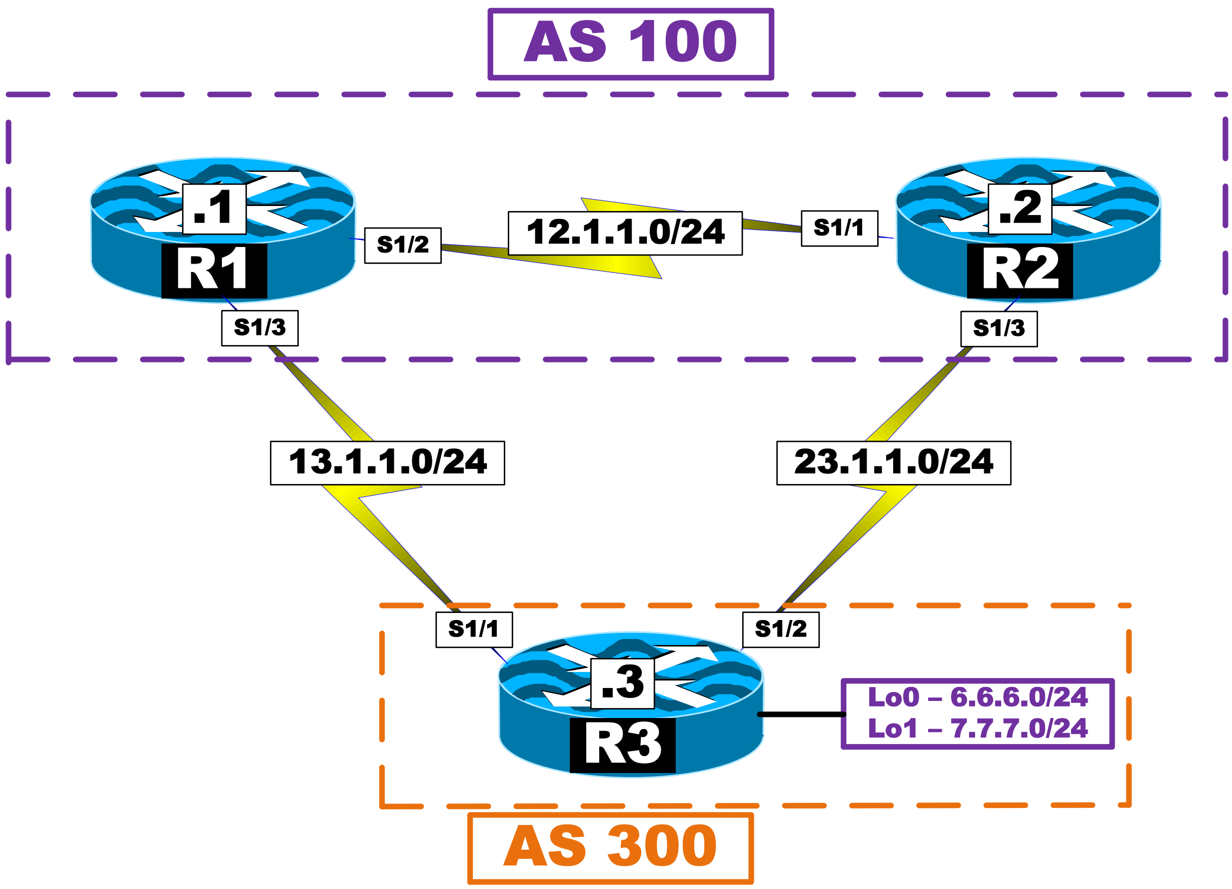
Lab -1 Private Communities - II



Task 1

Configure the above topology.

