



BGP User Guide

Version 23.8.1.2, 06 November 2023

| Registered Address | Support | Sales |
|---------------------------------------------------------------------|--------------------------------------------------------------|----------------------------------------------------------|
| 26, Kingston Terrace, Princeton, New Jersey 08540, United States | | |
| | | +91 80 4850 5445 |
| http://www.rtbrick.com | support@rtbrick.com | sales@rtbrick.com |

©Copyright 2023 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 | 4 |
| 1.1. Supported BGP Standards | 4 |
| 1.2. Supported Platforms | 4 |
| 1.3. Supported BGP Features | 4 |
| 1.3.1. MD5 Authentication | 5 |
| 1.3.2. IPv6 Provider Edge (6PE) | 5 |
| 1.3.3. Policies | 6 |
| 1.3.3.1. The Role of a Routing Policy | 6 |
| 1.3.3.2. Attachment Points | 6 |
| 1.3.3.3. Policy Processing | 6 |
| 1.4. BGP Best Path Selection Algorithm | 7 |
| 1.4.1. BGP Best Path Selection Algorithm | 7 |
| 2. BGP Configuration | 9 |
| 2.1. Configuration Hierarchy | 9 |
| 2.2. Configuration Syntax and Commands | 9 |
| 2.2.1. Daemon Options Configuration | 9 |
| 2.2.2. Instance Configuration | 10 |
| 2.2.2.1. Address Families | 11 |
| 2.2.2.2. TCP Authentication Configuration | 13 |
| 2.2.3. BGP Instance Configuration | 14 |
| 2.2.4. BGP Address Family Configuration | 16 |
| 2.2.5. Peer Group Configuration | 20 |
| 2.2.5.1. Peer Groups | 20 |
| 2.2.5.2. Address Families | 21 |
| 2.2.5.3. Maximum Prefix Limit | 23 |
| 2.2.6. Peer Configuration | 26 |
| 2.3. Sample Configuration | 28 |
| 3. BGP Operational Commands | 34 |
| 3.1. BGP Show Commands | 34 |
| 3.1.1. BGP Summary | 34 |
| 3.1.2. BGP Peer | 35 |
| 3.1.3. BGP Peer Group | 37 |
| 3.1.4. BGP FIB | 38 |
| 3.1.5. BGP RIB-in | 41 |
| 3.1.6. BGP RIB-out | 43 |
| 3.1.7. TCP Connections | 45 |
| 3.1.8. TCP Statistics | 47 |
| 3.2. BGP Clear Commands | 48 |

3.2.1. BGP Peer 48

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 the 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] |



RFC and draft compliance are partial except as specified.

1.2. Supported Platforms

Not all features are necessarily supported on each hardware platform. Refer to the *Platform Guide* for the features and the sub-features that are or are not supported by each platform.

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 next-hop 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
- Hostname/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.

1.3.1. 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 peers must be configured with the same password. If authentication fails, the BGP neighbor relationship is not established.

1.3.2. IPv6 Provider Edge (6PE)

The Provider Edge (6PE) solution enables IPv6 communication over the 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 next-hop information. The edge nodes in the 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 an 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.

1.3.3. Policies

1.3.3.1. The Role of a Routing Policy

Routing Policies are the rules that allow you to control and modify the default behavior 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.

1.3.3.2. 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.

1.3.3.3. 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 instances 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 instances or VRF tables.

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).

1.4. BGP Best Path Selection Algorithm

BGP routers typically receive multiple paths to the same destination. A BGP router forms a neighbor relationship by connecting to its neighbors and exchanging the routes, once the connection is established. The BGP route selection algorithm decides which is the best path to install in the IP routing table and to use for traffic forwarding.

1.4.1. BGP Best Path Selection Algorithm

The algorithm eliminates all routes whose next hop is not reachable. Circular route resolution is considered for route resolution.

