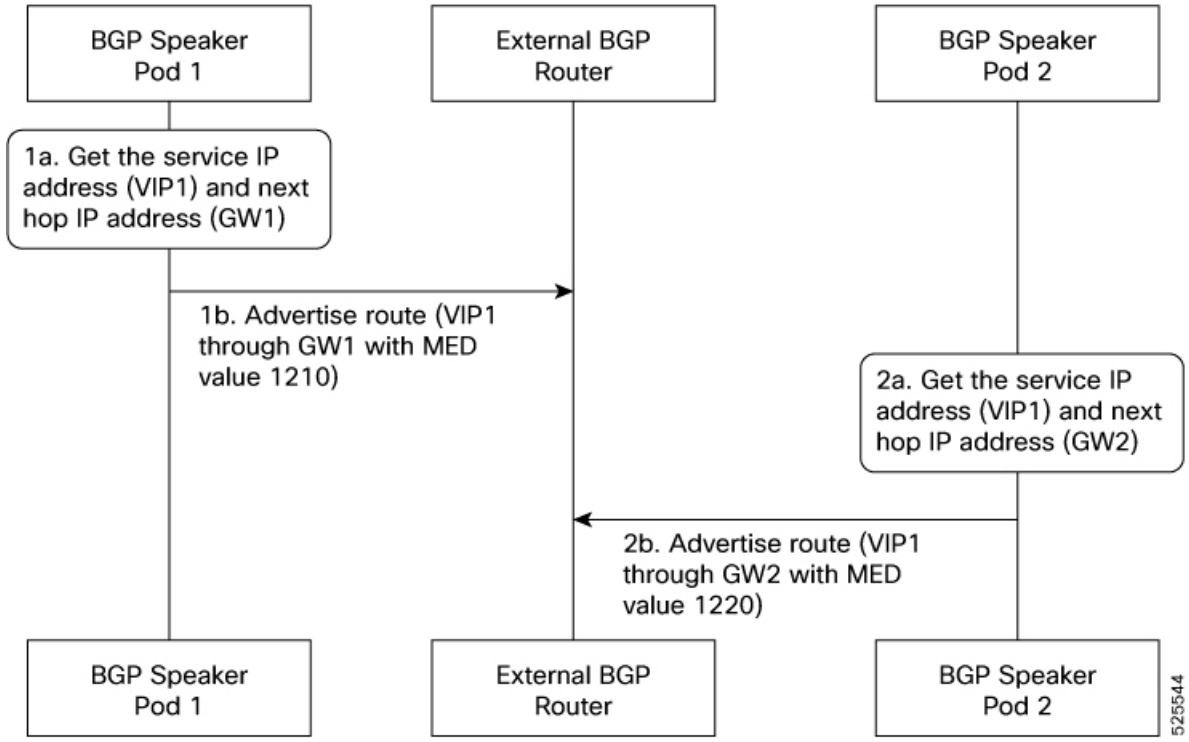


Table 4: Control Plane Call Flow Description

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 4: Data Plane Call Flow

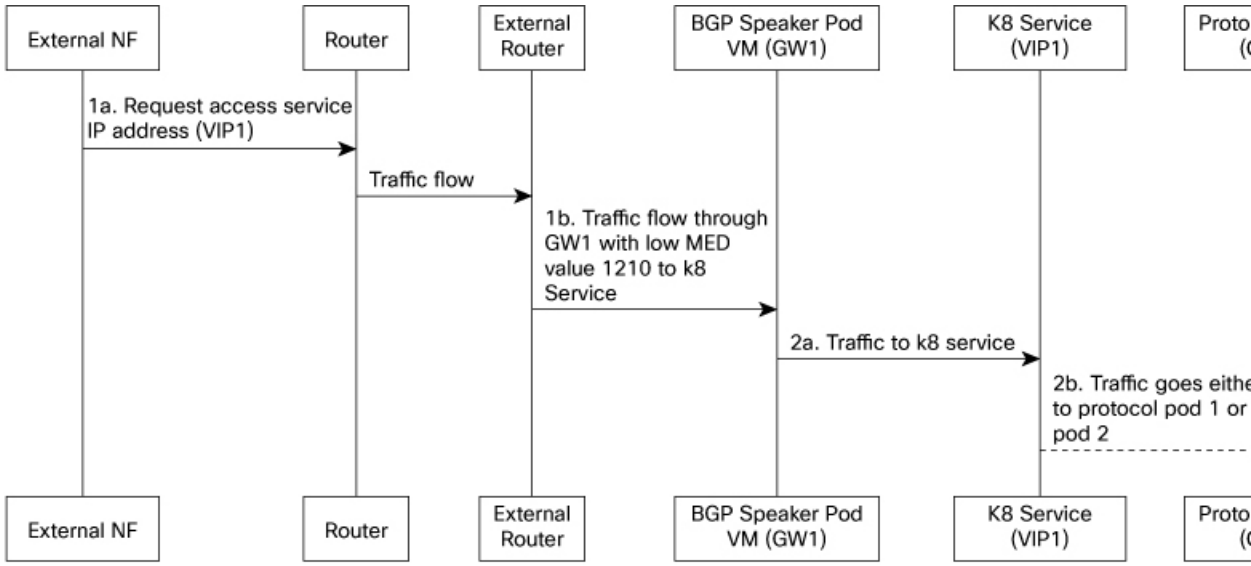


Table 5: Data Plane Call Flow Description

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 5: Single Protocol Pod Failure Call Flow