On R1:

```
R1(config)#int s1/2
R1(config-if)#ip addr 12.1.1.1 255.255.255.0
R1(config-if)#no shut

R1(config)#int s1/3
R1(config-if)#ip addr 13.1.1.1 255.255.255.0
```

```
R1(config-if)#no shu
```

On R2:

```
R2(config)#int s1/1
R2(config-if)#ip addr 12.1.1.2 255.255.255.0
R2(config-if)#no shu

R2(config)#int s1/3
R2(config-if)#ip addr 23.1.1.2 255.255.255.0
R2(config-if)#no shu
```

On R3:

```
R3(config)#int s1/1
R3(config-if)#ip addr 13.1.1.3 255.255.255.0
R3(config-if)#no shu

R3(config)#int s1/2
R3(config-if)#ip addr 23.1.1.3 255.255.255.0
R3(config-if)#no shut

R3(config)#int lo0
R3(config-if)#ip addr 6.6.6.6 255.255.255.0

R3(config)#int lo1
R3(config-if)#ip addr 7.7.7.7 255.255.255.0
```

Task 2

The provider and the customer have agreed on the following:

Community Value	Local Preference
100:300	130
100:250	125

The policy states that if the customer announces the prefixes with a community value of 100:300, then the provider sets the local preference of those routes to 130, and the provider sets the local preference to 125 if the customer sets the community value to 100:250.

This allows you to control the routing policy within the provider's network by using the community attribute.

The customer in AS 300 has to implement the following policy:

The ingress traffic destined to network 6.6.6.0/24 should use the R1-R3 link, if this link is down, then the traffic destined to 6.6.6.0/24 network should take the R1-R2 link.

The ingress traffic destined to network 7.7.7.0/24 should use the R1-R2 link, if this link is down, then the traffic destined to 7.7.7.0/24 network should take the R1-R3 link.

To implement this policy R3 announces its prefixes as follows:

R3 announces its prefixes as follows:

To R1:

6.6.6.0/24 with a community value of 100:300
7.7.7.0/24 with a community value of 100:250

To R2:

6.6.6.0/24 with a community value of 100:250
7.7.7.0/24 with a community value of 100:300

```
R3(config)#access-list 101 permit ip host 6.6.6.0 host 255.255.255.0
R3(config)#access-list 102 permit ip host 7.7.7.0 host 255.255.255.0

R3(config)#route-map to-R1 permit 10
R3(config-route-map)#match ip addr 101
R3(config-route-map)#set community 100:300

R3(config-route-map)#route-map to-R1 permit 20
R3(config-route-map)#match ip addr 102
R3(config-route-map)#set community 100:250

R3(config)#route-map to-R2 permit 10
R3(config-route-map)#match ip addr 101
R3(config-route-map)#set community 100:250

R3(config-route-map)#route-map to-R2 permit 20
R3(config-route-map)#match ip addr 102
R3(config-route-map)#set community 100:300

R3(config)#router bgp 300
R3(config-router)#network 6.6.6.0 mask 255.255.255.0
```

```
R3(config-router)#network 7.7.7.0 mask 255.255.255.0

R3(config-router)#neigh 13.1.1.1 remote 100
R3(config-router)#neigh 13.1.1.1 send-community
R3(config-router)#neigh 13.1.1.1 route-map to-R1 out

R3(config-router)#neigh 23.1.1.2 remote 100
R3(config-router)#neigh 23.1.1.2 send-community
R3(config-router)#neigh 23.1.1.2 route-map to-R2 out

R3(config)#ip bgp-community new-format
```

On R1:

```
R1(config)#ip bgp-community new-format

R1(config)#ip community-list 1 permit 100:300
R1(config)#ip community-list 2 permit 100:250

R1(config)#route-map to-R3 permit 10
R1(config-route-map)#match community 1
R1(config-route-map)#set local-preference 130

R1(config)#route-map to-R3 permit 20
R1(config-route-map)#match community 2
R1(config-route-map)#set local-preference 125

R1(config)#route-map to-R3 permit 30

R1(config)#router bgp 100
R1(config-router)#neigh 12.1.1.2 remote 100
R1(config-router)#neigh 12.1.1.2 next-hop-self

R1(config-router)#neigh 13.1.1.3 remote 300
R1(config-router)#neigh 13.1.1.3 route-map to-R3 in
```

