JUNIPER NETWORKS

Network Configuration Example

Conditional Installation of Prefixes in a Routing Table

Release
12.3

Published: 2012-11-14

Juniper Networks, Inc.
1194 North Mathilda Avenue
Sunnyvale, California 94089
USA
408-745-2000
www.juniper.net

Network Configuration Example Conditional Installation of Prefixes in a Routing Table
Release 12.3

The information in this document is current as of the date on the title page.

YEAR 2000 NOTICE

Juniper Networks hardware and software products are Year 2000 compliant. Junos OS has no known time-related limitations through the year 2038. However, the NTP application is known to have some difficulty in the year 2036.

END USER LICENSE AGREEMENT

The Juniper Networks product that is the subject of this technical documentation consists of (or is intended for use with) Juniper Networks software. Use of such software is subject to the terms and conditions of the End User License Agreement ("EULA") posted at http://www.juniper.net/support/eula.html. By downloading, installing or using such software, you agree to the terms and conditions of that EULA.

Table of Contents

Introduction

This document describes the advantages of and uses for installing prefixes in a routing table based on a defined condition. It also provides a step-by-step procedure for configuring and verifying the conditional installation of prefixes.

Conditional Installation of Prefixes Use Cases

Networks are usually subdivided into smaller, more-manageable units called autonomous systems (ASs). When BGP is used by routers to form peer relationships in the same AS, it is referred to as internal BGP (IBGP). When BGP is used by routers to form peer relationships in different ASs, it is referred to as external BGP (EBGP).

After performing route sanity checks, a BGP router accepts the routes received from its peers and installs them into the routing table. By default, all routers in IBGP and EBGP sessions follow the standard BGP advertisement rules. While a router in an IBGP session advertises only the routes learned from its direct peers, a router in an EBGP session advertises all routes learned from its direct and indirect peers (peers of peers). Hence, in a typical network configured with EBGP, a router adds all routes received from an EBGP peer into its routing table and advertises nearly all routes to all EBGP peers.

A service provider exchanging BGP routes with both customers and peers on the Internet is at risk of malicious and unintended threats that can compromise the proper routing of traffic, as well as the operation of the routers.

This has several disadvantages:

- **Non-aggregated route advertisements**—A customer could erroneously advertise all its prefixes to the ISP rather than an aggregate of its address space. Given the size of the Internet routing table, this must be carefully controlled. An edge router might also need only a default route out toward the Internet and instead be receiving the entire BGP routing table from its upstream peer.
- **BGP route manipulation**—If a malicious administrator alters the contents of the BGP routing table, it could prevent traffic from reaching its intended destination.
- **BGP route hijacking**—A rogue administrator of a BGP peer could maliciously announce a network's prefixes in an attempt to reroute the traffic intended for the victim network to the administrator's network to either gain access to the contents of traffic or to block the victim's online services.
- **BGP denial of service (DoS)**—If a malicious administrator sends unexpected or undesirable BGP traffic to a router in an attempt to use all of the router's available BGP resources, it might result in impairing the router's ability to process valid BGP route information.

Conditional installation of prefixes can be used to address all the problems previously mentioned. If a customer requires access to remote networks, it is possible to install a specific route in the routing table of the router that is connected with the remote network. This does not happen in a typical EBGP network and hence, conditional installation of prefixes becomes extremely essential.

ASs are not only bound by physical relationships but by business or other organizational relationships. An AS can provide services to another organization, or act as a transit AS between two other ASs. These transit ASs are bound by contractual agreements between the parties that include parameters on how to connect to each other and most importantly, the type and quantity of traffic they carry for each other. Therefore, for both legal and financial reasons, service providers must implement policies that control how BGP routes are exchanged with neighbors, which routes are accepted from those neighbors, and how those routes affect the traffic between the ASs.

There are many different options available to filter routes received from a BGP peer to both enforce inter-AS policies and mitigate the risks of receiving potentially harmful routes. Conventional route filtering examines the attributes of a route and accepts or rejects the route based on such attributes. A policy or filter can examine the contents of the AS-Path, the next-hop value, a community value, a list of prefixes, the address family of the route, and so on.