The algorithm for determining all the routes that have the same route prefix is as follows:

1. The first route selection is performed based on the lowest route source. Route from the local route source is always preferred over the received route. For example, when there is the same prefix route that is redistributed and received from a neighbor, the local (redistributed route) is always preferred. The locally learned route is preferred over the locally crossed or remote crossed route (in the case of VPN, a route might be learned locally in the VRF. The same prefix might be received from the remote as VPNv4. After importing into the VRF routing table, a locally learned route is preferred over the remote local crossed route).
2. Prefer the path with the highest local preference if the route source is the same. If a path does not have a local preference attribute (for example, it is received from an eBGP peer), then it is considered to have the local preference assigned in the given BGP instance. The `show bgp summary` command shows the local preference assigned in the system. This can be changed using the `set local-preference` value.
3. Prefer the route with the shortest AS path, if no route originated. If there is no `AS_PATH` attribute, then it is assumed to be of length 0. A single `AS_SET` is considered to be a length of 1.
4. Prefer the path with the lowest origin type, if the AS path length is the same as all the paths. The available three values include `IGP`, `EGP` and `Incomplete`. The lowest value is `IGP` and the highest value is `Incomplete`.
5. Prefer the path with the lowest Multi Exit Discriminator (MED), if the original codes are the same. (By default, MED values are only compared when routes are learned from the same AS. This behavior can be changed using the `always-compare-med` command. By default, the `always-compare-med` command is enabled. This command allows the MED values to be compared even if they are learned from different ASs. Routes without MED values are treated as if they have a MED value of 0, which is the lowest and, therefore, always the most preferred value.)

6. Prefer external BGP learned routes over internal BGP routes at this point after comparing the route type (internal BGP and external BGP).
7. Prefer the path whose next hop is resolved through the IGP route with the lowest metric.
8. Prefer the length with a shorter CLUSTER length path. If the CLUSTER attribute is not present, the length is assumed to be 0.
9. Prefer the path from the peer with the lowest router ID. For any path with an originator ID attribute, substitute the originator ID for the router ID during router ID comparison.
10. Prefer the lowest peer IP address as the tie-breaker, if the router-id is the same for both sessions. This is for BGP to make route selections in case of multiple peerings are used between the same routers.
11. If add path is enabled, then the same peer might advertise multiple paths for the same prefix. The path with a lower send path ID is preferred.

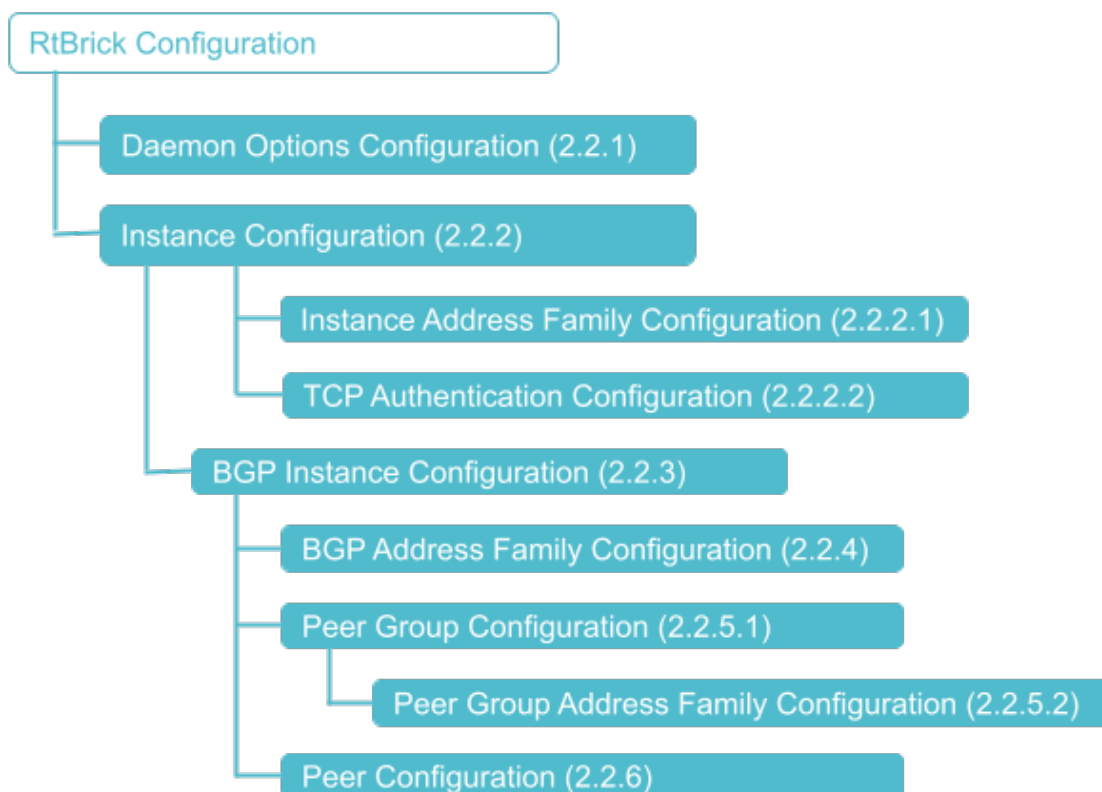
The BGP best path selection algorithm also provides a mechanism to discard paths that are not considered candidates for the best path. The following paths are discarded:

