BGP Communities

- BGP communities provide additional capability for tagging routes and for modifying BGP routing policy on upstream and downstream routers.
- BGP communities can be appended, removed, or modified selectively on each attribute from router to router.

BGP Communities

BGP communities are an optional transitive BGP attribute that can traverse from AS to AS. A BGP community is a 32-bit number that can be included with a route. A BGP community can be displayed as a full 32-bit number (0 through 4,294,967,295) or as two 16-bit numbers (0 through 65535):(0 through 65535), commonly referred to as new format.

By convention, with private BGP communities, the first 16 bits represent the AS of the community origination, and the second 16 bits represent a pattern defined by the originating AS.

The private BGP community pattern can vary from organization to organization and does not need to be registered. The pattern could signify geographic locations for one AS while signifying a method of route advertisement in another AS.

In 2006, RFC 4360 expanded the capabilities of BGP communities by providing an extended format. Extended BGP communities provide structure for various classes of information and are commonly used for VPN services.

BGP Communities

Enabling BGP Community Support

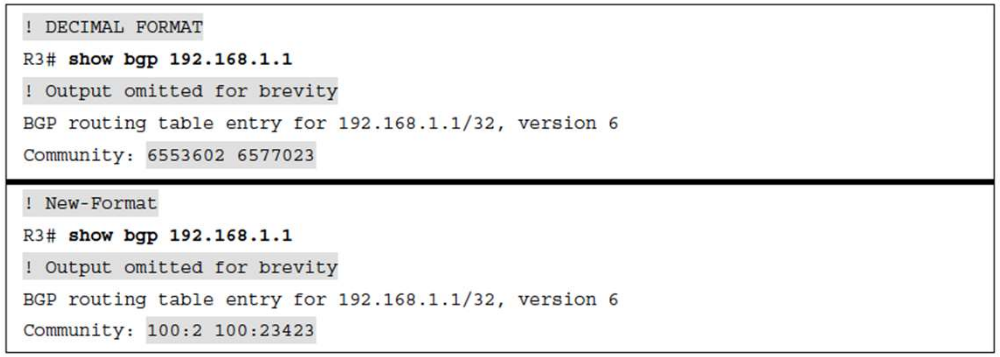
IOS and IOS XE routers do not advertise BGP communities to peers by default. Communities are enabled on a neighbor-by-neighbor basis with the BGP address family configuration command **neighbor** *ip-address* **send-community** [**standard** | **extended** | **both**] under the neighbor's address family configuration.

If a keyword is not specified, standard communities are sent by default.

IOS XE nodes can display communities in new format, and they are easier to read if you use the global configuration command **ip bgp-community new-format**.

Example 12-42 shows the BGP community in decimal format on top and then in new format.

Example 12-42 *BGP Community Formats*



```
! DECIMAL FORMAT
R3# show bgp 192.168.1.1
! Output omitted for brevity
BGP routing table entry for 192.168.1.1/32, version 6
Community: 6553602 6577023
```

```
! New-Format
R3# show bgp 192.168.1.1
! Output omitted for brevity
BGP routing table entry for 192.168.1.1/32, version 6
Community: 100:2 100:23423
```

Well-Known Communities

RFC 1997 defined a set of global communities (known as well-known communities) that use the community range 4,294,901,760 (0xFFFF0000) to 4,294,967,295 (0xFFFFFFFF). All routers that are capable of sending/receiving BGP communities must implement well-known communities.

The following are the common well-known communities:

- Internet
- No_Advertise
- No_Export
- Local-AS

BGP Communities

No-Advertise BGP Community

No_Advertise community routes should not be advertised to any BGP peer. The No_Advertise BGP community can be advertised from an upstream BGP peer or locally with an inbound BGP policy. The No_Advertise community is set with the command **set community no-advertise** within a route map.

- Figure 12-4 demonstrates that R1 is advertising the 10.1.1.0/24 network to R2.
- Example 12-43 shows R2's Network Layer Reachability Information (NLRI) for the 10.1.1.0/24 network prefix.
- BGP routes that are set with the No_Advertise community are quickly seen with the command **show bgp** *afi safi* **community no-advertise**, as shown in Example 12-44.

