



BGP Configuration Guide

BGP Configuration Guide version 2019.1.0
24 October 2019

www.rtbrick.com

Headquarters

N-17L, 1st Floor, 18th Cross Rd
Sector 3, HSR Layout
Bengaluru, Karnataka
India 560102

+91 80 4850 5445

www.rtbrick.com

Support

+TBC

support@rtbrick.com

Sales

+91 80 4850 5445

sales@rtbrick.com

©Copyright 2019 RtBrick, Inc. All rights reserved. The information contained herein is subject to change without notice. The trademarks, logos and service marks ("Marks") displayed in this documentation are the property of RtBrick in the United States and other countries. Use of the Marks are subject to RtBrick's Term of Use Policy, available at <https://www.rtbrick.com/privacy>. Use of marks belonging to other parties is for informational purposes only

Table of Contents

1	Introduction to RtBrick BGP	1
1.1	BGP Overview	1
1.2	Supported BGP Standards	1
1.3	Supported BGP Features	2
2	RBFS BGP Instance Hierarchy	3
2.1	Address Family	3
2.2	Redistribute	4
2.3	Modes and Commands	4
2.4	Hierarchy Levels and Configuration Statements	5
3	BGP Global Configuration Commands	6
3.1	Instance	6
3.2	Route Distinguisher	6
3.3	Route Target	6
4	BGP Instance Configuration Commands	8
4.1	bgp	8
4.2	address-family (and BGP Options)	8
4.3	download-count	10
4.4	download-policy	11
4.5	multipath	11
4.6	redistribute	12
4.6.1	IS-IS Redistribute Options	12
4.6.2	OSPF Redistribute Options	13
4.7	retain-route-target	13
4.8	srgb	14
4.9	bestroute-selection	14
4.10	cluster-id	15
4.11	domain-name	15
4.12	enforce-first-as	16

4.13	host-name.....	16
4.14	local-as	17
4.15	local-preference	17
4.16	out-metric	17
4.17	protocol-preference	18
4.18	router-id.....	18
4.19	timer	18
4.20	type-of-service	19
4.21	peer-group.....	19
4.22	address-family (peer-group)	20
4.23	peer.....	22
5	BGP Show Commands for APPD	24
5.1	show bgp rib	24
5.2	show bgp fib	25
5.3	show bgp ribout	26
6	BGP Show Commands for IOD	27
6.1	show bgp peer.....	27
6.2	show bgp peer detail	28
6.3	show bgp peer rib-in	31
7	Routing Policies and Examples	32
7.1	The Role of Routing Policy.....	32
7.2	RtBrick Routing Policy Implementation	32
7.3	Peer Group and Instance Details	33
7.4	Routing Policy Specifics	33
7.5	Routing Policy Examples.....	34
7.5.1	Routing Policy Example #1	34
7.5.2	Routing Policy Example #2	34
7.5.3	Routing Policy Example #3	36
7.5.4	Routing Policy Example #4	36
7.6	Routing Policy Summary.....	37

1 Introduction to RtBrick BGP

The RtBrick Full Stack (RBFS) is a distributed, web-scale network operating system based on micro-service architecture and packaged in a Linux container currently based on Ubuntu 18.04. The RBFS decreases the complexity of a network by providing users with capabilities such as composability, programmability, and elasticity in a visionary way.

The RBFS operating system operates on the leaf and spine framework. This framework provides the network operators with methods to configure and manage a network brick by brick, and provides full control of the network so the focus can be on the deployment of the new services and not the constant juggling of hardware.

1.1 BGP Overview

BGP is a standard exterior gateway protocol (EGP) supported by RtBrick. BGP is considered a “Path Vector” routing protocol and maintains a separate routing table based on shortest Autonomous system (AS) path and various other route attributes.


















1.2 Supported BGP Standards

RFC Number	Description
RFC 2545	Use of BGP-4 Multiprotocol Extensions for IPv6 Inter-Domain Routing
RFC 2918	Route Refresh Capability for BGP-4
RFC 4271	A Border Gateway Protocol 4 (BGP-4)
RFC 4364	BGP/MPLS IP Virtual Private Networks (VPNs)
RFC 4456	BGP Route Reflection: An Alternative to Full Mesh Internal BGP (IBGP)
RFC 4486	Subcodes for BGP Cease Notification Message
RFC 4760	Multiprotocol Extensions for BGP-4
RFC 5492	Capabilities Advertisement with BGP-4
RFC 6793	BGP Support for Four-Octet Autonomous System (AS) Number Space
RFC 6608	Subcodes for BGP Finite State Machine Error
RFC 6774	Distribution of Diverse BGP Paths [Partial Support]

1.3 Supported BGP Features

The RBFS supports the following BGP functions:

Basic BGP Protocol

-  Multiprotocol extension for BGP
-  Multipath for iBGP and eBGP
-  Four-byte AS numbers
-  Nexthop Self or nexthop unchanged
-  Fast external-failover
-  Route reflection
-  MD5 Authentication
-  Route refresh
-  Advanced route refresh
-  Route redistribution
-  Multihop EBGP
-  Route selection flexibility (always compare MED, ignore AS Path, and so on)
-  Add path
-  Host name/Domain name
-  Dynamic peers
-  Community, Extended Community, and Large Community support
-  6PE Support

The statements and commands required to configure and verify the functioning of BGP features are described in this guide.

2 RBFS BGP Instance Hierarchy

The RBFS uses a distinct hierarchy for the BGP statements and commands. When installed, the RBFS runs a default instance named "default." In most applications, more instances are needed, and many configuration statements apply to various levels of the statement hierarchy.

You can apply BGP configuration statements and commands at the instance, address family, or protocol BGP hierarchy level.

This BGP hierarchy can be represented as:

```
instance <instance>
- address-family
- protocol bgp
    - address-family
    - redistribute
    - peer-group < peer-group-name >
    - address-family
    - peer
```

The configuration of instances, the BGP protocol, peer groups, and peers are straightforward. A few words of explanation are needed for address families and redistribute.

2.1 Address Family

The address family configured at the instance level determines the types of address families that the virtual routing and forwarding (VRF) instance supports. At this level, you also configure the route target (RT), and the import and export policies for the instance. The forwarding daemon (FWDD) uses these configuration statements to create the VRF tables for each address family. Note that the route distinguisher (RD) is configured under the `instance` hierarchy level.

At the BGP protocol level, the address family configuration is used to determine which address families BGP supports. Multiple commands exist under this hierarchy level.

At the BGP peer group level, the address family configuration is used to determine which address families are needed to negotiate usage with the other BGP peers in the group.

There is no configuration under the peer hierarchy level. This statement establishes the peer, and all peers must be associated with a peer group. All of the parameters that apply to a peer are determined at the peer-group hierarchy level.

