Configure Allowas-in Feature in BGP

Contents

Introduction

This document describes a situation where two branch routers connect via an ISP and run BGP between them.

Prerequisites

Requirements

Cisco recommends that you have knowledge of these topics:

- Internet Service Provider (ISP)
- Border Gateway Protocol (BGP)

Components Used

This document is not restricted to specific software and hardware versions.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, ensure that you understand the potential impact of any command.

Conventions

Refer to Cisco Technical Tips Conventions for more information on document conventions.

Background Information

This document describes a scenario where two branch routers are connected via an Internet Server Provider (ISP) and run Border Gateway Protocol (BGP) between them. The two branch routers (R1 and R2), though at different locations, share the same AS number. Once the routes arrive from a branch (R1 in this case) to the Service Provider (SP) network, they can be tagged with the user AS. Once the SP passes it to the other branch router (R2), by default, the routes can be dropped if the other branch also runs BGP with the SP and uses the same AS number. In this scenario, the **neighbor allowas-in** command is issued in order to allow BGP at the other side to inject updates. This document provides a sample configuration that helps you to understand the Allowas-in feature in BGP.

> **Note**: This feature can only be used for true eBGP peers. You cannot use this feature for two peers that are members of different confederation sub-ASs.

Configure

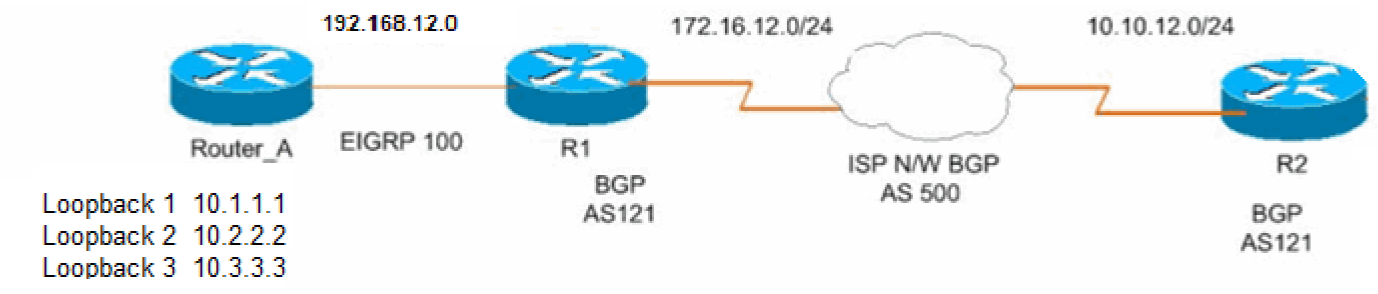
This section presents you with the information to configure the features that this document describes.

> **Note**: Use the Command Lookup Tool to find more information on the commands used in this document.

> **Note**: Only registered Cisco users can access internal Cisco tools and information.

Network Diagram

This document uses this network setup:



Network Diagram

Configurations

This document uses these configurations:

- Router_A
- Router R1
- Router R2

Configuration on Router_A

<#root>

```
Router_A#
interface Loopback1

 ip address 10.1.1.1 255.255.255.255
!
interface Loopback2
 ip address 10.2.2.2 255.255.255.255
!
interface Loopback3
 ip address 10.3.3.3 255.255.255.255
!
interface GigabitEthernet0/1
 no switchport
 ip address 192.168.12.2 255.255.255.0
!
router eigrp 100
 network 10.1.1.1 0.0.0.0
 network 10.2.2.2 0.0.0.0
 network 10.3.3.3 0.0.0.0
 network 192.168.12.0
 auto-summary
!
```

Configuration on Router R1

<#root>

```
R1#

interface Loopback22

 ip address 10.22.22.22 255.255.255.255
!
interface FastEthernet0/0
 ip address 192.168.12.1 255.255.255.0
 duplex auto
 speed auto
!
interface Serial1/0
 ip address 172.16.12.1 255.255.255.0
!
!
router eigrp 100
 network 192.168.12.0
 no auto-summary
!
router bgp 121
 no synchronization
 bgp router-id 10.22.22.22
 bgp log-neighbor-changes
 network 10.22.22.22 mask 255.255.255.255

!--- This is the advertising loopback address.

 redistribute eigrp 100

!--- This shows the redistributing internal routes in BGP.

 neighbor 172.16.12.2 remote-as 500

!--- This shows the EBGP connection with ISP.

 neighbor 172.16.12.2 ebgp-multihop 5
 no auto-summary
!
```

This example shows that the EIGRP runs between Router_A and R1:

```
<#root>

r1#

show ip eigrp neighbors

IP-EIGRP neighbors for process 100
H   Address                 Interface       Hold Uptime   SRTT   RTO  Q  Seq
                                            (sec)         (ms)       Cnt Num
0   192.168.12.2              Fa0/0             14 01:17:12  828  4968  0  7
```