On R2:

```
R2(config)#ip bgp-community new-format

R2(config)#ip community-list 1 permit 100:300
R2(config)#ip community-list 2 permit 100:250

R2(config)#route-map to-R3 permit 10
R2(config-route-map)#match community 1
R2(config-route-map)#set local-preference 130
```

```
R2(config)#route-map to-R3 permit 20

R2(config-route-map)#match community 2
R2(config-route-map)#set local-preference 125

R2(config)#route-map to-R3 permit 30

R2(config)#router bgp 100
R2(config-router)#neigh 12.1.1.1 remote 100
R2(config-router)#neigh 12.1.1.1 next-hop-self

R2(config-router)#neigh 23.1.1.3 remote 300
R2(config-router)#neigh 23.1.1.3 remote 300
R2(config-router)#neigh 23.1.1.3 route-map to-R3 in
```

To verify the configuration:

On All Routers:

```
Rx#cle ip bgp *

R1#sh ip bgp 6.6.6.0
BGP routing table entry for 6.6.6.0/24, version 8
Paths: (1 available, best #1, table default)
  Advertised to update-groups:
     5
  Refresh Epoch 1
  300
    13.1.1.3 from 13.1.1.3 (7.7.7.7)
      Origin IGP, metric 0, localpref 130, valid, external, best
      Community: 100:300
      rx pathid: 0, tx pathid: 0x0

R1#sh ip bgp 7.7.7.0
BGP routing table entry for 7.7.7.0/24, version 9
Paths: (2 available, best #1, table default)
  Advertised to update-groups:
     6
  Refresh Epoch 1
  300
    12.1.1.2 from 12.1.1.2 (23.1.1.2)
      Origin IGP, metric 0, localpref 130, valid, internal, best
      rx pathid: 0, tx pathid: 0x0
  Refresh Epoch 1
  300
    13.1.1.3 from 13.1.1.3 (7.7.7.7)
      Origin IGP, metric 0, localpref 125, valid, external
```

```
      Community: 100:250

      rx pathid: 0, tx pathid: 0

R1#sh ip bgp | b Network
     Network          Next Hop            Metric LocPrf Weight Path
 *>   6.6.6.0/24       13.1.1.3                 0    130      0 300 i
 *>i 7.7.7.0/24       12.1.1.2                 0    130      0 300 i
 *                     13.1.1.3                 0    125      0 300 i

R1#sh ip route 6.6.6.0
Routing entry for 6.6.6.0/24
  Known via "bgp 100", distance 20, metric 0
  Tag 300, type external
  Last update from 13.1.1.3 00:10:06 ago
  Routing Descriptor Blocks:
  * 13.1.1.3, from 13.1.1.3, 00:10:06 ago
      Route metric is 0, traffic share count is 1
      AS Hops 1
      Route tag 300
      MPLS label: none

R1#sh ip route 7.7.7.0
Routing entry for 7.7.7.0/24
  Known via "bgp 100", distance 200, metric 0
  Tag 300, type internal
  Last update from 12.1.1.2 00:10:14 ago
  Routing Descriptor Blocks:
  * 12.1.1.2, from 12.1.1.2, 00:10:14 ago
      Route metric is 0, traffic share count is 1
      AS Hops 1
      Route tag 300
      MPLS label: none
```

Let's add another loopback interface to R3 and advertise it in BGP.

```
R3(config)#int lo2
R3(config-if)#ip addr 3.3.3.3 255.255.255.0

R3(config)#router bgp 300
R3(config-router)#network 3.3.3.0 mask 255.255.255.0

R3(config)#route-map to-R1 permit 90
R3(config)#route-map to-R2 permit 90

R3#cle ip bgp * out
```

To verify the configuration:

On R1:

```
R1#trace 3.3.3.3 num
Type escape sequence to abort.
Tracing the route to 3.3.3.3
VRF info: (vrf in name/id, vrf out name/id)
  1 13.1.1.3 9 msec *  9 msec
```

As we can see that R1 takes R3 over R2, and the reason is that external routes are preferred over internal, to see this:

```
R1#sh ip bgp 3.3.3.0
BGP routing table entry for 3.3.3.0/24, version 10
Paths: (2 available, best #2, table default)
  Advertised to update-groups:
     5
  Refresh Epoch 1
  300
    12.1.1.2 from 12.1.1.2 (23.1.1.2)
      Origin IGP, metric 0, localpref 100, valid, internal
      rx pathid: 0, tx pathid: 0
  Refresh Epoch 3
  300
    13.1.1.3 from 13.1.1.3 (7.7.7.7)
      Origin IGP, metric 0, localpref 100, valid, external, best
      rx pathid: 0, tx pathid: 0x0
```

