



# BGP User Guide

Version 21.6.1, 25 June 2021

---

Registered Address	Support	Sales
26, Kingston Terrace, Princeton, New Jersey 08540, United States		
		+91 80 4850 5445
<a href="http://www.rtbrick.com">http://www.rtbrick.com</a>	<a href="mailto:support@rtbrick.com">support@rtbrick.com</a>	<a href="mailto:sales@rtbrick.com">sales@rtbrick.com</a>

©Copyright 2021 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. Overview .....	3
1.1. Supported BGP Standards .....	3
1.2. Supported BGP Features .....	3
1.2.1. MD5 Authentication .....	4
1.2.2. IPv6 Provider Edge (6PE) .....	4
1.2.3. Policies .....	4
1.2.3.1. The Role of a Routing Policy .....	4
1.2.3.2. Attachment Points .....	5
1.2.3.3. Policy Processing .....	5
2. BGP Configuration .....	6
2.1. Configuration Hierarchy .....	6
2.2. Configuration Syntax and Commands .....	6
2.2.1. Instance Configuration .....	6
2.2.1.1. Identifiers .....	6
2.2.1.2. Address Families .....	7
2.2.1.3. TCP Authentication Configuration .....	9
2.2.2. BGP Instance Configuration .....	10
2.2.3. BGP Address Family Configuration .....	12
2.2.4. Peer Group Configuration .....	16
2.2.4.1. Peer Groups .....	16
2.2.4.2. Address Families .....	17
2.2.5. Peer Configuration .....	19
2.3. Sample Configuration .....	20
3. BGP Operational Commands .....	27
3.1. BGP Show Commands .....	27
3.1.1. BGP Summary .....	27
3.1.2. BGP Peer .....	28
3.1.3. BGP Peer Group .....	31
3.1.4. BGP FIB .....	32
3.1.5. BGP RIB-in .....	34
3.1.6. BGP RIB-out .....	37
3.1.7. TCP Connections .....	39
3.1.8. TCP Statistics .....	41
3.2. BGP Clear Commands .....	42
3.2.1. BGP Peer .....	42

# 1. 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.

## 1.1. 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]

## 1.2. 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.

### **1.2.1. MD5 Authentication**

BGP supports the authentication mechanism using the Message Digest 5 (MD5) algorithm. When authentication is enabled, any Transmission Control Protocol (TCP) segment belonging to BGP exchanged between the peers is verified and accepted only if authentication is successful. For authentication to be successful, both the peers must be configured with the same password. If authentication fails, the BGP neighbor relationship is not be established.

### **1.2.2. IPv6 Provider Edge (6PE)**

The Provider Edge (6PE) solution enables IPv6 communication over MPLS IPv4 core network. IPv6 reachability information is associated with a label and transferred through MP-BGP(AFI: 2 SAFI:4). IPv4 mapped IPv6 address is used to encode the nexthop information. The edge nodes in MPLS IPv4 core have to support both IPv4 and IPv6. The IPv6 Labeled Unicast routes received from the 6PE peer is considered as IPv6 unicast routes and installed in IPv6 Unicast FIB. The received Label is attached to the IPv6 data traffic at the Ingress node and tunneled through a MPLS tunnel(SR) to the egress node, the label identifies the IPv6 traffic and the egress node would POP the label and forward the ipv6 traffic towards the destination.

### **1.2.3. Policies**

#### **1.2.3.1. The Role of a Routing Policy**

Routing Policies are the rules that allows you to control and modify the default behaviour of the routing protocols such as BGP and IS-IS. To use routing policies, you configure policies, and then apply policies to peer groups or instances.

### 1.2.3.2. Attachment Points

Policies are useful when they are applied to routes, for which they need to be made known to routing protocols. In BGP, for example, there are several situations where policies can be used, the most common of these is defining import and export policy. The policy attachment point is the point in which an association is formed between a specific protocol entity, in this case a BGP neighbor, and a specific named policy.

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.

### 1.2.3.3. Policy Processing

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 policies, but for the instance as a whole.

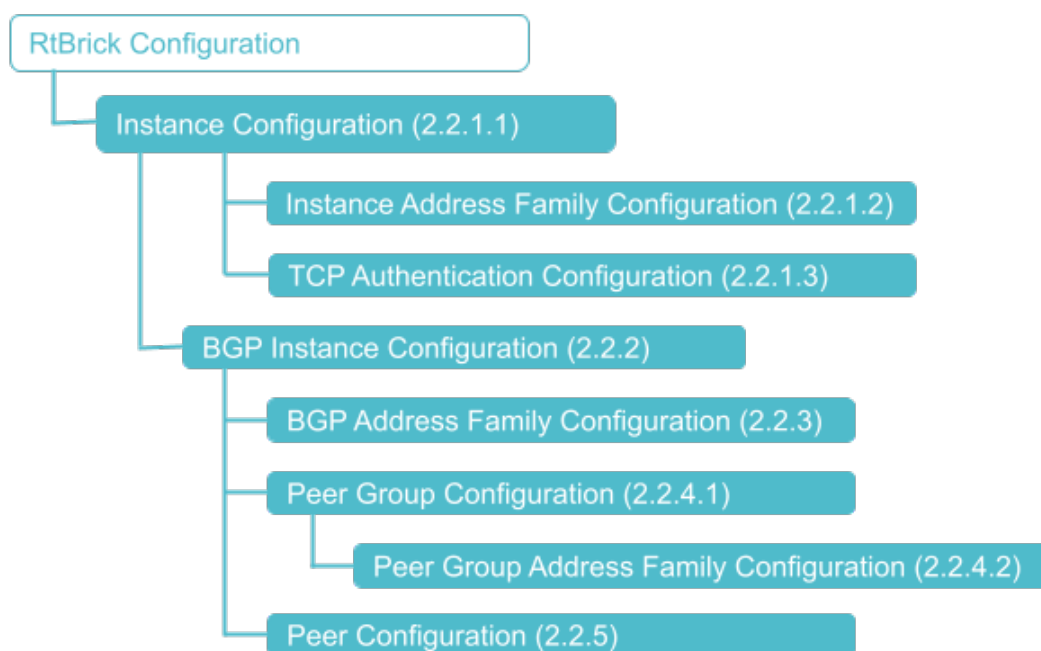
An import policy, when applied to an address family at the instance level, examines all incoming routes before accepting the information only from global or default tables to other instance or VRF table.

An export policy, when applied to an address family at the instance level, examines all outgoing routes before sending the information from the VRF to global, and then to the vpn table (default).

## 2. BGP Configuration

### 2.1. Configuration Hierarchy

The diagram illustrates the BGP configuration hierarchy. All BGP configuration is done within an instance, for example the default instance or a VPN service instance. The instance configuration hierarchy includes parameters required for BGP but not part of the BGP configuration hierarchy itself. The BGP instance configuration hierarchy includes parameters which are generic to the respective BGP instance. The sub-hierarchies include parameters which are specific to address families, peer groups, and peers.