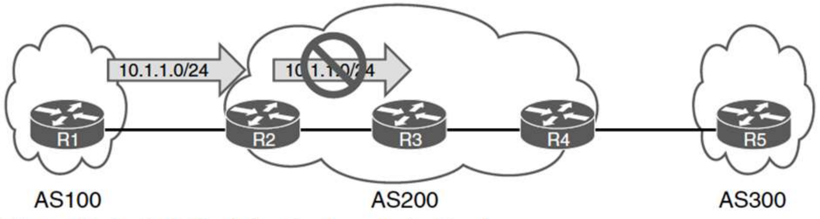
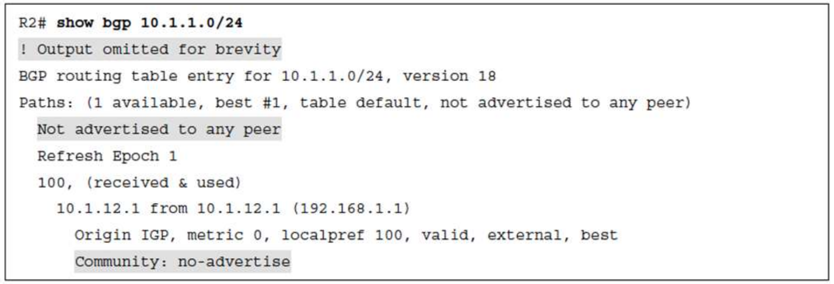


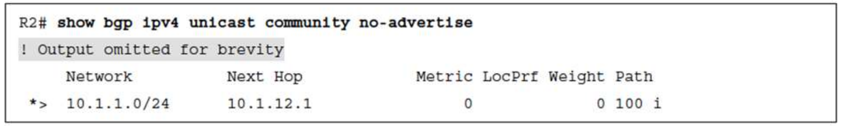
Figure 12-4 *BGP No_Advertise Community Topology*

Example 12-43 *BGP Attributes for No_Advertise Routes*



```
R2# show bgp 10.1.1.0/24
! Output omitted for brevity
BGP routing table entry for 10.1.1.0/24, version 18
Paths: (1 available, best #1, table default, not advertised to any peer)
  Not advertised to any peer
  Refresh Epoch 1
  100, (received & used)
    10.1.12.1 from 10.1.12.1 (192.168.1.1)
      Origin IGP, metric 0, localpref 100, valid, external, best
      Community: no-advertise
```

Example 12-44 *Display of Prefixes with No_Advertise Community*



```
R2# show bgp ipv4 unicast community no-advertise
! Output omitted for brevity
     Network          Next Hop            Metric LocPrf Weight Path
 *>  10.1.1.0/24      10.1.12.1                0             0 100 i
```

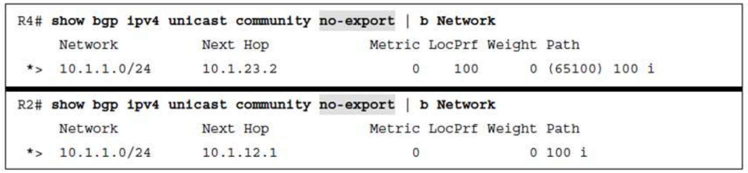
BGP Communities

No-Export BGP Community

When a route is received with the No_Export community, the route is not advertised to any eBGP peer. The No_Export community is set with the command **set community no-export** within a route map.

- Figure 12-5 shows a topology with three ASs.
- Example 12-45 confirms that the prefix is not advertised to any eBGP peer.
- The command **show bgp** *afi safi* **community no-export** shows all the BGP prefixes that contain the Local-AS community, as demonstrated in Example 12-46.

Example 12-46 *Viewing BGP Routes with the No_Export Community*



```
R4# show bgp ipv4 unicast community no-export | b Network
     Network          Next Hop            Metric LocPrf Weight Path
 *>  10.1.1.0/24      10.1.23.2                0    100      0 (65100) 100 i
```

```
R2# show bgp ipv4 unicast community no-export | b Network
     Network          Next Hop            Metric LocPrf Weight Path
 *>  10.1.1.0/24      10.1.12.1                0             0 100 i
```