In some cases, the standard "acceptance condition" of matching a particular attribute value is not enough. The service provider might need to use another condition outside of the route itself, for example, another route in the routing table. As an example, it might be desirable to install a default route received from an upstream peer, only if it can be verified that this peer has reachability to other networks further upstream. This conditional route installation avoids installing a default route that is used to send traffic toward this peer, when the peer might have lost its routes upstream, leading to black-holed traffic. To achieve this, the customer edge router can be configured to search for the presence of a particular route in the routing table, and based on this knowledge accept or reject another prefix.

"Example: Configuring Conditional Installation of Prefixes in a Routing Table" on page 5 explains how the conditional installation of prefixes can be configured and verified.

Related Documentation

- Understanding Conditional Installation of Prefixes in a Routing Table on page 3
- Example: Configuring Conditional Installation of Prefixes in a Routing Table on page 5

Understanding Conditional Installation of Prefixes in a Routing Table

BGP accepts all non-looped routes learned from neighbors and imports them into the RIB-In table. If these routes are accepted by the BGP import policy, they are then imported into the inet.0 routing table. In cases where only certain routes are required to be imported, provisions can be made such that the import takes place based on a condition or a set of conditions.

The condition for importing a route can be based on:

- The peer the route was learned from
- The interface the route was learned on
- Some other required attribute

This is known as conditional installation of prefixes and is described in "Example: Configuring Conditional Installation of Prefixes in a Routing Table" on page 5.

The Juniper Networks® Junos® Operating System (Junos OS) supports conditional import or export of routes based on the existence of another route in the routing table. It is also important to understand that in Junos OS, although an import policy (inbound route filter) might reject a route, not use it for traffic forwarding, and not include it in an advertisement to other peers, the router retains these routes as hidden routes. These hidden routes are not available for policy or routing purposes. However, they do occupy memory space on the router. A service provider filtering routes to control the amount of information being kept in memory and processed by a router might want the router to entirely drop the routes being rejected by the import policy.

The rules of BGP route retention are as follows:

- By default, all routes learned from BGP are retained, except those where the AS path is looped. (The AS path includes the local AS.)
- By configuring the **keep all** statement, all routes learned from BGP are retained, even those with the local AS in the AS path.
- By configuring the **keep none** statement, all routes received are discarded. When this statement is configured and the inbound policy changes, Junos OS re-advertises all the routes advertised by the peer.

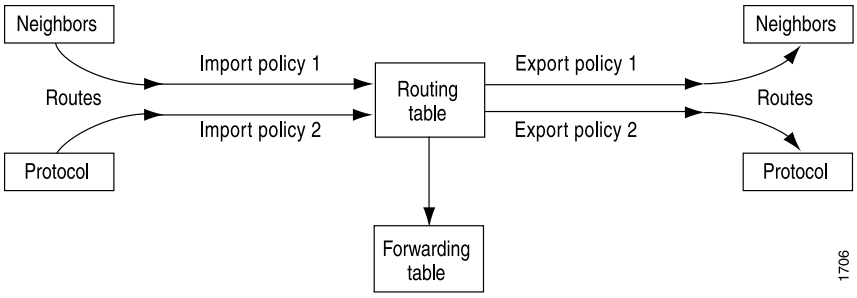
To enable conditional installation of prefixes, an import policy must be configured on the device where the prefix import has to take place. The import policy evaluates each received route to verify that it satisfies all the match conditions under the **from** statement. It also searches for the existence of the route defined under the **condition** statement (also configured under the **from** statement).

If the received route does not match the entire set of required conditions defined in the policy, or if the route defined under the **condition** statement does not exist in the routing table, the received route is not installed in the table, and becomes a hidden route (by default). Thus, a conditional import policy matches the received routes for the desired route or prefix you want installed in the routing table.

Hidden routes can be viewed by using the **show route receive-protocol bgp *neighbor-address* hidden** command. The hidden routes can then be retained or dropped from the routing table by configuring the **keep all | none** statement at the **[edit protocols bgp]** or **[edit protocols bgp group *group-name*]** hierarchy level.

Figure 1 on page 4 illustrates where BGP import and export policies are applied. An import policy is applied to inbound routes that are visible in the output of the **show route receive-protocol bgp *neighbor-address*** command. An export policy is applied to outbound routes that are visible in the output of the **show route advertising-protocol bgp *neighbor-address*** command.