- The paths for which next-hops are not resolved.
- The paths originated from an eBGP neighbor if a local AS is shown in the **AS-PATH** attribute.
- If the BGP **enforce-first-as** attribute is enabled and the update does not contain the AS number of the neighbor as the first AS number in the **AS-SEQUENCE** attribute.
- The paths which are marked as **Received-only**.

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. Daemon Options Configuration

This configuration associates the BGP daemons with routing instances, AFIs, and SAFIs.



The BGP daemon option configurations have been deprecated. These will be removed in a subsequent release.

Syntax:

set daemon-options <instance-name> <attribute> <value>

| Attribute | Description |
|-------------------|----------------------------------------------|
| <instance-name> | Name of the BGP instance |
| <afi> | Address family identifier (AFI). |
| <safi> | Subsequent address family identifier (SAFI). |
| <bd-type> | Daemon type |
| bd-name <bd-name> | Daemon name |

Example: Daemon Configuration

```
{
  "rtbrick-config:daemon-options": [
    {
      "instance-name": "*",
      "afi": "*",
      "safi": "*",
      "bd-type": "bgp.appd",
      "bd-name": "bgp.appd.1"
    },
    {
      "instance-name": "*",
      "afi": "*",
      "safi": "*",
      "bd-type": "bgp.iod",
      "bd-name": "bgp.iod.1"
    }
  ]
}
```


2.2.2. 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.

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> | <p>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>.</p> <div style="display: flex; align-items: center; margin-top: 10px;">  <p>If you want to use the format <as-number:id> with a 4-byte ASN, specify it with an "L". For example, set instance services route-distinguisher 4200000000L:101</p> </div> |
| 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": "198.51.100.41",
    "route-distinguisher": "198.51.100.41:101",
    <...>
  }
}

```

2.2.2.1. 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 |
| route-target (import export) <rt-value> | <p>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.</p> <div style="display: flex; align-items: center; margin-top: 20px;">  <p>If you want to use the format <as-number:id> with a 4-byte ASN, specify it with an "L". For example, set instance services address-family ipv4 unicast route-target export target:4200000000L:14</p> </div> |
| 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:198.51.100.70:14",
          "export": "target:198.51.100.70:14"
        }
      },
      {
        "afi": "ipv6",
        "safi": "unicast",
        "policy": {
          "export": "MY_V6_POLICY"
        },
        "route-target": {
          "import": "target:198.51.100.70:16",
          "export": "target:198.51.100.70:16"
        }
      }
    ],
    <...>
  }
}

```

2.2.2.2. TCP Authentication Configuration

In the instance TCP authentication hierarchy, you can optionally enable MD5 or HMAC SHA authentication. Authentication is not configured for BGP directly but for the TCP sessions used by BGP. It is necessary to bind authentication to a peer in order for the authentication to work.



BGP TCP authentication is not backward compatible.