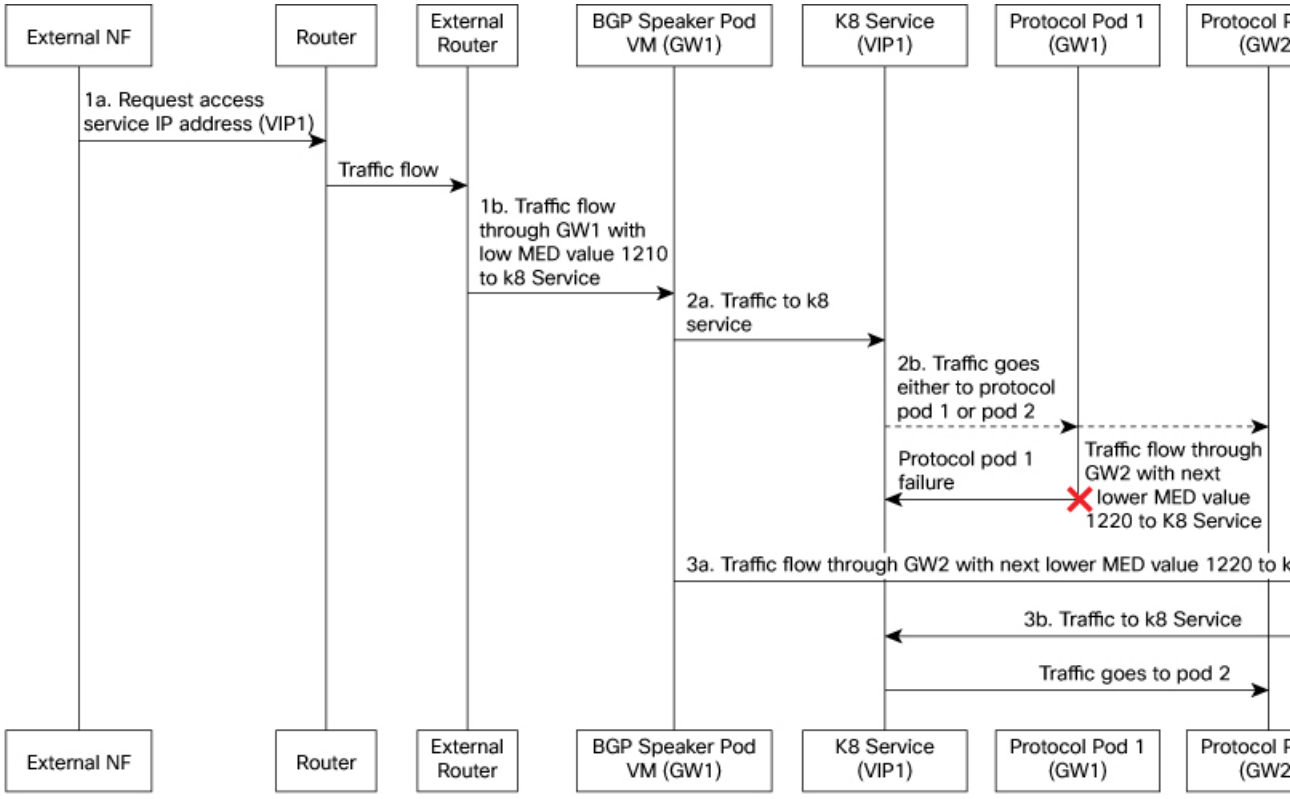


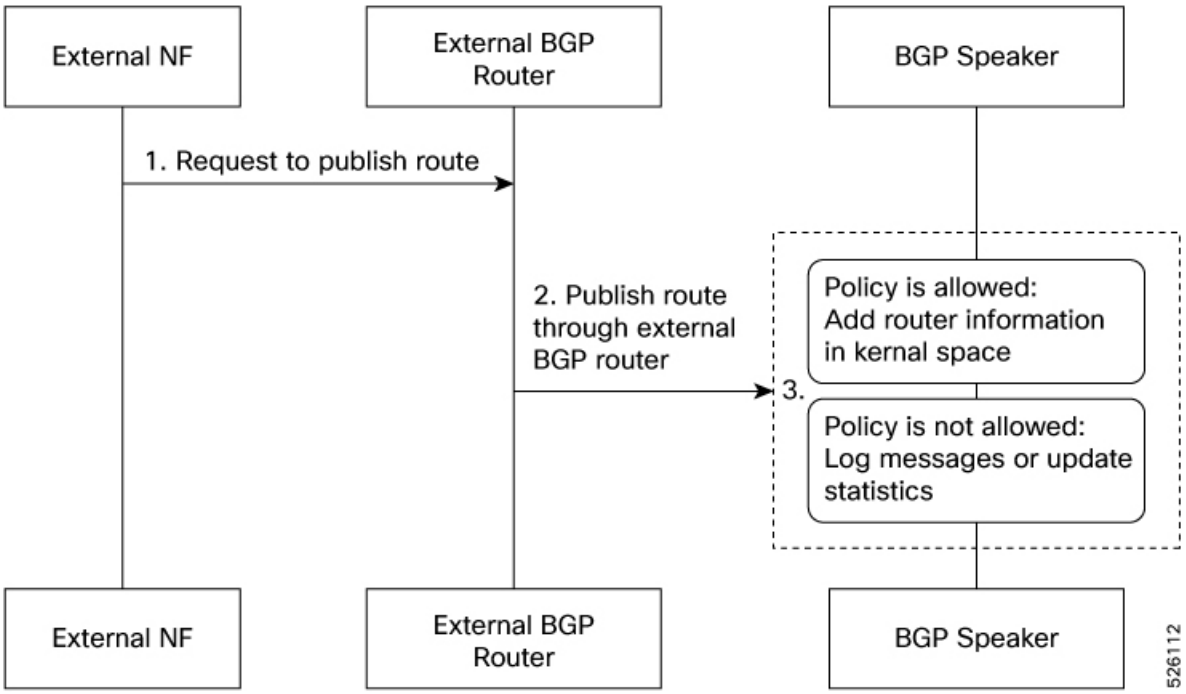
Table 6: Single Protocol Pod Failure Call Flow Description

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 6: Learn Route for Outgoing Traffic Call Flow



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

Table 7: 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.

  In a inter-rack redundancy deployment, you need to configure two Autonomous Systems (AS).

  - One AS for leaf and spine.
  - Second AS for both racks: Rack-1 and Rack-2.

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. BGP speaker uses this IPv4 address as the Router ID for BGP peering for both IPv4 BGP servers and Dual-mode BGP servers.

To configure a custom Router ID, assign an IPv4 address to the loopback (lo) interface on the node where the BGP Speaker pod is running. This assigned IPv4 address is then 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
```

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
```

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 cnSGW-C is sending information outside to AMF/PCF. 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 8: 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

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 9: 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

KPI Name	Type	Description/Formula	Label
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
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