Figure 1: BGP Import and Export Policies



To configure the conditional installation of prefixes with the help of an import policy:

1. Create a **condition** statement to check prefixes.

```
[edit]
policy-options {
  condition condition-name {
    if-route-exists address table table-name;
  }
}
```

2. Create an import policy with the newly created condition using the **condition** statement.

```
[edit]
policy-options {
  policy-statement policy-name {
    term 1 {
      from {
        protocols bgp;
        condition condition-name;
      }
      then {
        accept;
      }
    }
  }
}
```

3. Apply the import policy to the device that requires only selected prefixes to be imported into the routing table.

```
[edit]
protocols bgp {
```

```
    group group-name {
        import policy-name;
    }
}
```

Related Documentation

- Conditional Installation of Prefixes Use Cases on page 1
- Example: Configuring Conditional Installation of Prefixes in a Routing Table on page 5

Example: Configuring Conditional Installation of Prefixes in a Routing Table

This example shows how to configure conditional installation of prefixes in a routing table using BGP import policy.

Requirements

This example uses the following hardware and software components:

- M Series Multiservice Edge Routers, MX Series 3D Universal Edge Routers, or T Series Core Routers
- Junos OS Release 9.0 or later

Overview

The following example shows how a router sorts through the received BGP routes with the help of an import policy, and selects the preferred routes to be installed in the routing table.

Topology

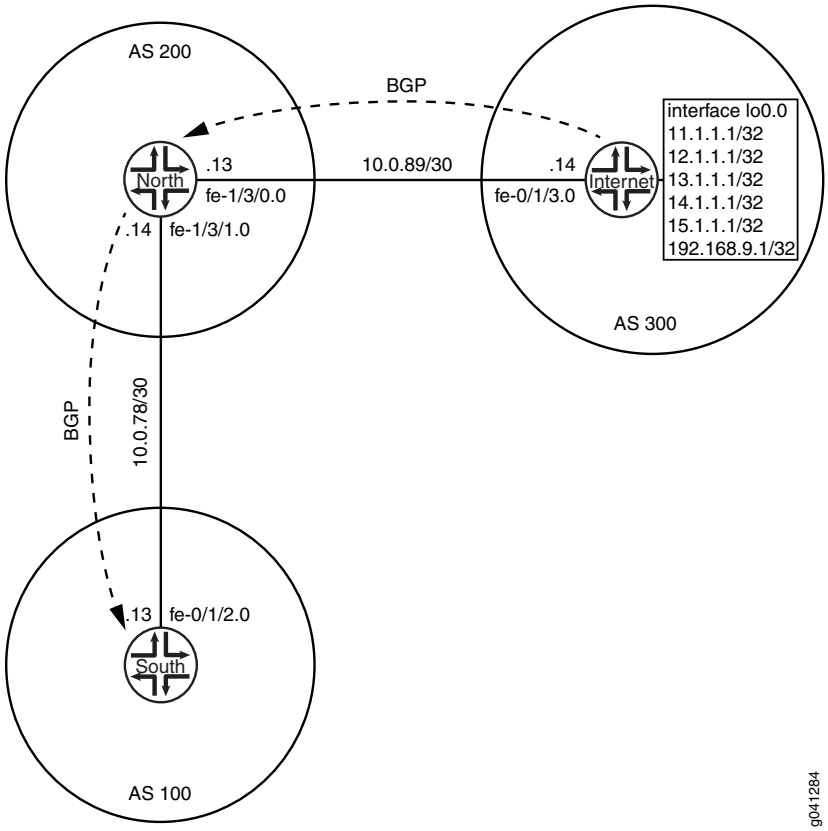
In this example, three routers in three different autonomous systems are connected and configured with the BGP protocol. Router Internet, which is the upstream router, has five addresses configured on its lo0.0 loopback interface (11.1.1.1/32, 12.1.1.1/32, 13.1.1.1, 14.1.1.1/32, and 15.1.1.1/32), and an extra loopback address (192.168.9.1/32) to be configured as the router ID. These six addresses are exported into BGP to emulate the contents of a BGP routing table of a router connected to the Internet, and advertised to Router North.

Router North exports a default route into BGP, and advertises the default route and the six BGP routes to Router South, which is the downstream router. Router South receives all seven routes, searches for the required route (in this example, 11.1.1.1/32), and installs this route and the default route in its routing table.