Syntax:

set instance <instance> **tcp authentication** <authentication-id> <attribute> <value>

| Attribute | Description |
|---------------------|----------------------------------------|
| <authentication-id> | Authentication identifier |
| type <type> | Authentication identifiers such as MD5 |
| key1-id <key1-id> | Key ID1 of the receiver |

| Attribute | Description |
|----------------------------------------------|-------------------------|
| key1-encrypted-text <key1-encrypted-text> | Encrypted text of key1 |
| key1-plain-text <key1-plain-text> | Plain text of key1 |
| key2-id <key2-id> | Key ID2 of the receiver |
| key2-encrypted-text <key2-encrypted-text> | Encrypted text of key2 |
| key2-plain-text <key2-plain-text> | Plain text of key2 |

Example: BGP TCP Authentication Configuration

```
{
  "rtbrick-config:tcp": {
    "authentication": [
      {
        "authentication-id": "auth1",
        "type": "MD5",
        "key1-id": 10,
        "key1-encrypted-text": "$2784cfa7523916c8cc5dfeba83562cbb4",
        "key2-id": 20,
        "key2-encrypted-text": "$2e9bb845e3cfcf8173973029e5c1d90d6"
      }
    ]
  }
}
```

2.2.3. 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 |

| Attribute | Description |
|-----------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 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 associates 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> | Keep a live 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. |
| 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": "198.51.100.51",
    "domain-name": "rtbrick.com",
    "host-name": "spine1",
    "local-as": 4200000100,
    "local-preference": 50,
    "router-id": "198.51.100.51",
    "type-of-service": {
      "precedence": "network-control"
    },
    "protocol-preference": {
      "internal": 180,
      "external": 20
    },
    "timer": {
      "hold-time": 30,
      "keepalive": 10
    },
    <...>
  }
}

```

2.2.4. 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 features specific to the address family.

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 |

| Attribute | Description |
|----------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 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 next-hop |
| resolve-nexthop safi <safi> | Sub-address family to resolve the next-hop |
| 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@spine1: 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"
          }
        ]
      }
    ]
  }
}
```

2.2.5. Peer Group Configuration

2.2.5.1. 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 sources, 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 allowing 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 |

2.2.5.2. Address Families

At this configuration hierarchy, you can enable the address families that shall be supported for the group peers, and enable features specific to the address family. 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 the group |
| extended-nexthop | Enable extended-next-hop encoding for BGP peer groups to allow the transfer of IPv4 prefixes over an IPv6 connection |

| Attribute | Description |
|----------------------------------------------------------|--------------------------------------------------------------------------|
| 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": "2001:db8:0:19::"
        }
      },
      {
        "afi": "ipv6",
        "safi": "labeled-unicast"
      },
      {
        "afi": "ipv6",
        "safi": "unicast"
      },
      {
        "afi": "ipv6",
        "safi": "vpn-unicast",
        "update-nexthop": {
          "ipv6-address": "2001:db8:0:19::"
        }
      }
    ]
  }
}

```

2.2.5.3. Maximum Prefix Limit

The BGP Maximum Prefix Limit feature enables you to set a limit for the maximum number of prefixes that a BGP router can receive from its peer router. If a BGP router receives prefixes that exceed the defined limit threshold, the BGP session gets reset and the session goes idle for a pre-defined period.

You can define a period of time as idle timeout so that the BGP peering gets re-established automatically after the specified time. If you do not specify the idle timeout, the BGP peering does not get re-established until or unless you execute the `clear bgp neighbor` command.

Before getting into inactive or idle mode, the router sends a notification message to the peer router about the exceeded threshold with the error code and the sub-code.

You can configure prefix limits for a peer group.

Syntax:

```
set instance <instance-name> protocol bgp peer-group <peer-group-name>
address-family <afi> <safi> prefix-limit <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, or vpn-multicast |
| count <count> | Number of maximum prefixes that the peer router is allowed to send. The default value is 0. It means no value is configured for prefix limit. |
| idle-timeout <idle-timeout> | Idle or inactive time after maximum limit is reached (in minutes). The allowed range is 1 - 2400 min. The default is Forever . |