Let's add a loopback on R1 and advertise it in BGP. This is done to provide a return path to the other routers, now, we can source the trace based on the loopback:

```
R1(config)#int lo1
R1(config-if)#ip addr 1.1.1.1 255.0.0.0
R1(config-if)#router bgp 100
R1(config-router)#netw 1.0.0.0

R1#trace 6.6.6.6 num source lo1
Type escape sequence to abort.
Tracing the route to 6.6.6.6
VRF info: (vrf in name/id, vrf out name/id)
  1 13.1.1.3 8 msec *  8 msec

R1#trace 7.7.7.7 num source lo1
Type escape sequence to abort.
Tracing the route to 7.7.7.7
```

```
VRF info: (vrf in name/id, vrf out name/id)
  1 12.1.1.2 4 msec 8 msec 8 msec

  2 23.1.1.3 12 msec *  13 msec
```

To see the reason for the above result:

```
R1#sh ip bgp | b Net
     Network          Next Hop            Metric LocPrf Weight Path
 *>  1.0.0.0          0.0.0.0                  0         32768 i
 * i 3.3.3.0/24       12.1.1.2                 0    100      0 300 i
 *>                   13.1.1.3                 0             0 300 i
 *>  6.6.6.0/24       13.1.1.3                 0    130      0 300 i
 *>i 7.7.7.0/24       12.1.1.2                 0    130      0 300 i
 *                    13.1.1.3                 0    125      0 300 i
```

Let's shutdown the link between R1 and R3 for testing purpose:

```
R1(config)#int s1/3
R1(config-if)#shut

R1#sh ip bgp | b Net
     Network          Next Hop            Metric LocPrf Weight Path
 *>  1.0.0.0          0.0.0.0                  0         32768 i
 *>i 3.3.3.0/24       12.1.1.2                 0    100      0 300 i
 *>i 6.6.6.0/24       12.1.1.2                 0    125      0 300 i
 *>i 7.7.7.0/24       12.1.1.2                 0    130      0 300 i

R1#trace 7.7.7.7 num source lo1
Type escape sequence to abort.
Tracing the route to 7.7.7.7
VRF info: (vrf in name/id, vrf out name/id)
  1 12.1.1.2 9 msec 8 msec 9 msec
  2 23.1.1.3 17 msec *  18 msec

R1#trace 6.6.6.6 num source lo1
Type escape sequence to abort.
Tracing the route to 6.6.6.6
VRF info: (vrf in name/id, vrf out name/id)
  1 12.1.1.2 8 msec 8 msec 9 msec
  2 23.1.1.3 18 msec *  17 msec
```

Let's enable the link between R1 and R3:

```
R1(config)#int s1/3
R1(config-if)#no shut
```

```
R1#sh ip bgp | b Net
     Network          Next Hop            Metric LocPrf Weight Path
 *>  1.0.0.0          0.0.0.0                  0         32768 i
 *>  3.3.3.0/24       13.1.1.3                 0             0 300 i
 * i                  12.1.1.2                 0    100      0 300 i
 *>  6.6.6.0/24       13.1.1.3                 0    130      0 300 i
 *   7.7.7.0/24       13.1.1.3                 0    125      0 300 i
 *>i                  12.1.1.2                 0    130      0 300 i
R1#
```