### 2.2. Configuration Syntax and Commands

The following sections describe the BGP configuration syntax and commands.

#### 2.2.1. Instance Configuration

The instance configuration hierarchy includes parameters that are required for or used by BGP, but that are not part of the BGP protocol configuration hierarchy itself.

##### 2.2.1.1. Identifiers

Route distinguishers and router IDs are configured directly at the instance hierarchy.

Syntax:

**set instance** <instance-name> <attribute> <value>

Attribute	Description
route-distinguisher <as-number   ipv4-address:id>	The route distinguisher (RD) uniquely defines routes within an IPv4 network. PE routers use route distinguishers to identify which VPN a packet belongs to. Supported formats are <as-number:id> or <ipv4-address:id>.
ipv4-router-id <ipv4-address>	The router ID of the routing instance.

### Example: Instance Identifier Configuration

```

supervisor@leaf1: cfg> show config instance services
{
  "rtbrick-config:instance": {
    "name": "services",
    "ipv4-router-id": "192.168.0.3",
    "route-distinguisher": "192.168.0.3:101",
    <...>
  }
}

```

## 2.2.1.2. Address Families

At the instance address family hierarchy, you can enable or disable address families for the instance, and configure parameters like route targets.

Please note default settings depend on the instance. For the 'default' instance, the IPv4 and IPv6 unicast, multicast, and labeled-unicast, as well as the MPLS unicast address families are enabled by default. For any non-default instance, no address family is enabled by default and needs to be enabled by configuration.

Syntax:

```

set instance <instance-name> address-family <afi> <safi> <attribute>
<value>

```

Attribute	Description
<afi>	Address family identifier (AFI). Supported values: ipv4, ipv6, or mpls
<safi>	Subsequent address family identifier (SAFI). Supported values: unicast, labeled-unicast, or multicast



Attribute	Description
route-target ( import   export ) <rt-value>	Route targets (RT) are used to transfer routes between VPN instances. The RT identifies a subset of routes that should be imported to or exported from a particular VPN instance. You can configure a RT for importing or exporting routes, or both.
policy ( import   export ) <policy-name>	There are two attachment points for BGP policies. At this configuration hierarchy, you can attach import or export policies to the instance. These policies apply when routes are imported from the BGP protocol into the instance, or exported from the instance to the BGP protocol.

### Example: Instance Address Family Configuration

```

supervisor@leaf1: cfg> show config instance services
{
  "rtbrick-config:instance": {
    "name": "services",
    <...>
    "address-family": [
      {
        "afi": "ipv4",
        "safi": "unicast",
        "policy": {
          "export": "MY_V4_POLICY"
        },
        "route-target": {
          "import": "target:192.168.0.0:14",
          "export": "target:192.168.0.0:14"
        }
      },
      {
        "afi": "ipv6",
        "safi": "unicast",
        "policy": {
          "export": "MY_V6_POLICY"
        },
        "route-target": {
          "import": "target:192.168.0.0:16",
          "export": "target:192.168.0.0:16"
        }
      }
    ],
    <...>
  }
}

```

### 2.2.1.3. TCP Authentication Configuration

At the instance TCP authentication hierarchy, you can optionally enabled MD5 or HMAC SHA authentication. Technically speaking, authentication is not configured for BGP directly, but for the TCP sessions used by BGP.

Syntax:

```
set instance <instance> tcp authentication-identifier <authentication-id>
<attribute> <value>
```

Attribute	Description
<authentication-id>	Authentication identifier
authentication-type <authentication-type>	Authentication identifier such as MD5
source-port <port-number>	Port number range 0-65535. If the default value 0 is used as a port-number, then match is done for any of the source-port
destination-port <port-number>	Port number range 0-65535. If the default value 0 is used as a port-number, then match is done for any of the destination-port
destination-prefix ipv4 <ipv4_address>	Specify a valid IPv4 address, for example 10.10.10.2/24
destination-prefix ipv6 <ipv6_address>	Specify a valid IPv6 address, for example fe80::780b:9ff:fe67:1/128
source-prefix ipv4 <ipv4_address>	Specify a valid source-prefix IPv4 address, for example 10.10.10.2/24
source-prefix ipv6 <ipv6_address>	Specify a valid source-prefix IPv6 address, for example fe80::7871:b4ff:fe93:2/128
receive-key1-id <receive-key1-id>	Key ID1 of the receiver.
receive-key1-encrypted-text <receive-key1-encrypted-text>	Encrypted text of key1
receive-key1-plain-text <receive-key1-plain-text>	Plain text of key1
receive-key2-id <receive-key2-id>	Key ID2 of the receiver.
receive-key2-encrypted-text <receive-key1-encrypted-text>	Encrypted text of key2

Attribute	Description
receive-key2-plain-text <receive-key1-plain-text>	Plain text of key2
send-key-id <send-key-id>	Send key ID
send-key-encrypted-text <send-key-encrypted-text>	Encrypted text of the send key
send-key-plain-text <send-key-plain-text>	Plain text of the send key



Passwords need to be mandatorily prefixed with a zero for MD5

### Example: BGP TCP Authentication Configuration

```

supervisor@leaf2: op> show config instance default tcp
{
  "rtbrick-config:tcp": {
    "authentication-identifier": [
      {
        "authentication-id": "spine2",
        "authentication-type": "MD5",
        "destination-prefix": {
          "ipv6": "fe80::7852:68ff:fe60:202/128"
        },
        "receive-key1-id": 1,
        "receive-key1-encrypted-text": "$29e5eb79c8debef5944b1163ffca6980b",
        "send-key-id": 1,
        "send-key-encrypted-text": "$29e5eb79c8debef5944b1163ffca6980b",
        "source-prefix": {
          "ipv6": "fe80::7828:3bff:fe60:102/128"
        }
      }
    ]
  }
}

```

## 2.2.2. BGP Instance Configuration

At this configuration hierarchy you configure BGP protocol parameters which are generic to the BGP instance.

Syntax:

**set instance** <instance-name> **protocol bgp** <attribute> <value>

Attribute	Description
host-name <host-name>	The name of the BGP host, to a maximum of 64 characters
domain-name <domain-name>	The name of the BGP routing domain, to a maximum of 64 characters
enforce-first-as <enable   disable>	By default, the BGP routing process enforces the First AS feature. It 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. Disable the First AS feature to accept updates without the peer's source AS matching the first AS in the AS_PATH attribute.
local-as <as-number>	The AS number in four-byte format. The numbers allowed are from 1 to 4294967285.
local-preference <preference-value>	The local preference for the BGP protocol. The numbers allowed are from 0 to 4294967285. The local preference is used to select the exit path for an AS.
med <med-value>	The BGP Multi-Exit Discriminator (MED) value. The numbers allowed are from 0 to 4294967285. 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.
protocol-preference (internal   external) <preference-value>	Protocol preference of routes learned by eBGP ('external'), iBGP ('internal'), or both. This preference is used to select routes learned from multiple protocols.
router-id <router-id>	Router identifier in IPv4 format
cluster-id <cluster-identifier>	The cluster ID 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.
timer hold-time <seconds>	Hold timer in seconds. The valid range is 5 to 65535.
timer keepalive <seconds>	Keepalive timer in seconds. The valid range is 5 to 65535.
type-of-service cost <low   normal>	ToS cost field (bit 6) for BGP packets
type-of-service delay <low   normal>	ToS delay field (bit 3) for BGP packets
type-of-service precedence <precedence>	ToS IP precedence bits (0 - 2) for BGP packets. Valid precedences are critics, flash, flash-override, immediate, internetwork-control, precedence, network-control, priority, and routine.