Example: BGP Maximum Prefix Limit Configuration

```
supervisor@leaf1: cfg> set instance default protocol bgp peer-group v4_100_as
address-family ipv4 unicast prefix-limit count idle-timeout 5
{
  "ietf-restconf:data": {
    "rtbrick-config:daemon-options": [
      {
        "instance-name": "*",
        "afi": "*",
        "safi": "*",
        "bd-type": "bgp.appd",
```



```
"bd-name": [
  "bgp.appd.1"
],
{
  "instance-name": "*",
  "afi": "*",
  "safi": "*",
  "bd-type": "bgp.iod",
  "bd-name": [
    "bgp.iod.1"
  ]
},
"rtbrick-config:interface": [
  {
    "name": "ifl-0/0/0",
    "host-if": "S1-1-S2",
    "unit": [
      {
        "unit-id": 0,
        "instance": "default",
        "address": {
          "ipv4": [
            {
              "prefix4": "198.51.100.91/24"
            }
          ]
        }
      }
    ]
  },
  {
    "name": "ifl-0/0/1",
    "host-if": "S1-2-S2",
    "unit": [
      {
        "unit-id": 1,
        "address": {
          "ipv4": [
            {
              "prefix4": "198.51.100.102/24"
            }
          ]
        }
      }
    ]
  },
  {
    "name": "lo-0/0/0",
    "unit": [
      {
        "unit-id": 0,
        "address": {
          "ipv4": [
            {
              "prefix4": "198.51.100.46/24"
            }
          ],
          "ipv6": [
            {

```

```

        "prefix6": "2001:db8:0:27::/32"
      }
    ]
  },
  {
    "name": "lo-0/0/1",
    "unit": [
      {
        "unit-id": 1,
        "address": {
          "ipv4": [
            {
              "prefix4": "198.51.100.111/24"
            }
          ],
          "ipv6": [
            {
              "prefix6": "2001:db8:0:223::/32"
            }
          ]
        }
      }
    ]
  }
],
"rtbrick-config:instance": [
  {
    "name": "default",
    "address-family": [
      {
        "afi": "ipv4",
        "safi": "labeled-unicast"
      },
      {
        "afi": "ipv4",
        "safi": "unicast"
      },
      {
        "afi": "ipv6",
        "safi": "labeled-unicast"
      },
      {
        "afi": "ipv6",
        "safi": "unicast"
      },
      {
        "afi": "mpls",
        "safi": "unicast"
      }
    ]
  },
  {
    "protocol": {
      "bgp": {
        "local-as": 200,
        "router-id": "198.51.100.111",
        "address-family": [
          {
            "afi": "ipv4",
            "safi": "unicast",

```

```

        "redistribute": [
            {
                "source": "direct"
            }
        ]
    },
    ],
    "peer": {
        "ipv4": [
            {
                "peer-address": "198.51.100.92",
                "update-source": "198.51.100.91",
                "peer-group": "v4_100_as"
            }
        ]
    },
    "peer-group": [
        {
            "pg-name": "v4_100_as",
            "local-as": 200,
            "remote-as": 100,
            "address-family": [
                {
                    "afi": "ipv4",
                    "safi": "unicast",
                    "prefix-limit": {
                        "count": 100,
                        "idle-timeout": 5
                    }
                }
            ]
        }
    ]
}

```

2.2.6. 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. Furthermore, it is possible to configure TCP authentication and bind it to a peer.

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>
```

Syntax to configure TCP Authentication for BGP peers:

set instance <instance-name> **protocol bgp peer** (ipv4 | ipv6) <peer-address>
<update-source> **authentication-id** <authentication-id>

| 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 |
| authentication-id <authentication-id> | Authentication identifier |

Example 1: BGP peer specified by IP addresses

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

{
  "rtbrick-config:peer": {
    "ipv4": [
      {
        "peer-address": "198.51.100.82",
        "update-source": "198.51.100.81",
        "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": "ifl-0/0/1/1",
        "peer-group": "spine"
      }
    ]
  }
}
```

Example 3: BGP peer authentication

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

{
  "rtbrick-config:peer": {
    "interface": [
      {
        "name": "ifl-0/0/1/1",
        "authentication-id": "auth1",
        "peer-group": "spine"
      }
    ]
  }
}
```

2.3. Sample Configuration

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