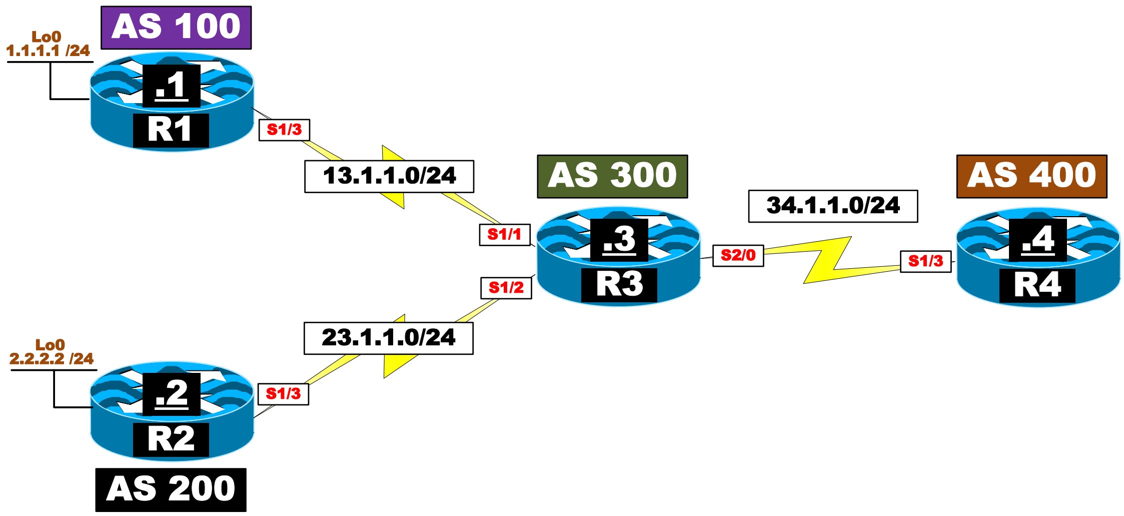
On R2:

```
R2#sh ip bgp | b Net
     Network          Next Hop            Metric LocPrf Weight Path
 *>i 1.0.0.0          12.1.1.1                 0    100      0 i
 * i 3.3.3.0/24       12.1.1.1                 0    100      0 300 i
 *>                   23.1.1.3                 0             0 300 i
 *>i 6.6.6.0/24       12.1.1.1                 0    130      0 300 i
 *                    23.1.1.3                 0    125      0 300 i
 *>  7.7.7.0/24       23.1.1.3                 0    130      0 300 i
```

Task 3

Erase the startup configuration on all routers and switches and reload them before proceeding to the next lab.

Lab 2 – BGP Automatic-tag, AS-path tag and Table-map



Task 1

Configure BGP based on the following requirements:

- Configure R1 in AS 100 to establish a BGP peer session with R3 in AS 300. R1 should be configured to advertise its loopback 0 interface in this AS.
- Configure R2 in AS 200 to establish a BGP peer session with R3 in AS 300. R2 should be configured to advertise its loopback 0 interface in this AS.
- Configure R3 to advertise a default route to R1 in AS 100 and R2 in AS 200.
- Configure R4 in AS 400. This router should NOT establish a peer session with any of the routers in this topology. The router-id of this router should be configured to be 4.4.4.4.

On R1:

```
R1(config)#router bgp 100
R1(config-router)#no au
R1(config-router)#neighbor 13.1.1.3 remote-as 200
R1(config-router)#netw 1.1.1.0 mask 255.255.255.0
```

On R2:

```
R2(config)#router bgp 200
R2(config-router)#no au
R2(config-router)#neighbor 23.1.1.3 remote-as 300
R2(config-router)#netw 2.2.2.0 mask 255.255.255.0
```

On R3:

```
R3(config)#router bgp 300
R3(config-router)#no au
R3(config-router)#neighbor 13.1.1.1 remote-as 100
R3(config-router)#neighbor 13.1.1.1 default-originate
R3(config-router)#neighbor 23.1.1.2 remote-as 200
R3(config-router)#neighbor 23.1.1.2 default-originate
```

On R4:

```
R4(config)#router bgp 400
R4(config-router)#no au
R4(config-router)#bgp router-id 4.4.4.4
```

To verify the configuration:

On R3:

```
R3#sh ip bgp | b Network

   Network          Next Hop            Metric LocPrf Weight Path
*> 1.1.1.0/24       13.1.1.1                 0             0 100 i
*> 2.2.2.0/24       23.1.1.2                 0             0 200 i
```

On R1:

```
R1#sh ip bgp | b Network

   Network          Next Hop            Metric LocPrf Weight Path
*> 0.0.0.0          13.1.1.3                 0             0 300 i
*> 1.1.1.0/24       0.0.0.0                  0         32768 i
*> 2.2.2.0/24       13.1.1.3                               0 300 200 i
```