NOTE: In this example, the sixth loopback address, 192.168.9.1/32, is configured as the router ID on Router Internet. You can also configure any of the first five loopback addresses or a non-loopback address as the router ID.

Figure 2 on page 6 shows the topology used in this example.

Figure 2: Conditional Installation of Prefixes



Configuration

CLI Quick Configuration

To quickly configure this example, copy the following commands, paste them into a text file, remove any line breaks, change any details necessary to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

Router Internet

```
set interfaces lo0 unit 0 family inet address 11.1.1.1/32
set interfaces lo0 unit 0 family inet address 12.1.1.1/32
set interfaces lo0 unit 0 family inet address 13.1.1.1/32
set interfaces lo0 unit 0 family inet address 14.1.1.1/32
set interfaces lo0 unit 0 family inet address 15.1.1.1/32
set interfaces lo0 unit 0 family inet address 192.168.9.1/32
set interfaces fe-0/1/3 unit 0 family inet address 10.0.89.14/30
set protocols bgp group toNorth local-address 10.0.89.14
set protocols bgp group toNorth peer-as 200
```

```
set protocols bgp group toNorth neighbor 10.0.89.13
set protocols bgp group toNorth export into-bgp
set policy-options policy-statement into-bgp term 1 from interface lo0.0
set policy-options policy-statement into-bgp term 1 then accept
set routing-options router-id 192.168.9.1
set routing-options autonomous-system 300
```

Router North

```
set interfaces fe-1/3/1 unit 0 family inet address 10.0.78.14/30
set interfaces fe-1/3/0 unit 0 family inet address 10.0.89.13/30
set interfaces lo0 unit 0 family inet address 192.168.8.1/32
set protocols bgp group toInternet local-address 10.0.89.13
set protocols bgp group toInternet peer-as 300
set protocols bgp group toInternet neighbor 10.0.89.14
set protocols bgp group toSouth local-address 10.0.78.14
set protocols bgp group toSouth peer-as 100
set protocols bgp group toSouth neighbor 10.0.78.13
set protocols bgp group toSouth export default_route
set policy-options policy-statement default_route term 1 from route-filter 0/0 exact
set policy-options policy-statement default_route term 1 then accept
set routing-options static route 0/0 reject
set routing-options router-id 192.168.8.1
set routing-options autonomous-system 200
```

Router South

```
set interfaces fe-0/1/2 unit 0 family inet address 10.0.78.13/30
set interfaces lo0 unit 0 family inet address 192.168.7.1/32
set policy-options condition prefix_11 if-route-exists 11.1.1.1/32
set policy-options condition prefix_11 if-route-exists table inet.0
set policy-options policy-statement conditional-import-bgp term conditional-default
  from route-filter 0.0.0.0/0 exact
set policy-options policy-statement conditional-import-bgp term conditional-default
  from condition prefix_11
set policy-options policy-statement conditional-import-bgp term conditional-default
  then accept
set policy-options policy-statement conditional-import-bgp term prefix_11 from route-filter
  11.0.0.0/8 orlonger
set policy-options policy-statement conditional-import-bgp term prefix_11 then accept
set policy-options policy-statement conditional-import-bgp term others then reject
set protocols bgp group toNorth local-address 10.0.78.13
set protocols bgp group toNorth peer-as 200
set protocols bgp group toNorth neighbor 10.0.78.14
set protocols bgp group toNorth import conditional-import-bgp
set routing-options router-id 192.168.7.1
set routing-options autonomous-system 100
```

Configuring Conditional Installation of Prefixes

Step-by-Step Procedure

The following example requires that you navigate various levels in the configuration hierarchy. For information about navigating the CLI, see Using the CLI Editor in Configuration Mode in the Junos OS CLI User Guide.