```
{
  "ietf-restconf:data": {
    "rtbrick-config:daemon-options": [
      {
        "instance-name": "*",
        "afi": "*",
        "safi": "*",
        "bd-type": "bgp.appd",
        "bd-name": "bgp.appd.1"
      }
    ],
    "rtbrick-config:instance": [
      {
        "name": "default",
        "ipv4-router-id": "198.51.100.51",
        "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": "ifl-0/1/1/1",
        "authentication-id": "auth1",
        "peer-group": "spine"
      },
      {
        "name": "ifl-0/2/1/1",
        "peer-group": "leaf1"
      },
      {
        "name": "ifl-0/2/2/1",
        "peer-group": "leaf2"
      }
    ]
  },
  "peer-group": [
    {
      "pg-name": "leaf1",
      "link-local-next-hop-only": "true",
      "remote-as": 420000201,
      "address-family": [
        {
          "afi": "ipv4",
          "safi": "vpn-unicast",
          "extended-next-hop": "true",
          "next-hop-unchanged": "true"
        },
        {
          "afi": "ipv6",
          "safi": "labeled-unicast"
        },
        {
          "afi": "ipv6",
          "safi": "unicast"
        },
        {
          "afi": "ipv6",
          "safi": "vpn-unicast",
          "next-hop-unchanged": "true"
        }
      ]
    },
    {
      "pg-name": "leaf2",
      "link-local-next-hop-only": "true",

```



```

{
  "ietf-restconf:data": {
    "rtbrick-config:instance": [
      {
        "name": "default",
        "ipv4-router-id": "198.51.100.53",
        "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": "ifl-0/1/1/1"
                  "authentication-
id": "auth1",
                  "peer-group": "spine"
                },
                {
                  "name": "ifl-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": "2001:db8:0:19::"
    }
  },
  {
    "afi": "ipv6",
    "safi": "labeled-unicast"
  },
  {
    "afi": "ipv6",
    "safi": "unicast"
  },
  {
    "afi": "ipv6",
    "safi": "vpn-unicast",
    "update-nexthop": {
      "ipv6-address": "2001:db8:0:19::"
    }
  }
]
}
}
},
{
  "name": "services",
  "ipv4-router-id": "198.51.100.41",
  "route-distinguisher": "198.51.100.41:101",
  "address-family": [
    {
      "afi": "ipv4",
      "safi": "unicast",
      "policy": {
        "export": "MY_V4_POLICY"
      },
      "route-target": {
        "import": "target:198.51.100.70:14",
        "export": "target:198.51.100.70:14"
      }
    },
    {
      "afi": "ipv6",
      "safi": "unicast",
      "policy": {
        "export": "MY_V6_POLICY"
      },
      "route-target": {
        "import": "target:198.51.100.70:16",
        "export": "target:198.51.100.70:16"
      }
    }
  ],
  "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: 198.51.100.102, Cluster ID: 198.51.100.102
  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 display information on BGP peers.

Syntax:

show bgp peer <option> ...

| Option | Description |
|-----------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------|
| - | Without any option, the commands display 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. |
| 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. |



Although 6PE routes are labeled, they are handled as unicast routes, and therefore will be shown as IPv6 unicast in the BGP peer statistics.

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

```

Example 2: BGP Peer Detail View

```

supervisor@rtbrick: op> show bgp peer detail
Peer: PE2, Peer IP: 198.51.100.39, Remote AS: 2000, Local: 198.51.100.29, 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      : 198.51.100.106, Local ID   : 198.51.100.102
  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

```

3.1.3. BGP Peer Group

The 'show bgp peer-group' commands display 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 display 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

```

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 display 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'. |

| Option | Description |
|------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <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', 'vpn-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. |
| <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, all SAFIs and all instances


```

supervisor@rtbrick: op> show bgp fib ipv6
Instance: default, AFI: ipv6, SAFI: unicast
  Prefix                               Preference   Out Label
Next Hop
  2001:db8:0:2::/32                    20          -
198.51.100.39
  2001:db8:0:2::/32                    20          -
198.51.100.39
Instance: services, AFI: ipv6, SAFI: unicast
  Prefix                               Preference   Out Label
Next Hop
  2001:db8:0:6::/32                    200         -
2001:db8:0:4::
Instance: default, AFI: ipv6, SAFI: labeled-unicast
  Prefix                               Preference   Out Label
Next Hop
  2001:db8:0:2::/32                    20          2003
198.51.100.39
  2001:db8:0:2::/32                    20          2003
198.51.100.39
Instance: default, AFI: ipv6, SAFI: vpn-unicast
  Prefix                               Preference   Out Label
Next Hop
  2001:db8:0:5::/32                    200         20003,bos:1
  2001:db8:0:6::/32                    200         20003,bos:1
  2001:db8:0:8::/32                    200         20003,bos:1
  2001:db8:0:9::/32                    20          20006,bos:1
2001:db8:0:7::
  2001:db8:0:10::/32                   20          20006,bos:1
2001:db8:0:7::
  2001:db8:0:11::/32                   20          20006,bos:1
2001:db8:0:7::
  2001:db8:0:12::/32                   20          20006,bos:1
2001:db8:0:7::