For example, this sequence configures IPv4 unicast support for the `vrf1` instance:

```
root@leaf1:confd> edit instance vrf1
instance vrf1 ]
root@leaf1:confd> edit address-family ipv4 unicast
```

```
[ instance vrf1 address-family ipv4 unicast ]
```

2.2 Redistribute

BGP determines support for route redistribution based on address family. The selected address family can redistribute local interface routes, static routes, routes learned by IS-IS, and so on. Complete redistribute statement information and examples are included in other sections.

The example below shows route redistribution possibilities:

```
root@leaf1:confd> edit instance vrf1
instance vrf1 ]
root@leaf1:confd> edit address-family ipv4 unicast
[ instance vrf1 address-family ipv4 unicast ]
root@leaf1:confd> set redistribute
direct          Redistribute connected routes
isis            Redistribute isis routes
local           Redistribute local routes
ospf            Redistribute ospf routes
pim             Redistribute pim routes
ppp             Redistribute ppp routes
static          Redistribute static routes
[<Enter>]       Execute the command
```

Some configuration statements start with `edit` and other start with `set`. These are called *modes* and *commands* in the RBFS.

2.3 Modes and Commands

The RBFS uses both modes (using “edit”) and commands (using “set”) to configure statements and determine parameters. All edits and sets are done through the configuration daemon CONFD. Their end result is similar, but the terms apply to slightly different configuration methods:

- **Mode**—An “edit” is used to provide a hierarchical configuration option for the module instance.
- **Commands**—A “set” is used to set the statement parameters and attributes for the module instance.

In a typical configuration session, you will use both methods:

```
root@leaf1:confd> edit instance
[ instance ]
root@leaf1:confd> edit protocol bgp
[ instance protocol bgp ]
root@leaf1:confd> set peer ipv4 peer-address 10.1.1.1 update-source
10.2.2.2 peer-group spine
```

You leave the edit mode with the `exit` command.

2.4 Hierarchy Levels and Configuration Statements

In many cases, you can configure the same feature from different levels of the configuration hierarchy. For example, you can set a BGP redistribution for direct addresses at the global, protocol, or address family level of the hierarchy.

The following three statements set redistribution of direct addresses for BGP.

At the global level:

```
[edit]
root@spine1:confd> set bgp address-family ipv4-unicast redistribute
direct
```

At the BGP protocol level:

```
[edit bgp]
root@spine1:confd> set address-family ipv4-unicast redistribute direct
```

At the address family level:

```
[edit bgp address-family ipv4-unicast]
root@spine1:confd> set redistribute direct
```

When committed without errors, all three result in the same configuration, displayed with `show running-configuration`:

```
[edit]
# show running-configuration
{
  "running-configuration": {
    "bgp": {
      "address-family:ipv4-unicast" {
        "redistribute": {
          "source:direct"
        }
      }
    }
  }
}
```

3 BGP Global Configuration Commands

When it comes to BGP configuration, global BGP commands configure the routing instance, and instance BGP commands configure how BGP behaves in the instance.

This describes the BGP global commands. These commands establish the instance, the route distinguisher (RD), and the route target (RT) for BGP to use.

3.1 Instance

Configure and log into the RBFS instance

edit instance <instance-name>

<instance-name>	Name of the routing instance
<Enter>	Execute command

Example: Configure vrf1 on spine1

```
root@spine1:confd> edit instance vrf1
```

3.2 Route Distinguisher

Configure the route distinguisher (RD) to define unique routes within an IPv4 network. Provider Edge (PE) routers use route distinguishers to identify which virtual private network (VPN) a packet belongs to in a PE network.

set instance <instance-name> rd <as-number | ipv4-address>:<rd-value>

<instance-name>	Name of the routing instance
<as-number>	The AS number to use for this RD (the RD is in two-byte AS format)
<ipv4-address>	The IPv4 address to use for the RD (in dotted decimal format)
<rd-value>	The unique RT identifier. Range: 0 through 429967295
<Enter>	Execute command

Example: Configure RD 192.168.1.1:65001 to vrf1

```
root@spine1:confd> set instance vrf1 rd 192.168.1.1:65001
```

3.3 Route Target

Configure the route target (RT) to use to transfer routes between VRFs and VPNs. The RT identifies a subset of routes within the BGP VPNv3 unicast table that should be used in a VRF for a particular customer. You configure an RT for importing or exporting routes, or both.

set rt (import | export } both) target:<rt-value>

import	Use this RT for imported routes
export	Use this RT for exported routes
both	Use this RT for both imported and exported routes
target:<rt-value>	The value to use for the RT, in the same format as the RD
<Enter>	Execute command

Example: Configure an RT for import and export routes

```
[ edit instance vrf1 address-family ipv4 unicast]
root@spine1:confd> set rt import target:192.168.1.1:111
[ edit instance vrf1 address-family ipv4 unicast]
root@spine1:confd> set rt export target:192.168.1.1:222
```

4 BGP Instance Configuration Commands

This section describes the commands used to configure BGP in an instance.

4.1 **bgp**

You must enable BGP inside an instance before you configure the BGP-specific instance options. When this enable command is successfully executed, the mode is changed from `edit` to `set`.

edit protocol bgp

(See below for BGP options)	
<Enter>	Execute command

Example: Configure BGP for the vrf1 instance

```
[ edit instance vrf1 ]
root@spine1:confd> edit protocol bgp
```

4.2 **address-family (and BGP Options)**

You must configure the BGP address families if you are using route redistribution, load balancing, or other advanced features. By default, BGP neighbor sessions support IPv4 unicast and multicast address families.

set protocol bgp address-family <address-family-afi> <address-family-safi>

<address-family-afi>	Enables the address family identifier (AFI) IPv4 or IPv6
<address-family-safi>	Enables the subsequent address family identifier (SAFI) options: unicast, labeled-unicast, or vpn-unicast
<Enter>	Execute command

Example: Enable the IPv4 unicast address family for BGP on the vrf1 instance

```
[ edit instance vrf1 ]
root@spine1:confd> set protocol bgp address-family ipv4 unicast
```

In addition to the basic family support, you can also configure a number of options. These are mostly self-explanatory and only basic information is provided.

Supported options, detailed in sections below, are:

default-information	Configure default route information to distribute
download-count	Forward packets over multiple paths, but set maximum prefixes to use
download-policy	Establish a policy for downloading routes
multipath	Enable load sharing among multiple BGP paths
redistribute	Enter redistribution configuration mode
retain-route-target	Retain VPN routes for all route targets
srgb	Establish Segment Routing Global Block for segment routing

NOTE: You should always configure BGP to distribute a default route.

Example: Configure BGP to distributes a default IPv4 unicast address in the vrf1 instance

[edit instance vrf1]

```
root@spine1:confd> set protocol bgp address-family ipv4 unicast default-information originate
```

4.3 download-count

This statement is for temporary use until full routing policy support is configured. Use this statement to configure the number of prefixes that the BGP unicast session can download to the forwarding daemon (FWDD). BGP downloads the attribute information (community, extended community, and as-path) for prefixes to the Routing Information Base (RIB) and Forwarding Information Base (FIB) for advertising.