To configure conditional installation of prefixes:

1. Configure the router interfaces forming the links between the three routers.

   Router Internet

```
[edit interfaces]
user@Internet# set fe-0/1/3 unit 0 family inet address 10.0.89.14/30
```

```
Router North
[edit interfaces]
user@North# set fe-1/3/1 unit 0 family inet address 10.0.78.14/30
user@North# set fe-1/3/0 unit 0 family inet address 10.0.89.13/30
```

```
Router South
[edit interfaces]
user@South# set fe-0/1/2 unit 0 family inet address 10.0.78.13/30
```

2. Configure five loopback interface addresses on Router Internet to emulate BGP routes learned from the Internet that are to be imported into the routing table of Router South, and configure an additional address (192.168.9.1/32) that will be configured as the router ID.

```
Router Internet
[edit interfaces lo0 unit 0 family inet]
user@Internet# set address 11.1.1.1/32
user@Internet# set address 12.1.1.1/32
user@Internet# set address 13.1.1.1/32
user@Internet# set address 14.1.1.1/32
user@Internet# set address 15.1.1.1/32
user@Internet# set address 192.168.9.1/32
```

   Also, configure the loopback interface addresses on Routers North and South.

```
Router North
[edit interfaces lo0 unit 0 family inet]
user@North# set address 192.168.8.1/32
```

```
Router South
[edit interfaces lo0 unit 0 family inet]
user@North# set address 192.168.7.1/32
```

3. Configure the static default route on Router North to be advertised to Router South.

```
[edit routing-options]
user@North# set static route 0/0 reject
```

4. Define the condition for importing prefixes into the routing table on Router South.

```
[edit policy-options condition prefix_11 if-route-exists]
user@South# set 11.1.1.1/32
user@South# set table inet.0
```

5. Define export policies (**into-bgp** and **default_route**) on Routers Internet and North respectively, to advertise routes to BGP.

```
Router Internet
[edit policy-options policy-statement into-bgp ]
user@Internet# set term 1 from interface lo0.0
user@Internet# set term 1 then accept
```

```
Router North
[edit policy-options policy-statement default_route]
user@North# set term 1 from route-filter 0/0 exact
user@North# set term 1 then accept
```

6. Define an import policy (**conditional-import-bgp**) on Router South to import the routes advertised by Router North into its routing table.

NOTE: Ensure that you reference the condition, prefix_11 (configured in Step 4), in the import policy.

```
[edit policy-options policy-statement conditional-import-bgp]
user@South# set term conditional-default from route-filter 0.0.0.0/0 exact
user@South# set term conditional-default from condition prefix_11
user@South# set term conditional-default then accept
user@South# set term prefix_11 from route-filter 11.0.0.0/8 orlonger
user@South# set term prefix_11 then accept
user@South# set term others then reject
```

7. Configure BGP on all three routers to enable the flow of prefixes between the autonomous systems.

NOTE: Ensure that you apply the defined import and export policies to the respective BGP groups for prefix advertisement to take place.

```
Router Internet
[edit protocols bgp group toNorth]
user@Internet# set local-address 10.0.89.14
user@Internet# set peer-as 200
user@Internet# set neighbor 10.0.89.13
user@Internet# set export into-bgp

Router North
[edit protocols bgp group toInternet]
user@North# set local-address 10.0.89.13
user@North# set peer-as 300
user@North# set local-as 200
user@North# set neighbor 10.0.89.14

[edit protocols bgp group toSouth]
user@North# set local-address 10.0.78.14
user@North# set peer-as 100
user@North# set neighbor 10.0.78.13
user@North# set export default_route

Router South
[edit protocols bgp group toNorth]
user@South# set local-address 10.0.78.13
user@South# set peer-as 200
user@South# set neighbor 10.0.78.14
user@South# set import conditional-import-bgp
```

8. Configure the router ID and autonomous system number for all three routers.

NOTE: In this example, the router ID is configured based on the IP address configured on the lo0.0 interface of the router.

Router Internet
[edit routing options]
user@Internet# **set router-id 192.168.9.1**
user@Internet# **set autonomous-system 300**

Router North
[edit routing options]
user@North# **set router-id 192.168.8.1**
user@North# **set autonomous-system 200**

Router South
[edit routing options]
user@South# **set router-id 192.168.7.1**
user@South# **set autonomous-system 100**

From configuration mode, confirm your configuration by entering the **show interfaces** *interface-name*, **show protocols bgp** , **show policy-options**, and **show routing-options** commands.