On R2:

```
R2#sh ip bgp | b Network

   Network          Next Hop            Metric LocPrf Weight Path
*> 0.0.0.0          23.1.1.3                 0             0 300 i
*> 1.1.1.0/24       23.1.1.3                               0 300 100 i
*> 2.2.2.0/24       0.0.0.0                  0         32768 i
```

Task 2

Configure OSPF based on the following requirements:

- Configure OSPF area 0 on R3's S2/0 interface. The router-id of this router should be set to 0.0.0.3.
- Configure OSPF area 0 on R4's S1/3 interface. The router-id of this router should be set to 0.0.0.4.
- R3 should be configured to redistribute BGP into OSPF.
- R4 should be configured to redistribute OSPF into BGP.
- R4 should have networks 1.1.1.0/24 and 2.2.2.0/24 in its BGP table with the AS numbers in which these two prefixes originated in.

On R3:

```
R3(config)#router ospf 1
R3(config-router)#router-id 0.0.0.3
R3(config-router)#netw 34.1.1.3 0.0.0.0 area 0
R3(config-router)#redistribute bgp 300 subnets
```

On R4:

```
R4(config)#router ospf 1
R4(config-router)#router-id 0.0.0.4
R4(config-router)#netw 34.1.1.4 0.0.0.0 area 0

R4(config)#router bgp 400
R4(config-router)#redistribute ospf 1 match internal external
```

To verify the configuration:

On R3:

```
R3#sh ip ospf database

            OSPF router with ID (0.0.0.3) (Process ID 1)

                router Link States (Area 0)

Link ID         ADV router      Age         Seq#       Checksum Link
count
0.0.0.3         0.0.0.3         173         0x80000004 0x00AA9F 2
0.0.0.4         0.0.0.4         179         0x80000003 0x009AB0 2

                Type-5 AS External Link States

Link ID         ADV router      Age         Seq#       Checksum Tag
1.1.1.0         0.0.0.3         265         0x80000001 0x000247 100
2.2.2.0         0.0.0.3         265         0x80000001 0x00ECF4 200
```

When BGP is redistributed into an IGP, the BGP AS number is copied as a route tag in that given IGP. By looking at the external routes, we can see that netw 1.1.1.0 has a Tag of 100, and netw 2.2.2.0 is tagged with 200.

Let's verify the BGP table on R4:

On R4:

```
R4#sh ip bgp | b Network

   Network          Next Hop            Metric LocPrf Weight Path
*> 1.1.1.0/24       34.1.1.3                 1         32768 ?
*> 2.2.2.0/24       34.1.1.3                 1         32768 ?
*> 34.1.1.0/24      0.0.0.0                  0         32768 ?
```

What happened to the AS-Path?

NOTE: When OSPF was redistributed into BGP, the AS-Path was lost and the origin code was set to unknown.

This AS-Path can be retrieved on R4 using the tag that was set when the BGP was redistributed into OSPF; this can be configured using the "AS-Path tag", let's configure this and verify:

On R4:

```
R4(config)#route-map tst permit 10
R4(config-route-map)#Set as-path tag

R4(config)#router bgp 400
R4(config-router)#redistribute ospf 1 match internal external route-map tst
```

To verify the configuration:

On R4:

```
R4#sh ip bgp | b Network

   Network          Next Hop            Metric LocPrf Weight Path
*> 1.1.1.0/24       34.1.1.3                 1         32768 100 ?
*> 2.2.2.0/24       34.1.1.3                 1         32768 200 ?
*> 34.1.1.0/24      0.0.0.0                  0         32768 ?
```

The output of the above show command reveals that the route tags which were the BGP AS numbers are copied into the AS-Path. But the second problem that we are facing is the origin code, which is set to unknown.
R3 can be configured to fix this problem, in this case R3 can be configured to use the "Automatic-tag and the Table-map" as follows:

On R3:

The following configures a route-map and sets the "automatic-tag":

```
R3(config)#route-map tst permit 10
R3(config-route-map)#set automatic-tag
```