You can assign counters based on BGP prefixes and attributes on a per-input interface basis.

set protocol bgp address-family <address-family-afi> <address-family-safi> unicast download-count <count-number>

<address-family-afi>	Enables the IPv4 or IPv6 address families
<address-family-safi>	Enables the subsequent address family identifier (SAFI) options: unicast, labeled-unicast, or vpn-unicast
unicast	BGP distributes unicast routes
download-count	Count prefixes
<count-number>	Number of prefixes to forward over multiple paths
<Enter>	Execute command

Example: Configure BGP for a maximum of 40 prefixes

```
[ edit instance vrf1 ]
root@spine1:confd> set protocol bgp address-family ipv4 unicast
download-count 40
```


4.4 download-policy

Apply the BGP routing policy for downloaded routes. The policy is defined elsewhere.

set protocol bgp address-family <address-family-afi> <address-family-safi> download-policy <policy-name>

<address-family-afi>	Enables the IPv4 or IPv6 address families
<address-family-safi>	Enables the subsequent address family identifier (SAFI) options: unicast, labeled-unicast, or vpn-unicast
<policy-name>	Name of the policy to apply
<Enter>	Execute command

NOTE: The policy is NOT configured here. Routing policy configuration is covered in another chapter.

Example: Configure download policy

```
[edit]
root@spine1:confd> set bgp address-family ipv4 unicast download-policy
fibpolicy
```

4.5 multipath

Allow load sharing among the configured number of multiple Exterior BGP (EBGP) and Internal BGP (IBGP) paths.

set protocol bgp address-family <address-family-afi> <address-family-safi> multipath <number-of-paths>

<address-family-afi>	Enables the IPv4 or IPv6 address families
<address-family-safi>	Enables the subsequent address family identifier (SAFI) options: unicast, labeled-unicast, or vpn-unicast
<number-of-paths>	Number of paths that can be installed in the forwarding table. This is an integer in the range 0 to 255.
<Enter>	Execute command

Example: Configure multipath for a value of 10 forwarding table entries

```
[edit]
root@spine1:confd> set bgp address-family ipv4 unicast multipath 10
```

4.6 redistribute

Enable the redistribution feature to dynamically inject routes from a certain protocol into the IP routing table.

set protocol bgp address-family <address-family-afi> <address-family-safi> redistribute <protocol>

<address-family-afi>	Enables the IPv4 or IPv6 address families
<address-family-safi>	Enables the subsequent address family identifier (SAFI) options: unicast, labeled-unicast, or vpn-unicast
<protocol>	Redistribute routes from this protocol. Protocols can be direct or local interfaces, IS-IS, OSPF, PIM, PPP, or static routes.
<Enter>	Execute command

NOTE: IS-IS and OSPF have additional options detailed below.

Example: Configure redistribution on direct routes into BGP.

```
[edit]
root@spine1:confd> set protocol bgp address-family ipv4 unicast
redistribute direct
```

4.6.1 IS-IS Redistribute Options

If you configure the redistribution of IS-IS routes, you have the option of testing for a `match` on either `level-1` or `level-2` IS-IS routers. The syntax is as follows:

set protocol bgp address-family <address-family-afi> <address-family-safi> redistribute is-is match <isis-level>

<address-family-afi>	Enables the IPv4 or IPv6 address families
<address-family-safi>	Enables the subsequent address family identifier (SAFI) options: unicast, labeled-unicast, or vpn-unicast
<isis-level>	Match routes from either IS-IS <code>level-1</code> or <code>level-2</code> routers
<Enter>	Execute command

Example: Configure redistribution for IS-IS Level 1 routes.

```
[edit]
root@spine1:confd> set protocol bgp address-family ipv4 unicast
redistribute isis match level-1
```

4.6.2 OSPF Redistribute Options

If you configure the redistribution of OSPF routes, you have the option of testing for a `match` on several types of OSPF routers. The syntax is as follows:

set protocol bgp address-family <address-family-afi> <address-family-safi> redistribute is-is match <ospf-type>

<address-family-afi>	Enables the IPv4 or IPv6 address families
<address-family-safi>	Enables the subsequent address family identifier (SAFI) options: unicast, labeled-unicast, or vpn-unicast
<ospf-type>	Match routes from this route type and install into the routing table: external-type-1 external-type-2 inter-area intra-area nssa-external-type-1 nssa-external-type-2
<Enter>	Execute command

Example: Configure redistribution for OSPF inter-area routes.

```
[edit]
```

```
root@spine1:confd> set protocol bgp address-family ipv4 unicast
redistribute ospf match inter-area
```

4.7 retain-route-target

By default, the forwarding table retains the VPN routes for all route targets when the routing protocol process shuts down. If you disable this retention of route targets, the VPN routes are removed from the forwarding table when the routing protocol shuts down.

set protocol bgp address-family <address-family-afi> <address-family-safi> retain-route-target disable

<address-family-afi>	Enables the IPv4 or IPv6 address families
<address-family-safi>	Enables the subsequent address family identifier (SAFI) for vpn-unicast (the only family supported)
<Enter>	Execute command

Example: Configure removal of VPN routes when routing process shuts down.

```
[edit]
root@spine1:confd> set bgp address-family ipv4 vpn-unicast retain-
route-target disable
```

4.8 srgb

The Segment Routing Global Block (SRGB) is the range of label values reserved for segment routing (SR). These values are assigned as segment identifiers (SIDs) to SR-enabled network nodes and have global significance throughout the routing domain.

NOTE: SRGB is supported only for labeled unicast.

You must specify the base starting value for the labels (a number greater than 15), a range for the labels, and an index within the label range.

set protocol bgp address-family <address-family-afi> labeled-unicast srgb base <base-value> range <range-value> index <index-value>

<address-family-afi>	Enables the IPv4 or IPv6 address families
base <base-value>	Starting base value, greater than 15, of the SRGB values
range <range-value>	Range of label values
index <index-value>	Index within the label range
<Enter>	Execute command

Example: Configure the SRGB label base to 2000, range to 100, and index to 2 (which selects label 2001 from the zero-based label range).

```
[edit]
root@spine1:confd> set protocol bgp address-family ipv4 labeled-unicast
base 2000 range 100 index 2
```

4.9 bestroute-selection

BGP route selection involves more than just adding routes to the BGP routing table and forwarding table. By default, BGP implementations provide their own criteria for selecting the “best route” based on the values of BGP attributes associated with the route. If the route selection process is not producing optimal routes within a routing domain, you can steer the BGP route selection process in a number of ways with the best route selection option.