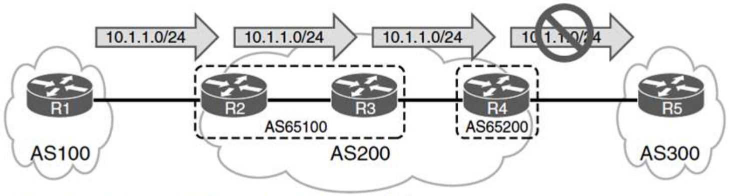
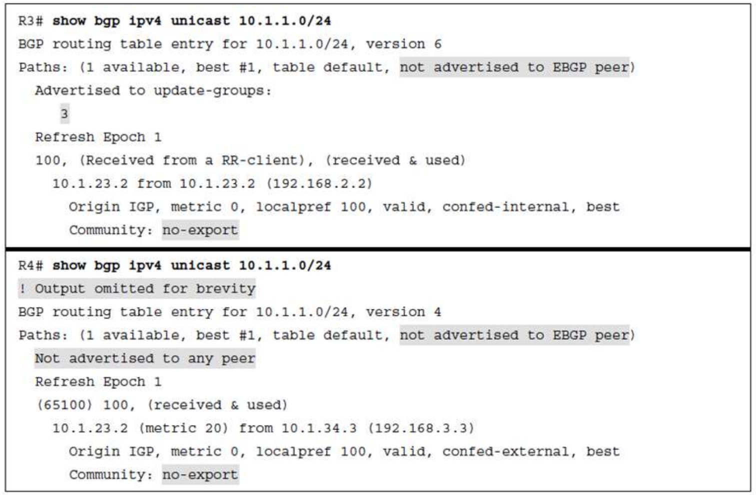


Figure 12-5 *BGP No-Export Community Topology*

Example 12-45 *BGP Attributes for No_Export Routes*



```
R3# show bgp ipv4 unicast 10.1.1.0/24
BGP routing table entry for 10.1.1.0/24, version 6
Paths: (1 available, best #1, table default, not advertised to EBGP peer)
  Advertised to update-groups:
     3
  Refresh Epoch 1
  100, (Received from a RR-client), (received & used)
    10.1.23.2 from 10.1.23.2 (192.168.2.2)
      Origin IGP, metric 0, localpref 100, valid, confed-internal, best
      Community: no-export
```

```
R4# show bgp ipv4 unicast 10.1.1.0/24
! Output omitted for brevity
BGP routing table entry for 10.1.1.0/24, version 4
Paths: (1 available, best #1, table default, not advertised to EBGP peer)
  Not advertised to any peer
  Refresh Epoch 1
  (65100) 100, (received & used)
    10.1.23.2 (metric 20) from 10.1.34.3 (192.168.3.3)
      Origin IGP, metric 0, localpref 100, valid, confed-external, best
      Community: no-export
```

BGP Communities

No-Export SubConfed Community

With the No_Export_SubConfed community known as the Local-AS community, a route is not advertised outside the local AS. The Local- AS community is set with the command **set community local-as** within a route map.

- Figure 12-6 shows a topology with three ASs.
- Example 12-47 confirms that the prefix is not advertised outside local AS and that the prefix is not advertised to any peer.
- The command **show bgp** *afi safi* **community local-as** shows all the BGP prefixes that contain the Local-AS community, as demonstrated in Example 12-48.

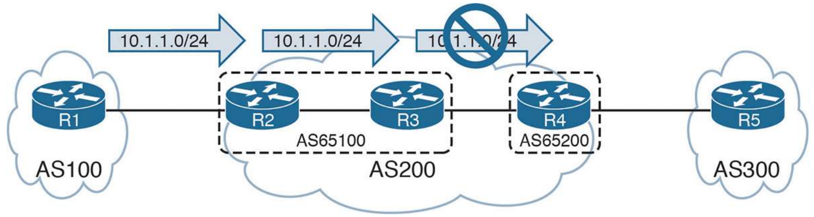
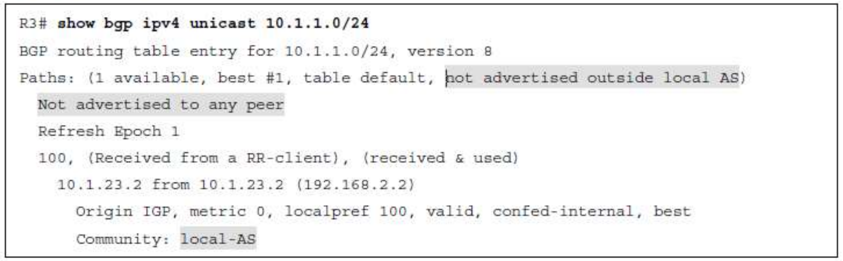