Attribute	Description
type-of-service reliability <high   normal>	ToS reliability field (bit 5) for BGP packets
type-of-service throughput <high   normal>	ToS throughput field (bit 4) for BGP packets

### Example: BGP Instance Configuration

The following example shows some global BGP instance configuration attributes. The further BGP configuration like peer groups and peers is shown in the examples in the subsequent sections.

```

supervisor@spine1: cfg> show config instance default protocol bgp
{
  "rtbrick-config:bgp": {
    "cluster-id": "10.99.0.1",
    "domain-name": "rtbrick.com",
    "host-name": "spine1",
    "local-as": 4200000100,
    "local-preference": 50,
    "router-id": "10.99.0.1",
    "type-of-service": {
      "precedence": "network-control"
    },
    "protocol-preference": {
      "internal": 180,
      "external": 20
    },
    "timer": {
      "hold-time": 30,
      "keepalive": 10
    },
    <...>
  }
}

```

## 2.2.3. BGP Address Family Configuration

This configuration hierarchy refers to parameters that are specific to address families but generic to the BGP instance, as opposed to peer-group specific address families configuration. At this hierarchy, you can enable or disable address families for BGP, and configure various address family specific features.

Syntax:

```

set instance <instance-name> protocol bgp address-family <afi> <safi>
<attribute> <value>

```

Attribute	Description
<afi>	Address family identifier (AFI). Supported values: ipv4, or ipv6
<safi>	Subsequent address family identifier (SAFI). Supported values: unicast, labeled-unicast, vpn-unicast, multicast, or vpn-multicast
default-information originate <true   false>	Generate and distribute a default route information
download-count <count>	Forward packets over multiple paths, set maximum prefixes to use
multipath <number>	Enable load sharing among multiple BGP paths
retain-route-target (enable   disable)	Retain VPN routes for all route targets, by default this feature is enabled
resolve-nexthop afi <afi>	Address family to resolve the nexthop
resolve-nexthop safi <safi>	Sub-address family to resolve the nexthop
redistribute <source>	Enable the redistribution feature to dynamically inject specific types of routes into the BGP protocol. Supported route sources are direct, static, ppp, igmp, pim, and isis.
redistribute <source> policy <policy>	Attach a policy to the redistribution process
srgb base <value>	Segment Routing Global Block (SRGB) start label. The 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. SRGB is supported for labeled unicast only.
srgb index <value>	Segment Routing Global Block (SRGB) index
srgb range <value>	Segment Routing Global Block (SRGB) label range

### Example 1: BGP Address Family Configuration with Segment Routing

```
supervisor@spinel: cfg> show config instance default protocol bgp
{
  "rtbrick-config:bgp": {
    <...>
    "address-family": [
      {
        "afi": "ipv4",
        "safi": "vpn-unicast"
      },
      {
        "afi": "ipv6",
        "safi": "labeled-unicast",
        "srgb": {
          "base": 5000,
          "range": 1000,
          "index": 11
        }
      },
      {
        "afi": "ipv6",
        "safi": "unicast"
      },
      {
        "afi": "ipv6",
        "safi": "vpn-unicast"
      }
    ],
    <...>
  }
}
```

### Example 2: BGP Address Family Configuration with Redistribution

```
supervisor@leaf1: cfg> show config instance services protocol bgp
{
  "rtbrick-config:bgp": {
    <...>
    "address-family": [
      {
        "afi": "ipv4",
        "safi": "unicast",
        "redistribute": [
          {
            "source": "direct"
          },
          {
            "source": "ppp"
          },
          {
            "source": "static"
          }
        ]
      },
      {
        "afi": "ipv6",
        "safi": "unicast",
        "redistribute": [
          {
            "source": "direct"
          },
          {
            "source": "ppp"
          },
          {
            "source": "static"
          }
        ]
      }
    ]
  }
}
```

Example 3: BGP Address Family Configuration with Redistribution and Redistribution Policy



```
supervisor@leaf1: cfg> show config instance services protocol bgp
{
  "rtbrick-config:bgp": {
    <...>
    "address-family": [
      {
        "afi": "ipv4",
        "safi": "unicast",
        "redistribute": [
          {
            "source": "direct"
            "policy": "MY_REDISTRIBUTION_POLICY"
          },
          {
            "source": "ppp"
          },
          {
            "source": "static"
          }
        ]
      },
      {
        "afi": "ipv6",
        "safi": "unicast",
        "redistribute": [
          {
            "source": "direct"
            "policy": "MY_REDISTRIBUTION_POLICY"
          },
          {
            "source": "ppp"
          },
          {
            "source": "static"
          }
        ]
      }
    ]
  }
}
```

## 2.2.4. Peer Group Configuration

### 2.2.4.1. Peer Groups

In BGP, neighbor peers with the same update policies can be grouped 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.

Syntax:

```
set instance <instance-name> protocol bgp peer-group <peer-group-name>
```

<attribute> <value>

Attribute	Description
local-as <as-number>	Local AS number for the peer group
remote-as <as-number>	Remote AS number for the peer group
any-as <true   false>	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 <true   false>	Enable BGPv6 peerings using the IPv6 link-local addresses
no-prepend <true   false>	Do not prepend the local AS for advertisements to the peer
replace-as <true   false>	Prepend only the local AS for advertisements to the peer

### 2.2.4.2. Address Families

At this configuration hierarchy, you can enable the address families that shall be supported for the group peers, and enable address family specific features. By default, BGP neighbor sessions support the IPv4 unicast and multicast address families.