If the output does not display the intended configuration, repeat the instructions in this example to correct the configuration.

```
user@South# show interfaces fe-0/1/2
unit 0 {
    family inet {
        address 10.0.78.13/30;
    }
}

user@South# show protocols bgp
group toNorth {
    local-address 10.0.78.13;
    import conditional-import-bgp;
    peer-as 200;
    neighbor 10.0.78.14;
}

user@South# show policy-options
policy-statement conditional-import-bgp {
    term conditional-default {
        from {
            route-filter 0.0.0.0/0 exact;
            condition prefix_11;
        }
        then accept;
    }
    term prefix_11 {
        from {
            route-filter 11.0.0.0/8 orlonger;
        }
        then accept;
    }
```

```
    term others {
        then reject;
    }
}
condition prefix_11 {
    if-route-exists {
        11.1.1.1/32;
        table inet.0;
    }
}
```

```
user@South# show routing-options
router-id 192.168.7.1;
autonomous-system 100;
```

Run these commands on Routers Internet and North to confirm the configurations. If you are done configuring the routers, enter **commit** from configuration mode.

Verification

Confirm that the configuration is working properly.

Verifying BGP

Purpose Verify that BGP sessions have been established between the three routers.

Action From operational mode, run the **show bgp neighbor *neighbor-address*** command.

1. Check the BGP session on Router Internet to verify that Router North is a neighbor.

```
user@Internet> show bgp neighbor 10.0.89.13
Peer: 10.0.89.13+179 AS 200  Local: 10.0.89.14+56187 AS 300
  Type: External    State: Established    Flags: [ImportEval Sync]
  Last State: OpenConfirm   Last Event: RecvKeepAlive
  Last Error: None
  Export: [ into-bgp ]
  Options: [Preference LocalAddress PeerAS Refresh]
  Local Address: 10.0.89.14 Holdtime: 90 Preference: 170
  Number of flaps: 0
  Peer ID: 192.168.8.1     Local ID: 192.168.9.1       Active Holdtime: 90
  Keepalive Interval: 30         Group index: 0    Peer index: 0
  BFD: disabled, down
  Local Interface: fe-0/1/3.0
  NLRI for restart configured on peer: inet-unicast
  NLRI advertised by peer: inet-unicast
  NLRI for this session: inet-unicast
  Peer supports Refresh capability (2)
  Stale routes from peer are kept for: 300
  Peer does not support Restarter functionality
  NLRI that restart is negotiated for: inet-unicast
```

```
  NLRI of received end-of-rib markers: inet-unicast
  NLRI of all end-of-rib markers sent: inet-unicast
  Peer supports 4 byte AS extension (peer-as 200)
  Peer does not support Addpath
  Table inet.0 Bit: 10000
    RIB State: BGP restart is complete
    Send state: in sync
    Active prefixes:              0
    Received prefixes:            0
    Accepted prefixes:            0
    Suppressed due to damping:    0
    Advertised prefixes:          6
  Last traffic (seconds): Received 9    Sent 18   Checked 28
  Input messages:  Total 12     Updates 1       Refreshes 0     Octets 232
  Output messages: Total 14     Updates 1       Refreshes 0     Octets 383
  Output Queue[0]: 0
```

2. Check the BGP session on Router North to verify that Router Internet is a neighbor.

```
user@North> show bgp neighbor 10.0.89.14
Peer: 10.0.89.14+56187 AS 300  Local: 10.0.89.13+179 AS 200
  Type: External    State: Established    Flags: [ImportEval Sync]
  Last State: OpenConfirm   Last Event: RecvKeepAlive
  Last Error: None
  Options: [Preference LocalAddress PeerAS Refresh]
  Local Address: 10.0.89.13 Holdtime: 90 Preference: 170
  Number of flaps: 0
  Peer ID: 192.168.9.1     Local ID: 192.168.8.1       Active Holdtime: 90
  Keepalive Interval: 30         Group index: 0    Peer index: 0
  BFD: disabled, down
  Local Interface: fe-1/3/0.0
  NLRI for restart configured on peer: inet-unicast
  NLRI advertised by peer: inet-unicast
  NLRI for this session: inet-unicast
  Peer supports Refresh capability (2)
  Stale routes from peer are kept for: 300
  Peer does not support Restarter functionality
  NLRI that restart is negotiated for: inet-unicast
  NLRI of received end-of-rib markers: inet-unicast
  NLRI of all end-of-rib markers sent: inet-unicast
  Peer supports 4 byte AS extension (peer-as 300)
  Peer does not support Addpath
  Table inet.0 Bit: 10001
    RIB State: BGP restart is complete
    Send state: in sync
    Active prefixes:              6
    Received prefixes:            6
    Accepted prefixes:            6
    Suppressed due to damping:    0
    Advertised prefixes:          0
  Last traffic (seconds): Received 14   Sent 3    Checked 3
  Input messages:  Total 16     Updates 2       Refreshes 0     Octets 402
  Output messages: Total 15     Updates 0       Refreshes 0     Octets 348
  Output Queue[0]: 0
```