```

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: 2001:db8:0:5::/32
    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:198.51.100.93:2']
    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: 2001:db8:0:6::/32
    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:198.51.100.93:2']
    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 on 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
  198.51.100.75/24      198.51.100.93          -        100                -
  198.51.100.76/24      198.51.100.94          -        100                -
  198.51.100.77/24      198.51.100.99          -        100                -
  198.51.100.78/24      198.51.100.94          -        100                -
  198.51.100.79/24      198.51.100.99          -        100                -
  198.51.100.82/24      198.51.100.94          -        100                -
  198.51.100.93/24      198.51.100.93          -        100                -
  198.51.100.94/24      198.51.100.94          -        100                -
  198.51.100.99/24      198.51.100.99          -        100                -
  198.51.100.99/24      198.51.100.99          -        100                -
Instance: default, AFI: ipv4, SAFI: vpn-unicast
Peer: None, Received routes: 4
  Prefix                Next Hop                MED      Local Preference  AS
Path
  198.51.100.14/24      2001:db8:0:1::/32      0        -                  -
4200000004
  198.51.100.17/24      2001:db8:0:1::/32      0        -                  -
4200000004
  198.51.100.16/24      2001:db8:0:1::/32      0        -                  -
4200000004

```

Example 2: Summary view of the BGP rib-in for IPv4, 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
  198.51.100.75/24      198.51.100.93          -        100               -
  198.51.100.76/24      198.51.100.94          -        100               -
  198.51.100.77/24      198.51.100.95          -        100               -
  198.51.100.78/24      198.51.100.94          -        100               -
  198.51.100.79/24      198.51.100.95          -        100               -
  198.51.100.82/24      198.51.100.94          -        100               -
  198.51.100.93/24      198.51.100.93          -        100               -
  198.51.100.94/24      198.51.100.94          -        100               -
  198.51.100.95/24      198.51.100.95          -        100               -
  198.51.100.99/24      198.51.100.95          -        100               -
Instance: default, AFI: ipv4, SAFI: vpn-unicast
Peer: None, Received routes: 4
  Prefix                Next Hop                MED      Local Preference  AS
Path
  198.51.100.14/24      2001:db8:0:13::        0        -                 -
4200000004
  198.51.100.17/24      2001:db8:0:13::        0        -                 -
4200000004

```

Example 3: Summary view of the received routes

```

supervisor@rtbrick: op> show bgp rib-in ipv4 unicast peer address 198.51.100.94
Instance: ip2vrf, AFI: ipv4, SAFI: unicast
Peer: None, Received routes: 13
  Prefix                Next Hop                MED      Local Preference  AS
Path
  198.51.100.75/24      198.51.100.93          -        100               -
  198.51.100.76/24      198.51.100.94          -        100               -
  198.51.100.77/24      198.51.100.95          -        100               -
  198.51.100.78/24      198.51.100.94          -        100               -
  198.51.100.79/24      198.51.100.95          -        100               -
  198.51.100.82/24      198.51.100.94          -        100               -
  198.51.100.93/24      198.51.100.93          -        100               -
  198.51.100.113/24     198.51.100.93          -        100               -
4200000003
  198.51.100.114/24     198.51.100.93          -        100               -
4200000004
  198.51.100.94/24      198.51.100.94          -        100               -
  198.51.100.95/24      198.51.100.95          -        100               -
  198.51.100.99/24      198.51.100.95          -        100               -

```

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'. |

| Option | Description |
|------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <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. |
| <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-out ipv4 unicast peer CE1-Vrf1
Instance: vrf1, AFI: ipv4, SAFI: unicast
Peer: CE1-Vrf1, Sent routes: 4
  Prefix                MED          Local Preference  Origin      Next Hop
AS Path
  198.51.100.104/24      0            -                 Incomplete -
65001
  198.51.100.113/24      0            -                 Incomplete -
65001
  198.51.100.117/24      0            -                 Incomplete -
65001
  198.51.100.106/24      0            -                 Incomplete -
65001

```

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