Syntax:

**set instance** <instance-name> **protocol bgp peer-group** <peer-group-name>  
**address-family** <afi> <safi> <attribute> <value>

Attribute	Description
<afi>	Address family identifier (AFI). Supported values: ipv4, or ipv6
<safi>	Subsequent address family identifier (SAFI). Supported values: unicast, labeled-unicast, vpn-unicast, multicast, or vpn-multicast
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 originate <true   false>	Generate and advertise a default route to peers in group
extended-nexthop	Enable extended nexthop capability so that peers in the group do not have to be directly connected

<b>Attribute</b>	<b>Description</b>
nexthop-self <true   false>	Set the advertised BGP nexthop to yourself, this is the default for eBGP
nexthop-unchanged <true   false>	Do not modify the advertised BGP nexthop, this is the default for iBGP
update-nexthop ( ipv4-address   ipv6-address ) <address>	BGP nexthop address for routes advertised to this peer group
remove-private-as <true   false>	Remove private AS numbers from routes advertised to group peers
route-reflect-client <true   false>	Configure this peer as a route reflector client
policy ( import   export ) <policy-name>	Apply a routing policy to the peer group

Example: BGP Peer Group Configuration

```

supervisor@leaf1: cfg> show config instance default protocol bgp peer-group
spine
{
  "rtbrick-config:peer-group": {
    "pg-name": "spine",
    "link-local-nexthop-only": "true",
    "remote-as": 4200000100,
    "address-family": [
      {
        "afi": "ipv4",
        "safi": "vpn-unicast",
        "extended-nexthop": "true",
        "update-nexthop": {
          "ipv6-address": "fd3d:3d:0:99::3"
        }
      },
      {
        "afi": "ipv6",
        "safi": "labeled-unicast"
      },
      {
        "afi": "ipv6",
        "safi": "unicast"
      },
      {
        "afi": "ipv6",
        "safi": "vpn-unicast",
        "update-nexthop": {
          "ipv6-address": "fd3d:3d:0:99::3"
        }
      }
    ]
  }
}

```

## 2.2.5. Peer Configuration

Once peer groups have been defined, BGP peers can be configured at the peer configuration hierarchy. A peer can be specified by address, or by interface when using IPv6 auto-discovered neighbors and link-local addresses.

Syntax to configure a BGP peer by address:

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

Syntax to configure a BGP peer using IPv6 link-local addresses:

```
set instance <instance-name> protocol bgp peer interface <name> peer-group
<peer-group>
```

Attribute	Description
interface <name>	Enable BGP peer using IPv6 link-local addresses
ipv4 <peer-address>	IPv4 address of a BGP peer
ipv6 <peer-address>	IPv6 address of a BGP peer
<update-source>	Local IP address to be used for the peering
peer-group <peer-group>	Assign the peer to a peer group
deactivate	Deactivate a configured peer

### Example 1: BGP peer specified by IP addresses

```

supervisor@rtbrick: cfg> show config instance default protocol bgp peer

{
  "rtbrick-config:peer": {
    "ipv4": [
      {
        "peer-address": "10.0.100.2",
        "update-source": "10.0.100.1",
        "peer-group": "spine"
      }
    ]
  }
}

```

### Example 2: BGP peer using IPv6 link-local addresses

```

supervisor@rtbrick: cfg> show config instance default protocol bgp peer

{
  "rtbrick-config:peer": {
    "interface": [
      {
        "name": "ifp-0/0/1/1",
        "peer-group": "spine"
      }
    ]
  }
}

```

## 2.3. Sample Configuration

### Example 1: BGP Configuration of a Spine Switch (Default Instance only)

```

{
  "data": {
    "rtbrick-config:instance": [

```

```

{
  "name": "default",
  "ipv4-router-id": "10.99.0.1",
  "protocol": {
    "bgp": {
      "domain-name": "rtbrick.com",
      "host-name": "spine1",
      "local-as": 4200000100,
      "address-family": [
        {
          "afi": "ipv4",
          "safi": "vpn-unicast"
        },
        {
          "afi": "ipv6",
          "safi": "labeled-unicast",
          "srgb": {
            "base": 5000,
            "range": 1000,
            "index": 11
          },
          "redistribute": [
            {
              "source": "direct"
            }
          ]
        },
        {
          "afi": "ipv6",
          "safi": "unicast",
          "redistribute": [
            {
              "source": "direct"
            }
          ]
        },
        {
          "afi": "ipv6",
          "safi": "vpn-unicast"
        }
      ],
      "peer": {
        "interface": [
          {
            "name": "memif-0/1/1/1",
            "peer-group": "spine"
          },
          {
            "name": "memif-0/2/1/1",
            "peer-group": "leaf1"
          },
          {
            "name": "memif-0/2/2/1",
            "peer-group": "leaf2"
          }
        ]
      },
      "peer-group": [

```

```
{
  "pg-name": "leaf1",
  "link-local-nextthop-only": "true",
  "remote-as": 4200000201,
  "address-family": [
    {
      "afi": "ipv4",
      "safi": "vpn-unicast",
      "extended-nextthop": "true",
      "nextthop-unchanged": "true"
    },
    {
      "afi": "ipv6",
      "safi": "labeled-unicast"
    },
    {
      "afi": "ipv6",
      "safi": "unicast"
    },
    {
      "afi": "ipv6",
      "safi": "vpn-unicast",
      "nextthop-unchanged": "true"
    }
  ]
},
{
  "pg-name": "leaf2",
  "link-local-nextthop-only": "true",
  "remote-as": 4200000202,
  "address-family": [
    {
      "afi": "ipv4",
      "safi": "vpn-unicast",
      "extended-nextthop": "true",
      "nextthop-unchanged": "true"
    },
    {
      "afi": "ipv6",
      "safi": "labeled-unicast"
    },
    {
      "afi": "ipv6",
      "safi": "unicast"
    },
    {
      "afi": "ipv6",
      "safi": "vpn-unicast",
      "nextthop-unchanged": "true"
    }
  ]
},
{
  "pg-name": "spine",
  "link-local-nextthop-only": "true",
  "remote-as": 4200000100,
  "address-family": [

```

```

        "afi": "ipv4",
        "safi": "vpn-unicast",
        "extended-nextthop": "true"
      },
      {
        "afi": "ipv6",
        "safi": "labeled-unicast",
        "nextthop-self": "true"
      },
      {
        "afi": "ipv6",
        "safi": "unicast",
        "nextthop-self": "true"
      },
      {
        "afi": "ipv6",
        "safi": "vpn-unicast"
      }
    ]
  }
}

```

### Example 2: BGP Configuration of a Leaf Switch with one VPN Instance

```

{
  "data": {
    "rtbrick-config:instance": [
      {
        "name": "default",
        "ipv4-router-id": "10.99.0.3",
        "protocol": {
          "bgp": {
            "domain-name": "rtbrick.com",
            "host-name": "leaf1",
            "local-as": 4200000201,
            "address-family": [
              {
                "afi": "ipv4",
                "safi": "vpn-unicast"
              },
              {
                "afi": "ipv6",
                "safi": "labeled-unicast",
                "srgb": {
                  "base": 5000,
                  "range": 1000,
                  "index": 13
                }
              },
              {
                "redistribute": [
                  {

```



```

        "source": "direct"
      }
    ]
  },
  {
    "afi": "ipv6",
    "safi": "unicast",
    "redistribute": [
      {
        "source": "direct"
      }
    ]
  },
  {
    "afi": "ipv6",
    "safi": "vpn-unicast"
  }
],
"peer": {
  "interface": [
    {
      "name": "memif-0/1/1/1",
      "peer-group": "spine"
    },
    {
      "name": "memif-0/1/2/1",
      "peer-group": "spine"
    }
  ]
},
"peer-group": [
  {
    "pg-name": "spine",
    "link-local-nexthop-only": "true",
    "remote-as": 4200000100,
    "address-family": [
      {
        "afi": "ipv4",
        "safi": "vpn-unicast",
        "extended-nexthop": "true",
        "update-nexthop": {
          "ipv6-address": "fd3d:3d:0:99::3"
        }
      },
      {
        "afi": "ipv6",
        "safi": "labeled-unicast"
      },
      {
        "afi": "ipv6",
        "safi": "unicast"
      },
      {
        "afi": "ipv6",
        "safi": "vpn-unicast",
        "update-nexthop": {
          "ipv6-address": "fd3d:3d:0:99::3"
        }
      }
    ]
  }
]