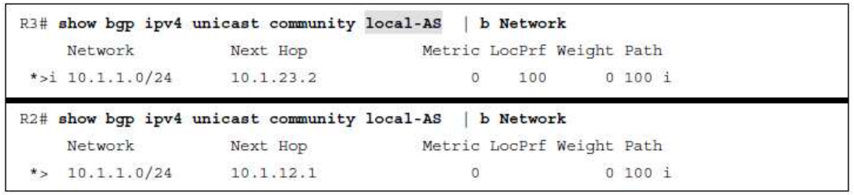
Figure 12-6 *BGP Local-AS Community Topology*

Example 12-47 *BGP Attributes for Local-AS Routes*



```
R3# show bgp ipv4 unicast 10.1.1.0/24
BGP routing table entry for 10.1.1.0/24, version 8
Paths: (1 available, best #1, table default, not advertised outside local AS)
  Not advertised to any peer
  Refresh Epoch 1
  100, (Received from a RR-client), (received & used)
    10.1.23.2 from 10.1.23.2 (192.168.2.2)
      Origin IGP, metric 0, localpref 100, valid, confed-internal, best
      Community: local-AS
```

Example 12-48 *Viewing BGP Routes with the Local-AS Community*



```
R3# show bgp ipv4 unicast community local-AS  | b Network
     Network          Next Hop            Metric LocPrf Weight Path
 *>i 10.1.1.0/24      10.1.23.2                0    100      0 100 i
```

```
R2# show bgp ipv4 unicast community local-AS  | b Network
     Network          Next Hop            Metric LocPrf Weight Path
 *>  10.1.1.0/24      10.1.12.1                0             0 100 i
```

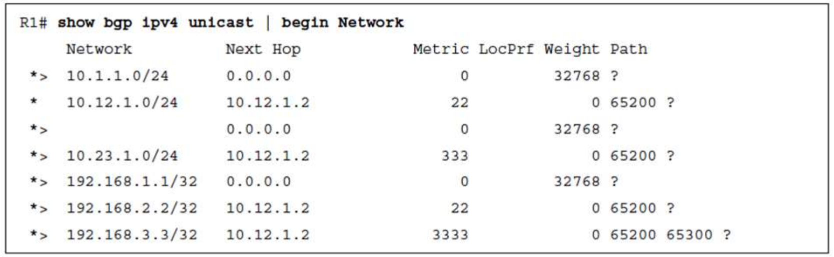
BGP Communities

Conditionally Matching BGP Communities

Conditionally matching BGP communities allows for selection of routes based on the BGP communities within the route’s path attributes so that selective processing can occur in route maps.

You display the entire BGP table by using the command **show bgp** *afi safi* **detail** and then manually select a route with a specific community. However, if the BGP community is known, you can display all the routes by using the command **show bgp** *afi safi* **community** *community*.

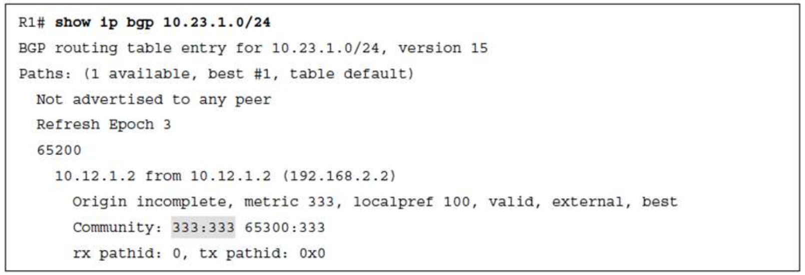
Example 12-49 *BGP Routes from R2 (AS 65200)*



```
R1# show bgp ipv4 unicast | begin Network
     Network          Next Hop           Metric LocPrf Weight Path
 *>  10.1.1.0/24      0.0.0.0                 0         32768 ?
 *   10.12.1.0/24     10.12.1.2              22             0 65200 ?
 *>                   0.0.0.0                 0         32768 ?
 *>  10.23.1.0/24     10.12.1.2             333             0 65200 ?
 *>  192.168.1.1/32   0.0.0.0                 0         32768 ?
 *>  192.168.2.2/32   10.12.1.2              22             0 65200 ?
 *>  192.168.3.3/32   10.12.1.2            3333             0 65200 65300 ?
```

Example 12-49 shows the BGP table for R1, which has received multiple routes from R2 (AS 65200).