set protocol bgp bestroute-selection <best-route-option>

<best-route-option>	<div> <div>Instead of default, consider best route selection using these options:</div> <table> <tr> <td>always-compare-med</td><td>Always compare MED</td></tr> <tr> <td>ignore-as-path</td><td>Ignore AS path</td></tr> <tr> <td>ignore-cluster-length</td><td>Ignore cluster length</td></tr> <tr> <td>ignore-local-preference</td><td>Ignore local preference</td></tr> <tr> <td>ignore-med</td><td>Ignore MED</td></tr> <tr> <td>ignore-origin</td><td>Ignore origin</td></tr> <tr> <td>ignore-peer-ip</td><td>Ignore peer IP address</td></tr> <tr> <td>ignore-route-source</td><td>Ignore route source</td></tr> <tr> <td>ignore-route-type</td><td>Ignore the type of route</td></tr> <tr> <td>ignore-router-id</td><td>Ignore router ID</td></tr> </table> </div>	always-compare-med	Always compare MED	ignore-as-path	Ignore AS path	ignore-cluster-length	Ignore cluster length	ignore-local-preference	Ignore local preference	ignore-med	Ignore MED	ignore-origin	Ignore origin	ignore-peer-ip	Ignore peer IP address	ignore-route-source	Ignore route source	ignore-route-type	Ignore the type of route	ignore-router-id	Ignore router ID
always-compare-med	Always compare MED																				
ignore-as-path	Ignore AS path																				
ignore-cluster-length	Ignore cluster length																				
ignore-local-preference	Ignore local preference																				
ignore-med	Ignore MED																				
ignore-origin	Ignore origin																				
ignore-peer-ip	Ignore peer IP address																				
ignore-route-source	Ignore route source																				
ignore-route-type	Ignore the type of route																				
ignore-router-id	Ignore router ID																				
<Enter>	Execute command																				

Example: Configure BGP to ignore AS path considerations during route selection.

```
[edit]
```

```
root@spine1:confd> set protocol bgp bestroute-selection ignore-as-path
```

NOTE: You cannot configure both the `always-compare-med` and `ignore-med` options. By default, the system always compares MEDs.

4.10 cluster-id

Configure cluster ID to associate routers in a group within a BGP routing instance. Routers belong to the same cluster if they have the same cluster ID. The cluster ID is formatted as an IPv4 address.

set protocol bgp cluster-id <cluster-identifier>

<cluster-identifier>	An IPv4 address
<Enter>	Execute command

Example: Configure cluster-id 192.168.1.1 for this BGP instance.

```
[edit]
```

```
root@spine1:confd> set protocol bgp cluster-id 192.168.1.1
```

4.11 domain-name

Configure the domain name for this BGP routing instance.

set protocol bgp domain-name <domain-name>

<domain-name>	The name of the BGP routing domain, to a maximum of 64 characters
<Enter>	Execute command

Example: Configure domain name *spine* for this BGP instance.

```
[edit]
root@spine1:confd> set protocol bgp domain-name spine
```

4.12 enforce-first-as

By default, the BGP routing process enforces the First AS feature. This feature discards updates received from an eBGP peer if the peer does not list its own AS number as the first segment in the AS_PATH BGP attribute.

To disable the First AS feature and accept updates without the peer's source AS matching the first AS in the AS_PATH attribute, configure the feature to disable the enforce First AS feature.

set protocol bgp enforce-first-as disable

disable	Disable the default First AS feature BGP behavior
<Enter>	Execute command

Example: Configure this BGP instance to disable the First AS feature.

```
[edit]
root@spine1:confd> set protocol bgp enforce-first-as disable
```

4.13 host-name

Configure a host name for BGP to use.

set protocol bgp host-name <host-name>

<host-name>	The name of the BGP host, to a maximum of 64 characters
<Enter>	Execute command

Example: Configure this BGP instance to use host name *spine1*.

```
[edit]
root@spine1:confd> set protocol bgp host-name spine1
```

4.14 local-as

Configure the local AS number in four-byte format for BGP to use.

set protocol bgp local-as <as4-number>

<as4-number>	The AS number in four-byte format. The numbers allowed are from 1 to 4294967285
<Enter>	Execute command

Example: Configure this BGP instance to use local AS Number 65001.

[edit]

```
root@spine1:confd> set protocol bgp local-as 65001
```

4.15 local-preference

Configure the value to use as BGP local preference. You can use the local AS preference number to choose the exit path for an AS.

set protocol bgp local-preference <preference-number>

<preference-number>	The local preference number for the BGP protocol. The numbers allowed are from 0 to 4294967285
<Enter>	Execute command

Example: Configure this BGP instance to use local preference 150.

[edit]

```
root@spine1:confd> set protocol bgp local-preference 150
```

4.16 out-metric

Configure the value to use as BGP Multi-Exit Discriminator (MED) value. When an AS has multiple links to another AS, the MED value is used to determine the exit to use to reach the other AS.

set protocol bgp out-metric <med-value>

<med-value>	The MED value for the BGP protocol. The numbers allowed are from 0 to 4294967285
<Enter>	Execute command

Example: Configure this BGP instance to use MED 10.

[edit]

```
root@spine1:confd> set protocol bgp out-metric 10
```

4.17 protocol-preference

Assign a protocol preference distance value to routes learned by eBGP, iBGP, or both.

set protocol bgp protocol-preference (internal | external) <preference-value>

internal	Use this distance value preference for routes learned by iBGP
external	Use this distance value preference for routes learned by eBGP
<preference-value>	Use this distance value as BGP route protocol preference
<Enter>	Execute command

Example: Configure this BGP instance to use a value of 50 for eBGP routes and a value of 100 for iBGP routes.

```
[edit]
```

```
root@spine1:confd> set protocol bgp protocol-preference external 50
```

```
root@spine1:confd> set protocol bgp protocol-preference internal 100
```

4.18 router-id

Configure the value used as the router ID.

set protocol bgp router-id <router-id>

<router-id>	Router identifier in IPv4 format
<Enter>	Execute command

Example: Configure the router identifier to be 192.168.1.1.

```
[edit]
```

```
root@spine1:confd> set protocol bgp router-id 192.168.1.1
```

4.19 timer

Configure the timer value used to keep the router active.

set protocol bgp timer hold-time <seconds> keep-alive <seconds>

hold-time	The time in seconds that the router keeps the connection as active when negotiating a connection with a peer. The valid range is 5 to 65535.
-----------	--

keep-alive	The frequency at which the router sends a keep-alive to retry a connection to keep the router active. The valid range is 5 to 65535
<seconds>	Time in seconds (ranges defined above)
<Enter>	Execute command

Example: Configure the router connect retry to 15 seconds, and the hold-time to 60 seconds.

```
[edit]
root@spine1:confd> set protocol bgp hold-time 60 keep-alive 30
```

4.20 type-of-service

Configure the value for the type-of-service (DSCP) bits.