The route-map is referenced under router BGP using the "Table-map":

```
R3(config)#router bgp 300
R3(config-router)#table-map tst
```

To implement the changes:

```
R3#clear ip bgp *
```

To verify the configuration:

On R4:

```
R4#sh ip bgp | b Network
```

```
   Network          Next Hop            Metric LocPrf Weight Path
*> 1.1.1.0/24       34.1.1.3                 1         32768 3489661028 i
*> 2.2.2.0/24       34.1.1.3                 1         32768 3489661128 i
*> 34.1.1.0/24      0.0.0.0                  0         32768 ?
```

So what happened to the AS number?
The AS numbers are correct but we have to go through a calculation to figure them, let's do this:

Take the number and convert it to Hex:
3489661028 → D0000064
Let's focus on the last 4 characters "0064" and convert it back to decimal → 100

Let's convert the AS number for netw 2.2.2.0/24:
3489661128 → D00000C8
Let's convert the last 4 characters "00C8" to decimal → 200

There MUST be an easier/better way to do this, how would you handle this kind of issue in the real world?

Let's remove the "route-map tst" that was configured on R4:

On R4:

```
R4(config)#no route-map tst

R4(config)#router bgp 400
R4(config-router)#no redis ospf 1 match inter exte 1 exte 2 route-map tst
R4(config-router)#redistr ospf 1 match inter exter

R4#clear ip bgp *
```

NOW, let's check the OSPF database and verify the tag for networks 1.1.1.0/24 and 2.2.2.0/24:

```
R4#sh ip ospf database


            OSPF router with ID (0.0.0.4) (Process ID 1)

                router Link States (Area 0)

Link ID         ADV router      Age         Seq#       Checksum Link
count
0.0.0.3         0.0.0.3         1607        0x80000004 0x009B0F 2
0.0.0.4         0.0.0.4         1606        0x80000002 0x008D1F 2
```

```
                Type-5 AS External Link States

Link ID         ADV router      Age         Seq#       Checksum Tag
1.1.1.0         0.0.0.3         1326        0x80000003 0x003A3C 3489661028
2.2.2.0         0.0.0.3         1326        0x80000003 0x0025E9 3489661128
```

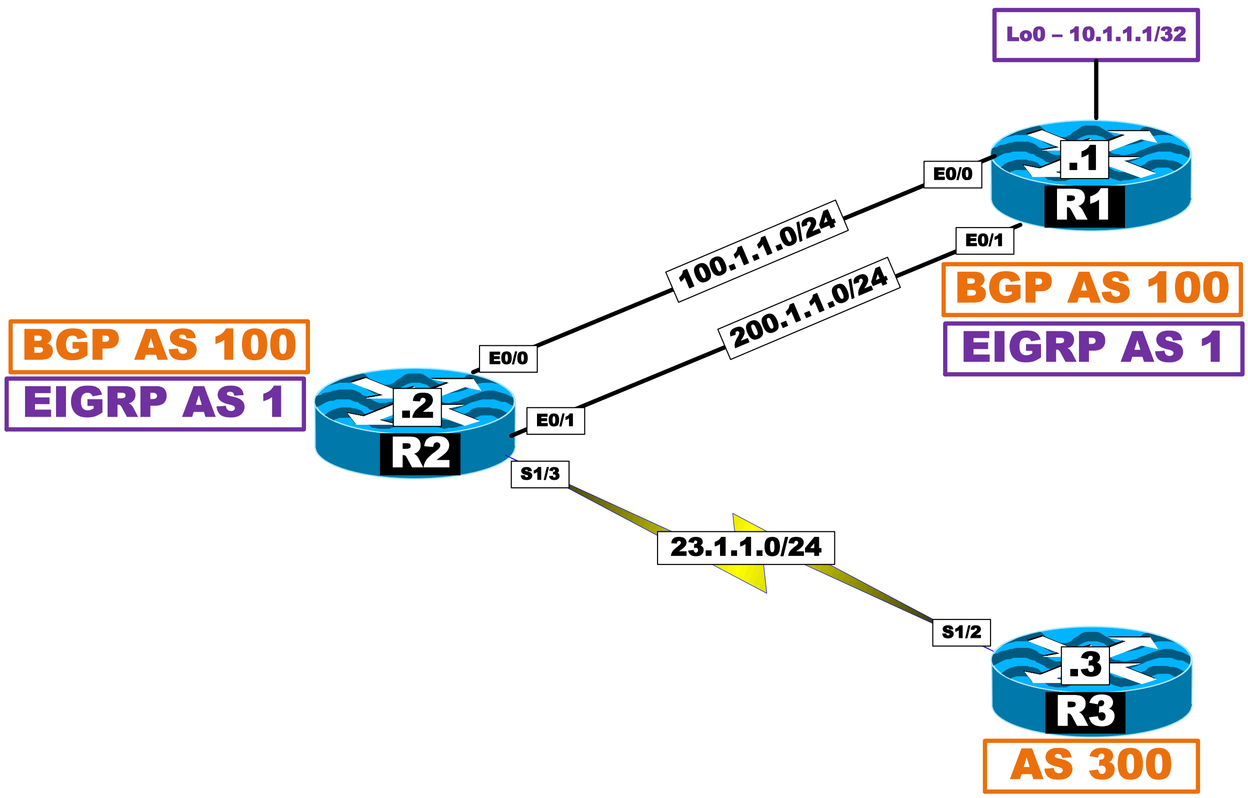
Based on the output of the above show command we can see that the "Automatic tag" is the feature that changed the tag to the two highlighted in yellow and blue. Let's verify the tags in the BGP table:

```
R4#sh ip bgp | b Network

   Network          Next Hop            Metric LocPrf Weight Path
*> 1.1.1.0/24       34.1.1.3                 1         32768 100 i
*> 2.2.2.0/24       34.1.1.3                 1         32768 200 i
*> 34.1.1.0/24      0.0.0.0                  0         32768 ?
```