```

```

    }
  ]
}
]
}
},
{
  "name": "services",
  "ipv4-router-id": "192.168.0.3",
  "route-distinguisher": "192.168.0.3:101",
  "address-family": [
    {
      "afi": "ipv4",
      "safi": "unicast",
      "policy": {
        "export": "MY_V4_POLICY"
      },
      "route-target": {
        "import": "target:192.168.0.0:14",
        "export": "target:192.168.0.0:14"
      }
    },
    {
      "afi": "ipv6",
      "safi": "unicast",
      "policy": {
        "export": "MY_V6_POLICY"
      },
      "route-target": {
        "import": "target:192.168.0.0:16",
        "export": "target:192.168.0.0:16"
      }
    }
  ],
  "protocol": {
    "bgp": {
      "domain-name": "rtbrick.com",
      "host-name": "leaf1",
      "local-as": 65003,
      "address-family": [
        {
          "afi": "ipv4",
          "safi": "unicast",
          "redistribute": [
            {
              "source": "direct"
            },
            {
              "source": "ppp"
            },
            {
              "source": "static"
            }
          ]
        }
      ],
      {
        "afi": "ipv6",

```

```
"safi": "unicast",
  "redistribute": [
    {
      "source": "direct"
    },
    {
      "source": "ppp"
    },
    {
      "source": "static"
    }
  ]
}
]
}
}
]
}
}
}
```

---

## 3. BGP Operational Commands

### 3.1. BGP Show Commands

The BGP show commands provide detailed information about the BGP protocol operation and BGP routes.

#### 3.1.1. BGP Summary

This command displays BGP protocol parameters like attributes or timers that are generic to the BGP instance.

Syntax:

**show bgp summary** <option>

Option	Description
-	Without any option, the commands displays the information for all instances.
instance <instance-name>	BGP summary information for the given instance.

Example: BGP summary for the default instance

```

supervisor@rtbrick: op> show bgp summary instance default
Instance: default
  General information
    Hostname: PE1, Domain name:
    Local AS: 1000, Version: 4
    Local preference: 100, Protocol preference: 200
    Router ID: 192.0.0.2, Cluster ID: 192.0.0.2
  Capabilities
    Route refresh: True, AS4: True, Graceful restart: False
  Best route selection
    Always compare MED: False, Ignore as path: False
    Ignore local preference: False, Ignore origin: False
    Ignore MED: False, Ignore route source: False
    Ignore router ID: False, Ignore uptime: True
    Ignore cluster length: False, Ignore peer IP: False
    Route select parameter: 0
  Timers
    Connect retry: 30s, Keepalive: 30s, Holdtime: 90s
  Statistics
    Peers configured: 1, Peers auto discovery: 0
    Peers in idle           : 0
    Peers in connect       : 0
    Peers in active        : 0
    Peers in opensent      : 0
    Peers in openconfirm   : 0
    Peers in established   : 1

```

### 3.1.2. BGP Peer

The 'show bgp peer' commands displays information on BGP peers.

Syntax:

**show bgp peer** <option> ...

Option	Description
-	Without any option, the commands displays all BGP peers in all instances in a summary table format.
detail	Detailed information on all BGP peers in all instances in a list view.
<peer-name>	Detailed information on the peer with the given name.
address <peer-address>	Detailed information on the peer with the given IP address.
instance <instance-name>	Summary of all BGP peers in the given instance.
instance <instance-name> detail	Detailed information on all BGP peers in the given instance.

Option	Description
instance <instance-name> detail <peer-name>	Detailed information on the peer with the given name in the given instance.
instance <instance-name> detail address <peer-address>	Detailed information on the peer with the given IP address in the given instance.
statistics	Received and sent BGP prefixes per AFI/SAFI for all peers in all instances.
statistics peer <peer-name>	Received and sent BGP prefixes per AFI/SAFI for the peer with the given name.
statistics peer address <peer-address>	Received and sent BGP prefixes per AFI/SAFI for the peer with the given IP address.
statistics instance <instance-name> peer <peer-name>	Received and sent BGP prefixes per AFI/SAFI for the peer with the given name in the given instance.
statistics instance <instance-name> peer address <peer-address>	Received and sent BGP prefixes per AFI/SAFI for the peer with the given IP address in the given instance.

### Example 1: BGP Peer Summary View

```

supervisor@rtbrick: op> show bgp peer
Instance: default
  Peer                               Remote AS   State      Up/Down
Time                               PfxRcvd    PfxSent
  PE2                               2000       Established
11d:22h:18m:30s                    12         20
Instance: default
  Peer                               Remote AS   State      Up/Down
Time                               PfxRcvd    PfxSent
  CE1                               65535      Established
6d:02h:28m:02s                      2          2
  CE1                               65535      Established
6d:02h:27m:45s                      2          2
supervisor@rtbrick: op>

```

### Example 2: BGP Peer Detail View

```

supervisor@rtbrick: op> show bgp peer detail
Peer: PE2, Peer IP: 23.0.0.3, Remote AS: 2000, Local: 23.0.0.2, Local AS:
1000, Any AS: False
  Type: ebgp, State: Established, Uptime: 11d:22h:18m:48s, Reason: Cease,
Sub-Code: Admin shutdown
  Discovered on interface: -
  Last transition: Thu Nov 19 05:33:28 GMT +0000 2020, Flap count: 1
  Peer ID           : 192.0.0.10, Local ID   : 192.0.0.2
  Instance          : default, Peer group: to_pe2

```

```

6PE enabled      : False
Timer values:
  Peer keepalive : 30s, Local keepalive: 30s
  Peer holddown  : 90s, Local holddown  : 90s
  Connect retry  : 30s
Timers:
  Connect retry timer : 0s
  keepalive timer     : expires in 1s 488011us
  Holddown timer      : expires in 1m 15s 85437us
NLRIs:
  Sent                : ['ipv6-unicast', 'ipv4-vpn-unicast', 'ipv6-vpn-unicast',
'ipv6-labeled-unicast']
  Received            : ['ipv6-unicast', 'ipv6-labeled-unicast', 'ipv4-vpn-
unicast', 'ipv6-vpn-unicast']
  Negotiated         : ['ipv6-unicast', 'ipv6-labeled-unicast', 'ipv4-vpn-
unicast', 'ipv6-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           never                               never
  IPv4 labeled-unicast  never                               never
  IPv6 unicast           Thu Nov 19 05:33:30 GMT +0000 2020 Thu
Nov 19 05:33:30 GMT +0000 2020
  IPv6 labeled-unicast  Thu Nov 19 05:33:30 GMT +0000 2020 Thu
Nov 19 05:33:30 GMT +0000 2020
  IPv4 VPN-unicast      Thu Nov 19 05:33:30 GMT +0000 2020 Thu
Nov 19 05:33:30 GMT +0000 2020
  IPv6 VPN-unicast      Thu Nov 19 05:33:30 GMT +0000 2020 Thu
Nov 19 05:33:30 GMT +0000 2020
  IPv4 VPN-multicast    never                               never
Message stats:
Session stats:
  Direction  Open      Update      Keepalive    Notify      Route
refresh
  Input      1         38          41196        0           0
  Output     1         22          41207        0           0
Total stats:
  Input      2         48          44618        1           0
  Output     3         32          44624        0           0
Route stats:
  Address family         Received      Sent
  IPv4 unicast           0             0
  IPv4 labeled-unicast  0             0
  IPv6 unicast           2             3
  IPv6 labeled-unicast  2             3