set protocol bgp type-of-service <tos-value>

<tos-value>	Value of the type-of-service bits
<Enter>	Execute command

Example: Configure the type-of-service value used to 32.

```
[edit]
root@spine1:confd> set protocol bgp type-of-service 32
```

4.21 peer-group

Configure a name for a peer group of BGP routers. Neighbor peers with the same update policies can be grouped as peers (equals) to simplify the initial configuration and updates. Peers share the same policies such as route maps, distribution lists, filter lists, update source, and so on, so peer groups only need one configuration statement for these values.

set protocol bgp peer-group <peer-group-name> (address-family <address-family-afi> | any-as | ebgp-multihop <hop-count> | link-local-nexthop-only | local-as <as-number> | remote-as <as-number>)

<peer-group-name>	Name of the peer group
address-family <address-family-afi> <address-family-safi>	Configure to address family AFI and SAFI for the peer group (for supported AFI and SAFI values, see the address-family BGP configuration statement)
any-as	Enable dynamic AS negotiation for this peer group

ebgp-multihop <hop-count>	By default, the maximum number of hops between eBGP peers is 1 (direct connection). This hop count overrides the default behavior allow connectivity between eBGP peers not directly connected.
link-local-nexthop-only	Enable link-local nexthop only feature
local-as	Configure local AS number
remote-as	Configure remote AS number
add-path <count (both receive-only send-only)>	Configure the number of additional paths to a destination to advertise (send), or to receive, or both.
<Enter>	Execute command

Example: Configure the BGP peer as spine and enable dynamic negotiation of the AS number, sets the eBGP multihop count to 3, and set the number of additional paths sent to the peer group to 20.

[edit]

```
root@spine1:confd> set protocol bgp peer-group spine any-as
root@spine1:confd> set protocol bgp peer-group spine ebgp-multihop 3
root@spine1:confd> set protocol bgp peer-group spine add-path 20 send-only
```

NOTE: The full set of address family names is listed under the address-family configuration statement.

4.22 address-family (peer-group)

You must configure the BGP address families for a peer-group if you are using route redistribution, load balancing, or other advanced features.

Note: The advanced features are NOT configured here, but under the BGP instance.

By default, BGP neighbor sessions support IPv4 unicast and multicast address families.

set protocol bgp peer-group <peer-group-name> address-family <address-family-afi> <address-family-safi> [options]

<peer-group-name>	Name of the peer group supporting these families
<address-family-afi>	Enables the address family identifier (AFI) IPv4 or IPv6
<address-family-safi>	Enables the subsequent address family identifier (SAFI) options: unicast, labeled-unicast, or vpn-unicast
<Enter>	Execute command

Additional options:

add-path	Negotiate additional path capabilities with these peers, so that more than one path can be active to the peers in the group
default-information	Enable default route information to be sent to peers in group
extended-nexthop	Enable extended nexthop capability so that peers in the group do not have to be directly connected
(nexthop-self nexthop-unchanged)	Propagate nexthop self or nexthop unchanged for advertised paths
policy	Apply a routing policy to the peer group
remove-private-as	Remove private AS numbers from routes advertised to group peers
route-reflect-client	Configure this peer as a route reflector client
update-nexthop	Update the nexthop for routes advertised to this peer group
<Enter>	Execute command

Example: Enable the IPv4 unicast address family for BGP on the vrf1 instance and make the peer a route-reflector client

```
[ edit instance vrf1 ]  
root@spine1:confd> set protocol bgp address-family ipv4 unicast route-  
reflect-client
```

4.23 peer

Configure parameters for a, IPv4 or IPv6 BGP peer. A BGP peer is a BGP router that has an active TCP connection to another BGP router. To establish point-to-point connections between peer autonomous systems (ASs) for eBGP or within the AS for iBGP, you configure a BGP session on an interface. When you configure a BGP peer, the configuration adds the IP address of the peer in the specified autonomous system to the multiprotocol BGP (mBGP) neighbor table maintained in the local router.

set instance protocol bgp peer (ipv4 | ipv6) peer-address <ip-address> (update-source >ip-address> | peer-group <peer-group-name> | interface <interface-name>)

(ipv4 ipv6)	Configure either an IPv4 or IPv6 BGP peer
peer-address <ip-address>	Configure the peer IPv4 or IPv6 address. You cannot use this option with the interface option below.
update-source <ip-address>	Configure an IP address to use as the source for packets sent to the peer address
peer-group <peer-group-name>	Assign this peer to this peer group
interface <interface-name>	Enable BGP on this interface. You cannot use this option with the peer-address option above. Peers configured with a BGP interface are known as <i>dynamic peers</i> .
<Enter>	Execute command

NOTE: The `peer-address` and `interface` options are mutually exclusive. In other words, you cannot locate a peer by both interface and address; only one or the other.

There are three main peer types to configure:

- IPv4 peers
- IPv6 peers
- IPv6 link local peers (dynamic peers)

Each type has a general peer configuration format associated with it:

- (IPv4 peer) `set instance protocol bgp peer ipv4 peer-address <ipv4-address> update-source <ipv4-address> peer-group <group-name>`
- (IPv6 peer) `set instance protocol bgp peer ipv6 peer-address <ipv6-address> update-source <ipv6-address> peer-group <group-name>`
- (IPv6 link-local dynamic peer) `set instance protocol bgp peer interface <interface-name> peer-group <group-name>`

Note that IPv6 link-local peers configure the interface.

The following examples configure a peer of each type.

Example 1: Configure an IPv4 BGP peer and assign the peer to the spine peer group.

```
[edit]  
root@spine1:confd> set instance protocol bgp peer ipv4 peer-address  
192.168.1.1 update-source 192.168.1.2 peer-group spine
```

Example 2: Configure an IPv6 peer and assign the peer to the spine peer group.

```
[edit]  
root@spine1:confd> set instance protocol bgp peer ipv6 peer-address  
1::1 update-source 1::2 peer-group spine
```

Example 3: Configure an IPv6 dynamic peer on interface ifl-0/0/1/1/1 and assign the peer to peer group spine.

```
[edit]  
root@spine1:confd> set instance protocol bgp peer interface ifl-  
0/0/1/1/1 peer-group spine
```

5 BGP Show Commands for APPD

Configuration statements can be shown to be working properly using `show` commands.

This section details the fields displayed by BGP show commands handled by the APPD module. BGP is implemented in two modules: APPD and IOD.