Example 12-50 *Viewing BGP Path Attributes for the 10.23.1.0/24 Network*



```
R1# show ip bgp 10.23.1.0/24
BGP routing table entry for 10.23.1.0/24, version 15
Paths: (1 available, best #1, table default)
  Not advertised to any peer
  Refresh Epoch 3
  65200
    10.12.1.2 from 10.12.1.2 (192.168.2.2)
      Origin incomplete, metric 333, localpref 100, valid, external, best
      Community: 333:333 65300:333
      rx pathid: 0, tx pathid: 0x0
```

Example 12-50 shows the explicit path entry for the 10.23.1.0/24 network and all the BGP path attributes.

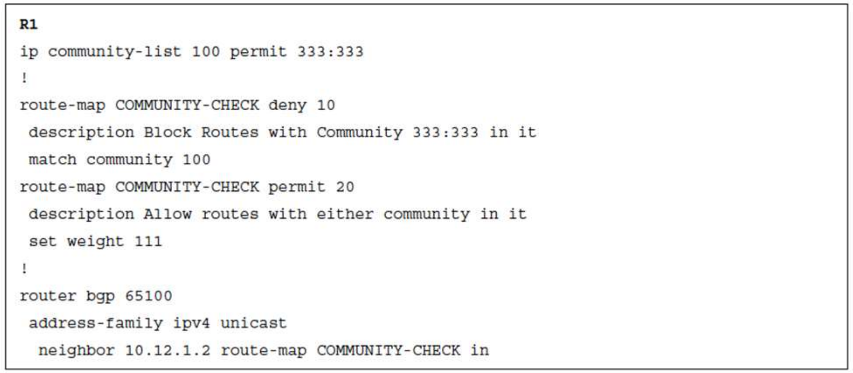
BGP Communities

Conditionally Matching BGP Communities (Cont.)

Conditionally matching requires the creation of a community list that shares a similar structure to an ACL, can be standard or expanded, and can be referenced by number or name.

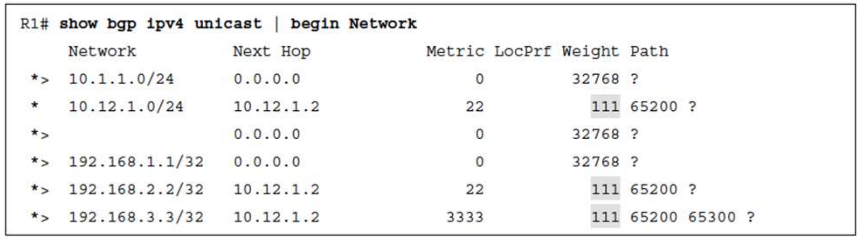
- The configuration syntax for a community list is **ip community-list** {*1-500* | **standard** *list-name* | **expanded** *list-name*} {**permit** | **deny**} *community-pattern*. The community list is referenced in a route map with the command **match community** *1-500*.
- Example 12-51 demonstrates the creation of a BGP community list that matches on the community 333:333.
- Example 12-52 shows the BGP table after the route map has been applied to the neighbor.

Example 12-51 *Conditionally Matching BGP Communities*



```
R1
ip community-list 100 permit 333:333
!
route-map COMMUNITY-CHECK deny 10
 description Block Routes with Community 333:333 in it
 match community 100
route-map COMMUNITY-CHECK permit 20
 description Allow routes with either community in it
 set weight 111
!
router bgp 65100
 address-family ipv4 unicast
  neighbor 10.12.1.2 route-map COMMUNITY-CHECK in
```

Example 12-52 *R1's BGP Table After Application of the Route Map*



```
R1# show bgp ipv4 unicast | begin Network
     Network          Next Hop            Metric LocPrf Weight Path
 *>  10.1.1.0/24      0.0.0.0                  0         32768 ?
 *   10.12.1.0/24     10.12.1.2               22           111 65200 ?
 *>                   0.0.0.0                  0         32768 ?
 *>  192.168.1.1/32   0.0.0.0                  0         32768 ?
 *>  192.168.2.2/32   10.12.1.2               22           111 65200 ?
 *>  192.168.3.3/32   10.12.1.2             3333           111 65200 65300 ?
```