```

```

IPv4 VPN-unicast          4          7
IPv6 VPN-unicast          4          7
IPv4 multicast            0          0
IPv4 VPN-multicast        0          0
<...>

```

### Example 3: BGP Peer Statistics

```

supervisor@rtbrick: op> show bgp peer statistics instance default peer PE2
Instance: default
Peer                AFI      SAFI                PfxRcvd
PfxSent
PE2
  ipv4      unicast          0          0
  ipv4      labeled-unicast 0          0
  ipv6      unicast          2          3
  ipv6      labeled-unicast 2          3
  ipv4      vpn-unicast      4          7
  ipv6      vpn-unicast      4          7
  ipv4      multicast        0          0
  ipv4      vpn-multicast    0          0

supervisor@rtbrick: op>

```

## 3.1.3. BGP Peer Group

The 'show bgp peer-group' commands displays parameters like BGP attributes that are specific to the respective peer groups.

Syntax:

**show bgp peer-group** <option> ...

Option	Description
-	Without any option, the commands displays information on all peer groups in all instances.
<peer-group-name>	Information on the peer group with the given name.
instance <instance-name>	All peer groups in the given instance.
instance <instance-name> <peer-group-name>	Information on the peer group with the given name in the given instance.

Example: BGP Peer Group



```

supervisor@rtbrick: op> show bgp peer-group to_pe2
Instance: default
  Peer group name      : to_pe2
  Remote AS           : 2000
  Import rule         : None
  Export rule         : None
  Remove AS           : None
  Nexthop self        : None
  Multipath iBGP      : None
  Multipath eBGP      : None
  Client-to-Client    : None
  Add path            : None
  eBGP multihop       : None
  Hop (TTL)           : None
  Any AS              : None
  Update VPNv4 NH     : None
  Update MVPN NH      : None
supervisor@rtbrick: op>

```

### 3.1.4. BGP FIB

The 'show bgp fib' commands display the BGP forwarding table. In contrast to the 'show bgp rib' commands, the output of the 'show bgp fib' commands includes only the selected routes. The BGP route selection occurs between the RIB and the FIB.

Syntax:

**show bgp fib** <option> ...

Option	Description
-	Without any option, the commands displays the BGP forwarding table for all address families and all instances in a summary table format.
<afi>	BGP forwarding table summary for the given address family (AFI), all sub-address families and all instances. Supported AFI values are 'ipv4' and 'ipv6'.
<afi> <safi>	BGP forwarding table summary for the given address family (AFI) and sub-address family (SAFI), and all instances. Supported SAFI values are 'unicast', 'labeled-unicast', 'multicast', and 'vpn-unicast'.
<afi> <safi> detail	Detailed list view of the BGP forwarding table for the given address family (AFI) and sub-address family (SAFI), and all instances.
<afi> <safi> <prefix>	BGP forwarding table entry for the given prefix and all instances.

Option	Description
<afi> <safi> instance <instance-name>	BGP forwarding table summary for the given AFI, SAFI, and instance.
<afi> <safi> instance <instance-name> detail	Detailed list view of BGP forwarding table for the given AFI, SAFI, and instance.
<afi> <safi> instance <instance-name> <prefix>	BGP forwarding table entry for the given prefix and instance.

### Example 1: Summary view of the BGP FIB for IPv6, and all SAFIs and all instances

```

supervisor@rtbrick: op> show bgp fib ipv6
Instance: default, AFI: ipv6, SAFI: unicast
  Prefix                Preference    Out Label
Next Hop
  192::3/128            20           -
23.0.0.3
  192::10/128           20           -
23.0.0.3
Instance: services, AFI: ipv6, SAFI: unicast
  Prefix                Preference    Out Label
Next Hop
  10::1/128             200          -
12::1
Instance: default, AFI: ipv6, SAFI: labeled-unicast
  Prefix                Preference    Out Label
Next Hop
  192::3/128            20           2003
23.0.0.3
  192::10/128           20           2003
23.0.0.3
Instance: default, AFI: ipv6, SAFI: vpn-unicast
  Prefix                Preference    Out Label
Next Hop
  12::/64               200          20003,bos:1
  10::1/128             200          20003,bos:1
  192:0:1::2/128        200          20003,bos:1
  34::/64               20           20006,bos:1
192::3
  10::4/128             20           20006,bos:1
192::3
  192:0:1::3/128        20           20006,bos:1
192::3
  192:0:4::10/128      20           20006,bos:1
192::3

```

### Example 2: Detailed view of the BGP FIB for IPv6 VPN unicast routes in the default instances

```

supervisor@rtbrick: op> show bgp fib ipv6 vpn-unicast instance default detail
Instance: default, AFI: ipv6, SAFI: vpn-unicast
  Prefix: 12::/64
    Next hop key: 2b38f6f1d2ae56178666d1edcffd18a85fd4509bcac9a21f
    Peer: None, Peer domain: None
    Route source: bgp-local, Send path ID: 405188370, Received path ID: None,
Path hash: None
    As path: None, Originator ID: None, Origin: Incomplete
    Community: None
    Extended community: ['target:192.0.1.2:2']
    Cluster list: None
    IGP metric: None, Local preference: 100, Multi exit discriminator: 0
    Preference: 200, External route: None, Readvertised route: None
    Label: 20003,bos:1, Route up: None
  Prefix: 10::1/128
    Next hop key: 62b6c375c2ee2cb053bd5482ec1b7df18e271b6e0d37a4b0
    Peer: None, Peer domain: None
    Route source: bgp-local, Send path ID: 2400017309, Received path ID:
None, Path hash: None
    As path: None, Originator ID: None, Origin: Incomplete
    Community: None
    Extended community: ['target:192.0.1.2:2']
    Cluster list: None
    IGP metric: None, Local preference: 100, Multi exit discriminator: None
    Preference: 200, External route: None, Readvertised route: None
    Label: 20003,bos:1, Route up: None

```

### 3.1.5. BGP RIB-in

This command displays the received routes.