```

supervisor@rtbrick: op> show bgp rib-out
Instance: vrf1, AFI: ipv4, SAFI: unicast
  Peer-group: pe1_cel, Sent routes: 4
    Prefix                                MED                                Local Preference  Origin
Next Hop                                AS Path
  198.51.100.104/24                       0                                  -
Incomplete -                               65001
  198.51.100.113/24                       0                                  -
Incomplete -                               65001
  198.51.100.105/24                       0                                  -
Incomplete -                               65001
  198.51.100.106/24                       0                                  -
Incomplete -                               65001
Instance: vrf1, AFI: ipv6, SAFI: unicast
  Peer-group: pe1_cel, Sent routes: 3
    Prefix                                MED                                Local Preference  Origin
Next Hop                                AS Path
  2001:db8:0:14::/24                      0                                  -
Incomplete -                               65001
  2001:db8:0:15::/24                      0                                  -
Incomplete -                               65001
  2001:db8:0:16::/24                      0                                  -
Incomplete -                               65001

```

3.1.7. TCP Connections

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

Syntax:

show bgp tcp bgp.iod.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@leaf1: cfg> show bgp tcp bgp.iod.1 connection
Instance      Local IP Address      Remote IP Address      Local
Port Remote Port State
default      2001:db8:0:189::    2001:db8:0:38::      179
49568        Established
default      2001:db8:0:61::     2001:db8:0:237::    50529
179          Established

```

Example 2: Detailed information of the BGP TCP connections

```
supervisor@leaf1: cfg> show bgp tcp bgp.iod.1 connection detail
Instance: default
  Local IPv6 address      : 2001:db8:0:189::
  Remote IPv6 address    : 2001:db8:0:38::
  Local port              : 179
  Remote port            : 49568
  State                   : Established
    Internal
      Options              : -- | Keepalive | --
      TOS                  : 0
      TTL                  : 1
      Priority              : 1
      Flags                : -|-|-|-|Nagle Disabled|-|Wnd Scale|-|-|-
      Last trigger         : 27
      Timer                : 37624
    Timers
      Poll                 : 0s
      Poll interval       : 0s
      Retransmission      : 65535s
    Receiver
      Expected sequence   : 32965979
      Available window    : 96816
      Announced window   : 95562
      Announced wnd RT edge : 33061541
      MSS                  : 1440
      RTT estimate        : 0
    Timeout
      Sequence            : 17683639s
      Retransmission      : 3s
      Retransmissions     : 0s
      Duplicate acks      : 0s
      Highest ack'd sequence : 17683658s
    Congestion
      Window              : 162834
      Persist count       : 0
      Send scale          : 5
      Receive scale       : 5
    Sender
      Next seq to send    : 17683658
      Last wnd update seq : 32965979
      Last wnd update ack : 17683658
      Window              : 96192
      Max window announced : 96800
      Acknowledged        : None
      Send buf            : 56476
      Send queue length   : 0
      Unsent oversize     : 0
      TS last ack sent    : 818020352
    Keepalive
      Next keepalive idle : 7200000
      Keepalive interval  : 75000
      Keepalive count     : 9
      Keep sent count     : 0
    Authentication
      Auth type           : HMAC-SHA-256-128
      key1-id             : 255
      key2-id             : 0
      Algorithm mismatch  : 0
```

```

Secret mismatch      : 43
Latest sent digest   : 850fea02c98912ce4497ec2b101a4f7c
Latest received digest : a718fb88e0d7fd4a00843e6aec03c864

```

3.1.8. TCP Statistics

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

Syntax:

show bgp tcp bgp.iod.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. |
| 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 addresses 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 a given instance, peer-ip and address-family. |
| instance <instance> all soft-out <afi> <safi> | Sends route update to all peers for given instance and addresses family. |

| Option | Description |
|-----------------------------------------------------------|---------------------------------------------------------------------------------------|
| instance <instance> <peer-ip> source <src-ip> stats | Clears the statistics of a specific peer for a given instance, peer-ip and source-IP. |
| instance <instance> <peer-ip> stats | Clears the statistics of the peer for a given instance and peer-IP. |
| instance <instance> all stats | Clears the statistics of all peers for a given instance. |

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

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