BGP Communities

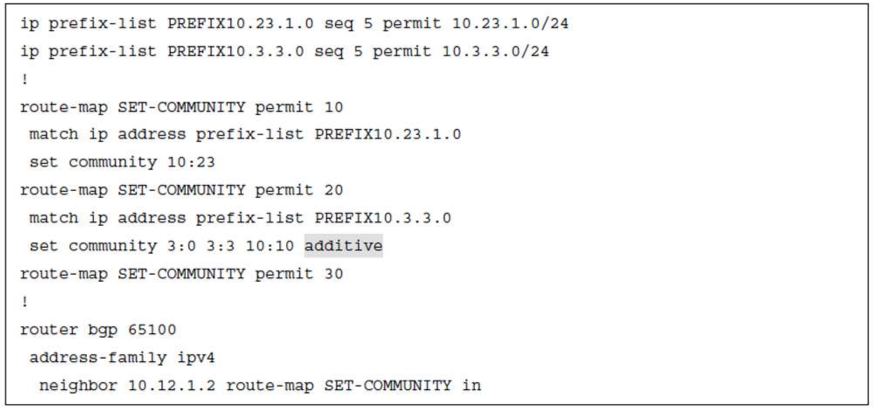
Private BGP Communities

You set a private BGP community in a route map by using the command set community bgp-community [additive]. By default, when you set a community, any existing communities are overwritten, but you can preserve them by using the optional additive keyword.

Example 12-54 shows the configuration where the BGP community is set on the 10.23.1.0/24 network.

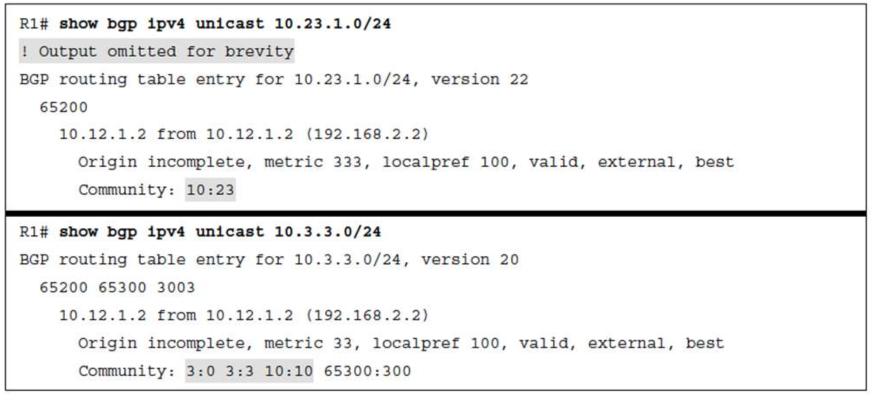
After the route map has been applied and the routes have been refreshed, the path attributes can be examined, as demonstrated in Example 12-55.

Example 12-54 *Private BGP Community Configuration*



```
ip prefix-list PREFIX10.23.1.0 seq 5 permit 10.23.1.0/24
ip prefix-list PREFIX10.3.3.0 seq 5 permit 10.3.3.0/24
!
route-map SET-COMMUNITY permit 10
 match ip address prefix-list PREFIX10.23.1.0
 set community 10:23
route-map SET-COMMUNITY permit 20
 match ip address prefix-list PREFIX10.3.3.0
 set community 3:0 3:3 10:10 additive
route-map SET-COMMUNITY permit 30
!
router bgp 65100
 address-family ipv4
  neighbor 10.12.1.2 route-map SET-COMMUNITY in
```

Example 12-55 *Verification of BGP Community Changes*



```
R1# show bgp ipv4 unicast 10.23.1.0/24
! Output omitted for brevity
BGP routing table entry for 10.23.1.0/24, version 22
  65200
    10.12.1.2 from 10.12.1.2 (192.168.2.2)
      Origin incomplete, metric 333, localpref 100, valid, external, best
      Community: 10:23
```

```
R1# show bgp ipv4 unicast 10.3.3.0/24
BGP routing table entry for 10.3.3.0/24, version 20
  65200 65300 3003
    10.12.1.2 from 10.12.1.2 (192.168.2.2)
      Origin incomplete, metric 33, localpref 100, valid, external, best
      Community: 3:0 3:3 10:10 65300:300
```