Check the following fields in these outputs to verify that BGP sessions have been established:

- **Peer**—Check if the peer AS number is listed.
- **Local**—Check if the local AS number is listed.

- **State**—Ensure that the value is **Established**. If not, check the configuration again and see **show bgp neighbor** for more details on the output fields.

Similarly, verify that Routers North and South form peer relationships with each other.

Meaning BGP sessions are established between the three routers.

Verifying Prefix Advertisement From Router Internet to Router North

Purpose Verify that the routes sent from Router Internet are received by Router North.

Action

1. From operational mode on Router Internet, run the **show route advertising-protocol bgp *neighbor-address*** command.

```
user@Internet> show route advertising-protocol bgp 10.0.89.13
inet.0: 8 destinations, 8 routes (8 active, 0 holddown, 0 hidden)
  Prefix                  Nexthop              MED     Lclpref    AS path
* 11.1.1.1/32             Self                                    I
* 12.1.1.1/32             Self                                    I
* 13.1.1.1/32             Self                                    I
* 14.1.1.1/32             Self                                    I
* 15.1.1.1/32             Self                                    I
* 192.168.9.1/32          Self                                    I
```

   The output verifies that Router Internet advertises the routes 11.1.1.1/32, 12.1.1.1/32, 13.1.1.1/32, 14.1.1.1/32, 15.1.1.1/32, and 192.168.9.1/32 (the loopback address used as router ID) to Router North.

2. From operational mode on Router North, run the **show route receive-protocol bgp *neighbor-address*** command.

```
user@North> show route receive-protocol bgp 10.0.89.14
inet.0: 12 destinations, 12 routes (12 active, 0 holddown, 0 hidden)
  Prefix                  Nexthop              MED     Lclpref    AS path
* 11.1.1.1/32             10.0.89.14                              300 I
* 12.1.1.1/32             10.0.89.14                              300 I
* 13.1.1.1/32             10.0.89.14                              300 I
* 14.1.1.1/32             10.0.89.14                              300 I
* 15.1.1.1/32             10.0.89.14                              300 I
* 192.168.9.1/32          10.0.89.14                              300 I
```

   The output verifies that Router North has received all the routes advertised by Router Internet.

Meaning Prefixes sent by Router Internet have been successfully installed into the routing table on Router North.

Verifying Prefix Advertisement From Router North to Router South

Purpose Verify that the routes received from Router Internet and the static default route are advertised by Router North to Router South.

Action

1. From operational mode on Router North, run the **show route 0/0 exact** command.

```
user@North> show route 0/0 exact
inet.0: 12 destinations, 12 routes (12 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

0.0.0.0/0          *[Static/5] 00:10:22
                       Reject
```

The output verifies the presence of the static default route (0.0.0.0/0) in the routing table on Router North.

2. From operational mode on Router North, run the **show route advertising-protocol bgp *neighbor-address*** command.

```
user@North> show route advertising-protocol bgp 10.0.78.13
inet.0: 12 destinations, 12 routes (12 active, 0 holddown, 0 hidden)
  Prefix                  Nexthop              MED     Lclpref    AS path
* 0.0.0.0/0               Self                                    I
* 11.1.1.1/32             Self                                    300 I
* 12.1.1.1/32             Self                                    300 I
* 13.1.1.1/32             Self                                    300 I
* 14.1.1.1/32             Self                                    300 I
* 15.1.1.1/32             Self                                    300 I
* 192.168.9.1/32          Self                                    300 I
```

The output verifies that Router North is advertising the static route and the routes received from Router Internet, to Router South.

Verifying BGP Import Policy for Conditional Installation of Prefixes

Purpose Verify that the BGP import policy successfully installs the required prefixes.