5.1 show bgp rib

Display information about the BGP routing information base local (rib-local) table for an address family. The rib-local table is used to determine entries for the forwarding table.

show bgp rib <afi> <safi>

Command Options:

<afi>	AFI value: ipv4 or ipv6
<safi>	SAFI value: unicast, labeled-unicast, or vpn-unicast
<Enter>	Execute command

Output Fields:

Prefix	The Network Layer Routing Information (NLRI) in BGP RIB-in
Snd-Path-ID	Path ID to be sent for the NLRI
Rcv-Path-ID	Add Path-ID received for the NLRI
Peer	Peer IP address supplying this information
Next-Hop	Next-Hop address for the NLRI
Up Time	Time since this NLRI was installed in table

Example Output: show bgp rib ipv4 unicast.

```
ubuntu@oli_core:~$ rtb bgp.appd.1 show bgp rib ipv4 unicast
```

Prefix	Snd-Path-ID	Rcv-Path-ID	Peer	Next-Hop	Up Time
0.0.0.0/0	3909607965	0	11.11.0.2	11.11.0.2	0d:0h:0m:0s
0.0.0.0/0	4032670015	0	7.11.1.7	7.11.1.7	0d:0h:0m:0s
0.0.0.0/0	1488459730	3	0.0.0.0		0d:0h:0m:0s
192.1.0.0/16	4032670015	0	7.11.1.7	7.11.1.7	0d:0h:0m:0s
192.2.0.0/16	4032670015	0	7.11.1.7	7.11.1.7	0d:0h:0m:0s
192.3.0.0/16	4032670015	0	7.11.1.7	7.11.1.7	0d:0h:0m:0s
192.4.0.0/16	4032670015	0	7.11.1.7	7.11.1.7	0d:0h:0m:0s
192.5.0.0/16	4032670015	0	7.11.1.7	7.11.1.7	0d:0h:0m:0s
10.1.1.0/24	4032670015	0	7.11.1.7	7.11.1.7	0d:0h:0m:0s
10.2.1.0/24	4032670015	0	7.11.1.7	7.11.1.7	0d:0h:0m:0s
10.3.1.0/24	4032670015	0	7.11.1.7	7.11.1.7	0d:0h:0m:0s
10.4.1.0/24	4032670015	0	7.11.1.7	7.11.1.7	0d:0h:0m:0s
10.5.1.0/24	4032670015	0	7.11.1.7	7.11.1.7	0d:0h:0m:0s

5.2 show bgp fib

Display information summary about the BGP routes downloaded to the Forwarding Information Base (FIB) table.

show bgp rib <afi> <safi>

Command Options:

<afi>	AFI value: ipv4 or ipv6
<safi>	SAFI value: unicast, labeled-unicast, or vpn-unicast
<Enter>	Execute command

Output Fields:

Prefix	The Network Layer Routing Information (NLRI) in the FIB
Preference	Preference value for the NLRI
Next-Hop	Next-Hop address for the NLRI

Example Output: show bgp fib ipv4 unicast.

```
ubuntu@ol1_core:~$ rtb bgp.appd.1 show bgp fib ipv4 unicast
```

Prefix	Preference	Next-Hop
192.1.0.0/16	20	7.11.1.7
192.2.0.0/16	20	7.11.1.7
192.3.0.0/16	20	7.11.1.7
192.4.0.0/16	20	7.11.1.7
192.5.0.0/16	20	7.11.1.7
10.1.1.0/24	20	7.11.1.7
10.2.1.0/24	20	7.11.1.7
10.3.1.0/24	20	7.11.1.7
10.4.1.0/24	20	7.11.1.7
10.5.1.0/24	20	7.11.1.7

5.3 show bgp ribout

Display information summary about the BGP routes in the peer group table that will be sent to peers from the Routing Information Base output (RIB-out) table.

show bgp rib <peer-group-name> <afi> <safi>

Command Options:

<peer-group-name>	Name of the peer group
<afi>	AFI value: ipv4 or ipv6
<safi>	SAFI value: unicast, labeled-unicast, or vpn-unicast
<Enter>	Execute command

Output Fields:

Prefix	The Network Layer Routing Information (NLRI) to be sent to peer
Rcv-Path-ID	Add Path-ID received for the NLRI
MED	MED for this NLRI
LocalPref	Local Preference value for the NLRI
Origin	Origin value for the NLRI
Next-Hop	Next-Hop address for the NLRI
AS-Path	AS-Path for the NLRI

Example Output: show bgp ribout vanilla_bgp ipv4 unicast.

```
ubuntu@oli_core:~$ rtb bgp.appd.1 show bgp ribout vanilla_bgp ipv4 unicast
```

Prefix	Rcv-Path-ID	MED	LocalPref	Origin	Next-Hop	AS-Path
0.0.0.0/0	3		100	Incomplete		
192.1.0.0/16	0		100	Incomplete		
192.2.0.0/16	0		100	Incomplete		
192.3.0.0/16	0		100	Incomplete		
192.4.0.0/16	0		100	Incomplete		
192.5.0.0/16	0		100	Incomplete		
10.1.1.0/24	0		100	Incomplete		
10.2.1.0/24	0		100	Incomplete		
10.3.1.0/24	0		100	Incomplete		
10.4.1.0/24	0		100	Incomplete		
10.5.1.0/24	0		100	Incomplete		

6 BGP Show Commands for IOD

Configuration statements can be shown to be working properly using `show` commands.

This section details the fields displayed by BGP show commands handled by the APPD module. BGP is implemented in two modules: APPD and IOD.

6.1 `show bgp peer`

Display information summary about the BGP peers.

`show bgp peer (instance <instance-name>)`

Command Options:

instance	Display information about peers in an instance
<instance-name>	Name of the instance: you must specify "default" for the default instance.
<Enter>	Execute command

Output Fields:

Peer	Peer at the remote end of the BGP session
Remote-AS	AS number of the peer
State	State of the peer (active, established, and so on)
Up/Down Time	Time since the peer has been in session or down
PfxRcvd	Prefixes received
PfxSent	Prefixes sent

Example Output: `show bgp peer`

```
ubuntu@leaf1:~$ rtb bgp.iod.1 show bgp peer
```

Peer	Remote-AS	State	Up/Down Time	PfxRcvd	PfxSent
spine-1	65001	Established	0d:0h:0m:58s	7	16
spine-2	65001	Established	0d:0h:0m:58s	5	16

```
ubuntu@leaf1:~$
```

6.2 show bgp peer detail

Display detailed information about the BGP peers.

show bgp peer <peer-ip> detail (instance <instance-name>)

Command Options:

<peer-ip>	Display information about the peer at this IP address
instance	Display information about peers in an instance
<instance-name>	Name of the instance: you must specify "default" for the default instance.
<Enter>	Execute command

Output Fields:

Peer	Peer host name
Peer Domain	Peer domain name
Peer IP	IP address of the peer
AS	AS number of the peer
Local	Source IP address for the peer session
Level	Level negotiated for the peer
AS	Local AS number
Any-AS	Dynamic AS negotiation configuration
Type	Type of BGP session
State	State of the peer session
Up Time	Time since the peer has been in session
Reason	Last transition reason
Sub-Code	Code for transition reason
Discovered on interface	Interface where peer is located
Last Transition	Time of last transistion
Flap count	Number of times the peer has flapped
Peer ID	Peer BGP identifier for the session