Syntax:

**show bgp rib-in** <option> ...

Option	Description
-	Without any option, the command displays information of the received BGP routing table on all instances in a summary table format.
<afi>	BGP routing table summary for the given address family (AFI), all sub-address families and all instances. Supported AFI values are 'ipv4' and 'ipv6'.
<afi> <safi>	BGP routing table summary for the given address family (AFI) and sub-address family (SAFI), and all instances. Supported SAFI values are 'labeled-unicast', 'unicast', 'vpn-multicast', and 'vpn-unicast'.
<afi> <safi> detail	Detailed list view of the BGP routing table for the given address family (AFI) and sub-address family (SAFI), and all instances.

Option	Description
<afi> <safi> <prefix>	BGP routing table entry for the given prefix and all instances.
<afi> <safi> instance <instance-name>	BGP routing table summary for the given AFI, SAFI, and instance.
<afi> <safi> instance <instance-name> detail	Detailed list view of BGP routing table for the given AFI, SAFI, and instance.
<afi> <safi> instance <instance-name> <prefix>	BGP routing table entry for the given prefix and instance.
<afi> <safi> peer <name> / peer address <ip>	Peer name or address

### Example 1: Summary view of the BGP rib-in.

```

supervisor@rtbrick: op> show bgp rib-in
Instance: ip2vrf, AFI: ipv4, SAFI: unicast
Peer: None, Received routes: 10
  Prefix                               Next Hop                                MED
Local Preference  AS Path
10.0.27.0/31      -                                         192.1.4.2      -
100
10.0.68.0/31     -                                         192.0.0.8      -
100
10.0.69.0/31     -                                         192.0.0.9      -
100
10.0.78.0/31     -                                         192.0.0.8      -
100
10.0.79.0/31     -                                         192.0.0.9      -
100
10.0.89.0/31     -                                         192.0.0.8      -
100
192.1.4.2/32     -                                         192.1.4.2      -
100
192.0.0.8/32     -                                         192.0.0.8      -
100
192.0.0.9/32     -                                         192.0.0.9      -
100
192.0.0.99/32    -                                         192.0.0.9      -
100
Instance: default, AFI: ipv4, SAFI: vpn-unicast
Peer: None, Received routes: 4
  Prefix                               Next Hop                                MED
Local Preference  AS Path
192.1.1.4/32     4200000004 192:1::4      0      -
192.1.2.4/32     4200000004 192:1::4      0      -
192.1.3.4/32     4200000004 192:1::4      0      -

```

### Example 2: Summary view of the BGP rib-in for IPv4, and all SAFIs and all instances

```
supervisor@rtbrick: op> show bgp rib-in ipv4
Instance: ip2vrf, AFI: ipv4, SAFI: unicast
Peer: None, Received routes: 10
  Prefix                               Next Hop                               MED
Local Preference AS Path
  10.0.27.0/31          192.1.4.2          -
100
  10.0.68.0/31         192.0.0.8          -
100
  10.0.69.0/31         192.0.0.9          -
100
  10.0.78.0/31         192.0.0.8          -
100
  10.0.79.0/31         192.0.0.9          -
100
  10.0.89.0/31         192.0.0.8          -
100
  192.1.4.2/32         192.1.4.2          -
100
  192.0.0.8/32         192.0.0.8          -
100
  192.0.0.9/32         192.0.0.9          -
100
  192.0.0.99/32        192.0.0.9          -
100
Instance: default, AFI: ipv4, SAFI: vpn-unicast
Peer: None, Received routes: 4
  Prefix                               Next Hop                               MED
Local Preference AS Path
  192.1.1.4/32          192:1::114          0
4200000004
  192.1.2.4/32         192:1::114          0
4200000004
```

Example 3: Summary view of the received routes

```

supervisor@rtbrick: op> show bgp rib-in ipv4 unicast peer address 192.0.0.8
Instance: ip2vrf, AFI: ipv4, SAFI: unicast
Peer: None, Received routes: 13
  Prefix                               Next Hop                               MED
Local Preference  AS Path
10.0.27.0/31      -                               192.1.4.2      -
100
10.0.68.0/31     -                               192.0.0.8      -
100
10.0.69.0/31     -                               192.0.0.9      -
100
10.0.78.0/31     -                               192.0.0.8      -
100
10.0.79.0/31     -                               192.0.0.9      -
100
10.0.89.0/31     -                               192.0.0.8      -
100
192.1.4.2/32     -                               192.1.4.2      -
100
192.1.4.3/32     4200000003                    192.1.4.2      -
100
192.1.4.4/32     4200000004                    192.1.4.2      -
100
192.0.0.8/32     -                               192.0.0.8      -
100
192.0.0.9/32     -                               192.0.0.9      -
100
192.0.0.99/32    -                               192.0.0.9      -
100
supervisor@rtbrick: op>

```

### 3.1.6. BGP RIB-out

This command displays the send routes.

Syntax:

**show bgp rib-out** <option> ...

Option	Description
-	Without any option, the command displays advertised BGP routes for all instances.
<afi>	BGP routing table summary for the given address family (AFI), all sub-address families and all instances. Supported AFI values are 'ipv4' and 'ipv6'.
<afi> <safi>	BGP routing table summary for the given address family (AFI) and sub-address family (SAFI), and all instances. Supported SAFI values are 'unicast', 'labeled-unicast', 'multicast', and 'vpn-unicast'.
<afi> <safi> detail	Detailed list view of the BGP routing table for the given address family (AFI) and sub-address family (SAFI), and all instances.

Option	Description
<afi> <safi> <prefix>	BGP routing table entry for the given prefix and all instances.
<afi> <safi> instance <instance-name>	BGP routing table summary for the given AFI, SAFI, and instance.
<afi> <safi> instance <instance-name> detail	Detailed list view of BGP routing table for the given AFI, SAFI, and instance.
<afi> <safi> instance <instance-name> <prefix>	BGP routing table entry for the given prefix and instance.
<afi> <safi> peer <name> / peer address <ip>	Peer name or address

### Example 1: Summary view of the routes advertised to a peer

```

supervisor@rtbrick: op> show bgp rib-in ipv4 unicast peer address 192.0.0.8
Instance: ip2vrf, AFI: ipv4, SAFI: unicast
Peer: None, Received routes: 11
  Prefix                               Next Hop                               MED
Local Preference  AS Path
10.0.27.0/31      -                               192.1.4.2      -
100
10.0.68.0/31     -                               192.0.0.8      -
100
10.0.69.0/31     -                               192.0.0.9      -
100
10.0.78.0/31     -                               192.0.0.8      -
100
10.0.79.0/31     -                               192.0.0.9      -
100
10.0.89.0/31     -                               192.0.0.8      -
100
192.1.4.2/32     -                               192.1.4.2      -
100
192.1.4.3/32     -                               192.1.4.2      -
100
192.0.0.8/32     4200000003                       192.0.0.8      -
100
192.0.0.9/32     -                               192.0.0.9      -
100
192.0.0.99/32    -                               192.0.0.9      -
100
supervisor@rtbrick: op>

```

### Example 2: Detailed view of the routes advertised to a peer

```

supervisor@rtbrick: op> show bgp rib-in ipv4 unicast peer address 192.0.0.8
detail
Instance: ip2vrf, AFI: ipv4, SAFI: unicast
Peer: None, Received routes: 12
  10.0.27.0/31, Received path ID: 0, Next hop: 192.1.4.2
    Protocol source: bgp, Send path ID: 2135053547
    AS path: -
    MED: -, Local preference: 100
    Community: -
    Extended community: -
    Large community: -
    Originator ID: 192.1.4.2
    Cluster list: ['192.0.0.8']
    Last update: Never
  10.0.68.0/31, Received path ID: 0, Next hop: 192.0.0.8
    Protocol source: bgp, Send path ID: 2135053547
    AS path: -
    MED: -, Local preference: 100
    Community: -
    Extended community: -
    Large community: -
    Originator ID: -
    Cluster list: -
    Last update: Never

```

### 3.1.7. TCP Connections

This command displays information of the TCP connections used by BGP.