Action See if the import policy on Router South is operational by checking if only the static default route from Router North and the 11.1.1.1/32 route from Router Internet are installed in the routing table.

From operational mode, run the **show route receive-protocol bgp *neighbor-address*** command.

```
user@South> show route receive-protocol bgp 10.0.78.14
inet.0: 10 destinations, 10 routes (5 active, 0 holddown, 5 hidden)
  Prefix                  Nexthop              MED     Lclpref    AS path
* 0.0.0.0/0               10.0.78.14                              200 I
* 11.1.1.1/32             10.0.78.14                              200 300 I
```

The output verifies that the BGP import policy is operational on Router South, and only the static default route of 0.0.0.0/0 from Router North and the 11.1.1.1/32 route from Router Internet have leaked into the routing table on Router South. The remaining routes are hidden.

Meaning The conditional installation of prefixes is successful because of the configured BGP import policy.

Verifying the Presence of Routes Hidden by Policy (Optional)

Purpose Verify the presence of routes hidden by the import policy configured on Router South.

NOTE: This section demonstrates the effects of various changes you can make to the configuration depending on your needs.

Action View routes hidden from the routing table of Router South by:

- Using the **hidden** option for the **show route receive-protocol bgp** *neighbor-address* command.
- Deactivating the import policy.

1. From operational mode, run the **show route receive-protocol bgp** *neighbor-address* **hidden** command to view hidden routes.

```
user@South> show route receive-protocol bgp 10.0.78.14 hidden
inet.0: 10 destinations, 10 routes (5 active, 0 holddown, 5 hidden)
  Prefix                  Nexthop              MED     Lclpref    AS path
  12.1.1.1/32             10.0.78.14                              200 300 I
  13.1.1.1/32             10.0.78.14                              200 300 I
  14.1.1.1/32             10.0.78.14                              200 300 I
  15.1.1.1/32             10.0.78.14                              200 300 I
  192.168.9.1/32          10.0.78.14                              200 300 I
```

   The output verifies the presence of routes hidden by the import policy (12.1.1.1/32, 13.1.1.1/32, 14.1.1.1/32, 15.1.1.1/32, and 192.168.9.1/32) on Router South.

2. Deactivate the BGP import policy by configuring the **deactivate import** statement at the **[edit protocols bgp group** *group-name***]** hierarchy level.

```
[edit protocols bgp group toNorth]
user@South# deactivate import
user@South# commit
```

3. Run the **show route receive-protocol bgp** *neighbor-address* operational mode command to check the routes after deactivating the import policy.

```
user@South> show route receive-protocol bgp 10.0.78.14
inet.0: 10 destinations, 10 routes (10 active, 0 holddown, 0 hidden)
  Prefix                  Nexthop              MED     Lclpref    AS path
* 0.0.0.0/0               10.0.78.14                              200 I
* 11.1.1.1/32             10.0.78.14                              200 300 I
* 12.1.1.1/32             10.0.78.14                              200 300 I
* 13.1.1.1/32             10.0.78.14                              200 300 I
* 14.1.1.1/32             10.0.78.14                              200 300 I
* 15.1.1.1/32             10.0.78.14                              200 300 I
* 192.168.9.1/32          10.0.78.14                              200 300 I
```

   The output verifies the presence of previously hidden routes (12.1.1.1/32, 13.1.1.1/32, 14.1.1.1/32, 15.1.1.1/32, and 192.168.9.1/32).

4. Activate the BGP import policy and remove the hidden routes from the routing table by configuring the **activate import** and **keep none** statements respectively at the **[edit protocols bgp group** *group-name***]** hierarchy level.

```
[edit protocols bgp group toNorth]
user@South# activate import
user@South# set keep none
```

```
user@South# commit
```

5. From operational mode, run the **show route receive-protocol bgp** *neighbor-address* **hidden** command to check the routes after activating the import policy and configuring the **keep none** statement.

   ```
   user@South> show route receive-protocol bgp 10.0.78.14 hidden
   inet.0: 5 destinations, 5 routes (5 active, 0 holddown, 0 hidden)
   ```

   The output verifies that the hidden routes are not maintained in the routing table because of the configured **keep none** statement.

Related Documentation

- Conditional Installation of Prefixes Use Cases on page 1
- Understanding Conditional Installation of Prefixes in a Routing Table on page 3