Output Fields	(continued)
Local ID	Local BGP identifier for the session
Instance	Instance of the session
Peer Group	Peer group for this peer

Detail Fields:

Timers	Time to expiration for timers listed
NLRI	Address families configured, received, or negotiated
Capabilites	Session capabilities configured, received, negotiated, or supported
End-Of-RIB	End-of-RIB sent and received time for each address family
Message Stats	BGP message sent and received count
Route Stats	NLRI sent and received count by address family

Example Output: show bgp peer detail

```

ubuntu@leaf1:~$ rtb bgp.iod.1 show bgp peer detail
Peer: spine-1, Peer Domain: spine-1, Peer IP: fe80::200:faff:febd:1030, AS 65001, Local: fe80::200:faff:febd:3010, AS 65003, Any-AS: false
Type: ebgp, State: Established, Up Time: 0d:0h:3m:21s, Reason: Cease, Sub-Code: Admin shutdown, Discovered on interface: twc-0/0/3/1/1
Last Transition: Tue Oct 08 14:22:42 Flap count : 0
Peer ID : 1.2.3.4 Local ID : 3.3.3.3
Instance : default Peer Group : spine
Timer values:
  Peer Keepalive : 30 Local Keepalive : 30
  Peer Holddown : 90 Local Holddown : 90
  Connect retry : 30
Timers:
  Keepalive timer: expires in 9s 892745us
  Holddown timer: expires in 1m 29s 245621us
NLRIIs:
  Sent : [ ipv6-unicast, ipv4-vpn-unicast, ipv6-labeled-unicast ]
  Received : [ ipv6-unicast, ipv6-labeled-unicast, ipv4-vpn-unicast ]
  Negotiated: [ ipv6-unicast, ipv6-labeled-unicast, ipv4-vpn-unicast ]
Capabilities:
  Addpath Sent : None
  Addpath Received : None
  Addpath Negotiated: None
  Extended Nexthop Sent : ipv4-vpn-unicast
  Extended Nexthop Received : ipv4-vpn-unicast
  Extended Nexthop Negotiated: ipv4-vpn-unicast
Capabilities
  Feature Sent Received Negotiated
  Route Refresh true true true
  4-Byte-AS true true true
  Graceful-Restart false false false
  Link Local Only false false false
End-Of-RIB:
  Address Family Sent Received
  IPv4 Unicast - -
  IPv4 Labeled Unicast - -
  IPv6 Unicast Tue Oct 08 14:22:44 Tue Oct 08 14:22:47
  IPv6 Labeled Unicast Tue Oct 08 14:22:44 Tue Oct 08 14:22:47
  IPv4 VPN-Unicast Tue Oct 08 14:22:44 Tue Oct 08 14:22:47
  IPv6 VPN-Unicast - -
  IPv4 VPN-Multicast - -
Message Stats:
Session Stats:
  Direction Open Update Keepalive Notify Route-Refresh
  Input 1 10 9 0 0
  Output 1 12 8 0 0
Total Stats:
  Direction Open Update Keepalive Notify Route-Refresh
  Input 1 10 9 0 0
  Output 1 12 8 3 0
Route Stats:
  Address Family Received Sent
  IPv4 Unicast 0 0
  IPv4 Labeled Unicast 0 0
  IPv6 Unicast 2 4
  IPv6 Labeled Unicast 2 4
  IPv4 VPN-Unicast 3 8
  IPv6 VPN-Unicast 0 3
  IPv4 VPN-Multicast 0 0

```

6.3 show bgp peer rib-in

Display information summary about the RIB-in table and a peer.

show bgp peer <peer-ip> rib-in <afi> <safi>

Command Options:

<peer-ip>	Display information about the peer at this IP address
<afi>	AFI value: ipv4 or ipv6
<safi>	SAFI value: unicast, labeled-unicast, or vpn-unicast
<Enter>	Execute command

Output Fields:

Prefix	The NLRI received from the peer
Rcv-Path-ID	Add Path-ID received for the NLRI
MED	MED received for this NLRI
LocalPref	Local Preference value received for the NLRI
Origin	Origin value received for the NLRI
Next-Hop	Next-Hop address received for the NLRI
AS-Path	AS-Path received for the NLRI

Example Output: show bgp peer 10.11.1.7 rib-in ipv4 unicast

```
ubuntu@oli_core:~$ rtb bgp.iod.1 show bgp peer 7.11.1.7 rib-in ipv4 unicast
```

Prefix	Rcv-Path-ID	MED	LocalPref	Origin	Next-Hop	AS-Path
0.0.0.0/0	0		100	Incomplete	7.11.1.7	
192.1.0.0/16	0		100	Incomplete	7.11.1.7	
192.2.0.0/16	0		100	Incomplete	7.11.1.7	
192.3.0.0/16	0		100	Incomplete	7.11.1.7	
192.4.0.0/16	0		100	Incomplete	7.11.1.7	
192.5.0.0/16	0		100	Incomplete	7.11.1.7	
10.1.1.0/24	0		100	Incomplete	7.11.1.7	
10.2.1.0/24	0		100	Incomplete	7.11.1.7	
10.3.1.0/24	0		100	Incomplete	7.11.1.7	
10.4.1.0/24	0		100	Incomplete	7.11.1.7	
10.5.1.0/24	0		100	Incomplete	7.11.1.7	

7 Routing Policies and Examples

This section covers the RtBrick way of configuring routing policies.

7.1 The Role of Routing Policy

The main job of network nodes like routers is to maintain information about “what networks are where?” and also about the links between the routers. The network nodes route traffic from one to another so packets are always making progress toward a destination (that is, there are no loops or black holes). Routers use routing protocols to share their knowledge of the network topology and interfaces so that all parts of the networks are reachable, even if some links and routers are not working properly.


However, it is also the job of the network nodes to filter or modify routing information as it is advertised from router to router in the network. This filtering is needed so that a simple edge router with a local and remote interface does not have to carry a core routing table with hundreds of thousands of entries. The modification of the free flow of routing information around the network is known as establishing a *routing policy* for the router. A routing policy determines which routing updates are accepted by the router, which properties should be changed, and which routes should be advertised to other routers.

Routing policies can be applied to any routing protocol, internal to an AS (an IGP) or not (an EGP). Nevertheless, the most critical routing protocol for applied routing policy is BGP. Because BGP comes in IGP (iBGP) and EGP (eBGP) forms, BGP information potentially goes *everywhere* in the network. To control this flow, which might overwhelm smaller devices, BGP routes are carefully filtered before and during acceptance, modification, and advertisement.