Syntax:

**show bgp tcp bgp.ioc.1 connection** <option> ...

Option	Description
-	Without any option, the command displays the TCP connections used by BGP for all instances.
detail	Detailed list view of the the TCP connections for the given address family (AFI) and sub-address family (SAFI), and all instances.
prefix	TCP connections for the given prefix and all instances.
instance <instance-name>	TCP connections summary for the given instance.

Example 1: Summary view of the BGP TCP connections



```

supervisor@rtbrick: op> show bgp tcp bgp.iod.1 connection
Instance          Local IP Address          Remote IP Address
Local Port  Remote Port  State
default      23.0.0.2          23.0.0.3
179          59614            Established
default      12.0.0.2          12.0.0.1
179          59612            Established
default      12::2             12::1
58143       179              Established

```

## Example 2: Detailed information of the BGP TCP connections

```

supervisor@rtbrick: op> show bgp tcp bgp.iod.1 connection detail
Instance: default
  Local IPv4 address      : 23.0.0.2
  Remote IPv4 address     : 23.0.0.3
  Local port              : 179
  Remote port             : 59614
  State                   : Established
  Internal
    Options                : -- | Keepalive | --
    TOS                    : 0
    TTL                    : 1
    Priority                : 1
    Flags                   : -|-|-|-|Nagle Disabled|-|Wnd Scale|-|-|-
    Last trigger           : 139
    Timer                  : 2239640
  Timers
    Poll                   : 0s
    Poll interval         : 0s
    Retransmission        : 65535s
  Receiver
    Expected sequence     : 1900007
    Available window      : 98304
    Announced window     : 98038
    Announced wnd RT edge : 1998045
    MSS                   : 1460
    RTT estimate          : 0
  Timeout
    Sequence              : 1164345s
    Retransmission        : 3s
    Retransmissions       : 0s
    Duplicate acks        : 0s
    Highest ack'd sequence : 1164364s
  Congestion
    Window                : 16063
    Persist count         : 0
    Send scale            : 5
    Receive scale         : 5
  Sender
    Next seq to send      : 1164364
    Last wnd update seq   : 1900007
    Last wnd update ack   : 1164364
    Window                : 97216

```

```

Max window announced      : 98304
Acknowledged               : None
Send buf                   : 57344
Send queue length         : 0
Unsent oversize           : 0
TS last ack sent          : 448397312
Keepalive
  Next keepalive idle      : 7200000
  Keepalive interval       : 75000
  Keepalive count          : 9
  Keep sent count          : 0
Authentication
  Auth type                : None
  Send key                  : None
  Receive key1              : None
  Receive key2              : None
  Algorithm mismatch        : None
  Secret mismatch          : None
  Latest sent digest        : None
  Latest received digest    : None

```

### 3.1.8. TCP Statistics

This command displays TCP statistics information of the TCP connections used by BGP.

Syntax:

**show bgp tcp bgp.ioc.1 statistics** <option> ...

Option	Description
-	Without any option, the command displays the TCP statistics information of the TCP connections used by BGP for all instances.
instance <instance-name>	TCP connections summary for the given instance.

Example: TCP statistics information of the TCP connections used by BGP for the default instance

```

supervisor@rtbrick: op> show bgp tcp bgp.iod.1 statistics instance default
Instance: default
  IP statistics
    Transmitted packets      : 3103242412
    Received packets        : 47351
    Forwarded packets        : 0
    Dropped packets         : 0
    Checksum error           : 0
    Invalid length error    : 0
    Out of memory error     : 0
    Routing error           : 0
    Protocol error          : 0
    Error in options        : 0
    Misc error              : 0
    Cachehit                : 0
  TCP statistics
    Transmitted packets      : 365499779
    Received packets        : 5577
    Forwarded packets        : 3014656
    Dropped packets         : 46
    Checksum error           : 0
    Invalid length error    : 0
    Out of memory error     : 0
    Routing error           : 3014656
    Protocol error          : 46
    Error in options        : 0
    Misc error              : 2097152
    Cachehit                : 1557594144

```

## 3.2. BGP Clear Commands

Clear commands allow to reset operational states.

### 3.2.1. BGP Peer

This commands resets BGP peerings.

Syntax:

**clear bgp peer** <option> ...

Option	Description
all	Clears all the BGP peers.
all soft-in <afi> <safi>	Sends route refresh to all neighbors.
all soft-out <afi> <safi>	Re-advertises all the routes previously sent to the peer.
all stats	Clears the statistics of all the BGP peers.

Option	Description
instance <instance> <peer-ip>	Clears the peer for the given instance and peer IP address.
instance <instance> <peer-ip> source <src-ip>	Clears a specific peer for the given peer IP address and source IP address in the specified instance.
instance <instance> all	Clears all peers in the given instance.
instance <instance> <peer-ip> source <src-ip> soft-in <afi> <safi>	Sends route refresh to specific peer for the given instance, peer-ip, source-ip and address-family.
instance <instance> <peer-ip> soft-in <afi> <safi>	Sends route refresh to peer for the given instance, peer-ip and address-family.
instance <instance> all soft-in <afi> <safi>	Sends route refresh to all peers for the given instance and address-family.
instance <instance> <peer-ip> source <src-ip> soft-out <afi> <safi>	Re-advertises all the routes previously sent to the specific peer for the given instance, peer-ip, source-ip and address-family.
instance <instance> <peer-ip> soft-out <afi> <safi>	Sends route refresh to peer for given instance, peer-ip and address-family.
instance <instance> all soft-out <afi> <safi>	Sends route update to all peers for given instance and address-family.
instance <instance> <peer-ip> source <src-ip> stats	Clears the statistics of a specific peer for given instance, peer-ip and source-ip.
instance <instance> <peer-ip> stats	Clears the statistics of the peer for given instance and peer-ip.
instance <instance> all stats	Clears the statistics of all peers for given instance.

Example: The example below shows how to clear all the BGP peers.

```
supervisor@rtbrick: op> clear bgp peer all
```