This example shows how Router R1 learns routes from Router_A through EIGRP:

```
<#root>
```

```
r1#
```

show ip route eigrp 100

```
D    10.0.0.1/8 [90/156160] via 192.168.12.2, 00:02:24, FastEthernet0/0
D    10.0.0.2/8 [90/156160] via 192.168.12.2, 00:02:24, FastEthernet0/0
D    10.0.0.3/8 [90/156160] via 192.168.12.2, 00:02:24, FastEthernet0/0
```

This example shows how Router R1 establishes a BGP connection with an ISP running BGP AS500:

<#root>

```
r1#
```

show ip bgp summary

```
BGP router identifier 10.22.22.22, local AS number 121
BGP table version is 19, main routing table version 19
7 network entries using 924 bytes of memory
7 path entries using 364 bytes of memory
5/4 BGP path/bestpath attribute entries using 840 bytes of memory
1 BGP AS-PATH entries using 24 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
Bitfield cache entries: current 1 (at peak 2) using 32 bytes of memory
BGP using 2184 total bytes of memory
BGP activity 40/33 prefixes, 42/35 paths, scan interval 60 secs

Neighbor        V          AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
172.16.12.2     4        500      86      76       19    0    0 00:25:13        2
```

This example shows how R1 announces the BGP learned routes:

<#root>

```
r1#
```

show ip bgp

```
BGP table version is 19, local router ID is 10.22.22.22
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 10.0.0.1          192.168.12.2         156160         32768 ?
*> 10.0.0.2          192.168.12.2         156160         32768 ?
*> 10.0.0.3          192.168.12.2         156160         32768 ?
*> 10.10.12.0/24    172.16.12.2             0              0 500 i
*> 10.22.22.22/32   0.0.0.0                 0          32768 i
r> 172.16.12.0/24   172.16.12.2             0              0 500 i
*> 192.168.12.0       0.0.0.0                 0         32768 ?
```

<#root>

```
r1#

ping 10.10.12.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.10.12.2, timeout is 2 seconds:
!!!!!
!--- This is the connectivity with Router 2 across the Internet cloud.
```

Configuration on Router R2
<#root> R2# **interface Loopback33**   ip address 10.33.33.33 255.255.255.255 ! interface Serial1/0 ip address 10.10.12.1 255.255.255.0 router bgp 121 no synchronization bgp router-id 10.33.33.33 bgp log-neighbor-changes network 10.33.33.33 mask 255.255.255.255 *!--- This is the advertising loopback address.*   neighbor 10.10.12.2 remote-as 500 *!--- This is the EBGP connection with ISP.*   neighbor 10.10.12.2 ebgp-multihop 5 no auto-summary

Router R2 does not learn any routes from router R1.

This is natural behavior because the BGP tries to avoid routing loops. For example, the readvertisement of all prefixes that contain duplicate Autonomous System Numbers (ASNs) is disabled by default.

Redistributed EIGRP routes (10.0.0.1, 10.0.0.2, 10.0.0.3) and the BGP internal route 10.22.22.22 from R1 are not received by R2 because they originate from the same ASN across the Internet. Since R2 see its own AS number (121) in the AS-PATH, R2 does not take those routes.

```
<#root>

r2#

show ip bgp

BGP table version is 20, local router ID is 10.33.33.33
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
```

```
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
r> 10.10.12.0/24    10.10.12.2               0             0 500 i
*> 10.33.33.33/32   0.0.0.0                  0         32768 i
*> 172.16.12.0/24   10.10.12.2               0             0 500 i
```

In order to allow the readvertisement of all prefixes that contain duplicate ASNs, use the **neighbor allowas-in** command in router configuration mode in Router R2.

<#root>

```
r2(config-router)#
```

`neighbor 10.10.12.2 allowas-in`

```
r2#
```

`clear ip bgp*`

```
r2#
```

`show ip bgp`

```
BGP table version is 10, local router ID is 10.33.33.33
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 10.0.0.1          10.10.12.2                              0 500 121 ?
*> 10.0.0.2          10.10.12.2                              0 500 121 ?
*> 10.0.0.3          10.10.12.2                              0 500 121 ?
r> 10.10.12.0/24    10.10.12.2               0             0 500 i
*> 10.22.22.22/32   10.10.12.2                             0 500 121 i
*  10.33.33.33/32   10.10.12.2                             0 500 121 i
*>                  0.0.0.0                  0         32768 i
*> 172.16.12.0/24   10.10.12.2               0             0 500 i
*> 192.168.12.0       10.10.12.2                             0 500 121 ?
```

Now try to ping from R1 to R2:

<#root>

```
r2#
```

`ping 10.22.22.22`

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.22.22.22, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/57/60 ms
```

Verify

There is currently no verification procedure available for this configuration.

Troubleshoot

Error Message

The **%BGP% Neighbor A.B.C.D recv bogus route : AS `loop`** error message is received.

This notification means that the BGP route received by the CE router has its own AS number in the AS path and is considered a router loop for the CE router. As a workaround, configure the CE router with the allowas-in feature as illustrated in the previous example.

Related Information

- **Border Gateway Protocol (BGP)**
- **Cisco Technical Support & Downloads**