7.2 RtBrick Routing Policy Implementation

There is no standard for routing policy implementation. Therefore, each hardware or software vendor is free to create their own implementation of features and goals. RtBrick supports extensive options for policies. This document contains basic examples that can easily be extended to cover other situations.

RtBrick supports attaching a BGP routing policy at two levels:

-  Peer group `address-family level`
-  Instance `address-family level`

In each case, you can apply the policy as an `import` or `export` policy and filter. As expected, import filters determine which routing updates are accepted and export filters determine which routes are advertised to other peers.

7.3 Peer Group and Instance Details

An import policy, when applied to an address family at the peer group level, examines all *incoming* routes from all BGP peers in the peer group, but only for that address family.

An export policy, when applied to an address family at the peer group level, examines all *outgoing* routes to all BGP peers in the peer group, but only for that address family.

At the instance level, routing policies that are applied to an address family can work as import or export filters, but for the instance as a whole.

An import policy, when applied to an address family at the instance level, examines all *incoming* routes before accepting the information for the global VPN or other instance level, but only for that address family.

An export policy, when applied to an address family at the instance level, examines all *outgoing* routes before sending the information from the global VPN or other instance level, but only for that address family.

7.4 Routing Policy Specifics

There are three types of routing policy in this implementation:

- 📦 Route filter—a list of conditions to be matched to determine a permit or deny operation.

The match conditions can be configured to:

- Match all conditions listed
- Match any one condition in the list

- 📦 Route policy

- 📦 Route map

Do not be confused by the use of the terms “filter” and “policy” as specific types of routing policy. Instead of these terms referring to generic route filtering and application of a routing policy, RtBrick uses these terms as specific forms of an overall approach to BGP policy and routing.

7.5 Routing Policy Examples

Here are some examples of useful routing policies for BGP. In some cases, only two steps are required, for a route filter and import or export policy application: configure the filter policy and apply where needed. On others, three or even four steps are needed: configure the filter policy, then the route policy, then the route map, and finally, apply the routing policy where needed. This section gives examples of each

7.5.1 Routing Policy Example #1

Allow only the default route (0.0.0.0/0) from a peer (two steps).

1. Configure the filter policy named FILTER:

```
edit route filter FILTER operation permit match all
edit rules
set rule 1 match equal rd 192.168.1.1:65000
set rule 2 match equal prefix4 0.0.0.0/0
exit
exit
```

The `rd` keyword matches the BGP route descriptor configured. The keyword `prefix4` indicates the match is on the IPv4 address and mask that follows. Both rules must evaluate to true (`match all`) in order for the route to be accepted.

2. Apply the policy FILTER as an import policy to the correct peer group. In this case, the peer group is called `spine`:

```
edit instance protocol bgp
edit peer-group spine
edit address-family ipv4 vpn-unicast
set policy policy-name FILTER import
exit
exit
exit
```

7.5.2 Routing Policy Example #2

Modify the MED of two prefixes (0.0.0.0/0 and 10.1.1.1/32) before advertising the routes to a peer (four steps).

1. Configure the filter policy:


```
edit route filter FILTER operation permit match any
edit rules
set rule 1 match equal prefix4 0.0.0.0/0
set rule 1 match equal prefix4 10.1.1.1/32
exit
exit
```

The keyword `prefix4` indicates the match is on the IPv4 address and mask that follows. Either rule can evaluate to true (`match any`) in order for the route to be accepted.

2. Configure the route policy named `UPDATE_MED`:

```
edit route policy UPDATE_MED
edit rules
set rule 1 operation set med 50
exit
exit
```

This policy changes the MED for matching routes to a value of 50.

3. Configure the route map named `DEFAULT_PREFIX_UPDATE_MED`:

```
edit route map DEFAULT_PREFIX_UPDATE_MED
edit rules
set rule 1 if(FILTER) apply-continue UPDATE_MED
exit
exit
```

If the IP addresses listed in `FILTER` are matched, the MEDs are updated as specified in the `UPDATE_MED` policy.

4. Apply the route map to the correct peer group (spine) as an export policy:

```
edit instance protocol bgp
edit peer-group spine
edit address-family ipv4 unicast
set policy-name DEFAULT_PREFIX_UPDATE_MED export
exit
exit
exit
```

Note how the various pieces of route filter, policy, map, and import or export all act together to change the MEDs on the correct routes.

7.5.3 Routing Policy Example #3

Set the export route target (RT) for all routes exported from the instance `vrf1` (two steps).

1. Configure the route policy named `UPDATE_RT`:

```
edit route policy UPDATE_RT
edit rules
set rule 1 operation set extended-community
    target:192.168.1.1:111
exit
exit
```

This rule applies to all route because there is no condition to match..

2. Configure a route map named `PREFIX_FILTER`:

```
edit instance vrf1
edit address-family ipv4 unicast
set policy policy-name UPDATE_RT export
exit
exit
```

There is no need to specify the protocol when the policy applies to other than the main instance.

7.5.4 Routing Policy Example #4

Import only two prefixes into an instance (three steps, with two filters).

- 1, Configure the route filters named `FILTER1` and `FILTER2`:

```
edit route filter FILTER1 operation permit match all
edit rules
set rule 1 match equal rd 192.168.1.1:65000
set rule 2 match equal prefix4 0.0.0.0/0
exit
exit

edit route filter FILTER2 operation permit match all
edit rules
set rule 1 match equal rd 192.168.100.100:65000
set rule 2 match equal prefix4 10.1.1.1/32
exit
```

Note the presence of the common 192.168.1.1:65000 prefix in both rules. The rule is applied to all routes (there is no condition to match). Because there are two rules and the route must match all conditions in the rule set, the set matches two *different* VPN routes.

2. Configure the route map for the two rules:

```
edit route-map PREFIX_FILTER
edit rules
set rule1 if(FILTER1 | FILTER2) operation permit
exit
exit
```

This route map creates a rule that is true with an OR condition, meaning that the rule is true if *either* FILTER1 or FILTER2 has the routes listed in step (a). Then this route map permits the routes to pass.








3. Apply the PREFIX_FILTER route map to the instance `vrf1` for the address family `ipv4 unicast`:

```
edit instance vrf1
edit address-family ipv4 unicast
set policy policy-name UPDATE_RT export
exit
exit
```

The `vrf1` instance exports routes that match the listed route filters.

7.6 Routing Policy Summary

These examples show only the basics of the complete application of routing policy. Almost all BGP attributes can be used, along with many possible keywords. The current list is as follows:

-  `as-path`
-  `bgp-nh4`
-  `bgp-nh6`
-  `cluster-list`
-  `community`
-  `extended-community`
-  `label`
-  `large-community`
-  `local-preference`
-  `med`

- 📦 origin
- 📦 originator-id
- 📦 peer4
- 📦 peer6
- 📦 prefix4
- 📦 prefix6
- 📦 rd
- 📦 router-id4
- 📦 src-ipv4-address
- 📦 src-ipv6-address