Task 3

Erase the startup configuration on all routers and switches, delete vlan.dat on all switches and reload all devices before proceeding to the next lab.

Lab 3 – BGP Suppress-inactive



Task 1

Configure the above topology.

On SW1:

```
SW1(config)#int range e0/2-3
SW1(config-if-range)#duple hal
SW1(config-if-range)#swi
SW1(config-if-range)#swi mode acc
SW1(config-if-range)#swi acc v 23
SW1(config-if-range)#spannin portf
```

```
SW1(config-if-range)#no shu
```

On R1:

```
R1(config)#int lo0
R1(config-if)#ip addr 10.1.1.1 255.255.255.255

R1(config-if)#int e0/0
R1(config-if)#ip addr 100.1.1.1 255.255.255.0
R1(config-if)#no shut

R1(config)#int e0/1
R1(config-if)#ip addr 200.1.1.1 255.255.255.0
R1(config-if)#no shu
```

On R2:

```
R2(config)#int e0/0
R2(config-if)#ip addr 100.1.1.2 255.255.255.0
R2(config-if)#no shu

R2(config)#int e0/1
R2(config-if)#ip addr 200.1.1.2 255.255.255.0
R2(config-if)#no shu

R2(config-if)#int s1/3
R2(config-if)#ip addr 23.1.1.2 255.255.255.0
R2(config-if)#no shu
```

On R3:

```
R3(config)#int s1/2
R3(config-if)#ip addr 23.1.1.3 255.255.255.0
R3(config-if)#no shut
```

Task 2

Configure R1 to establish a BGP peer session with R2 using the 200.1.1.2 IP address. R1 should advertise its lo0 to R2 using BGP.
These routers should also run EIGRP on their E0/0 interfaces. R1 should advertise its lo0 interface to R2 using EIGRP.

On R1:

```
R1(config)#router bgp 100
R1(config-router)#netw 10.1.1.1 mask 255.255.255.255
R1(config-router)#neigh 100.1.1.2 remote 100

R1(config-router)#router eigrp 1
R1(config-router)#netw 200.1.1.1 0.0.0.0
R1(config-router)#netw 10.1.1.1 0.0.0.0
```

On R2:

```
R2(config)#router bgp 100
R2(config-router)#neigh 100.1.1.1 remote 100
R2(config-router)#neigh 23.1.1.3 remote 300

R2(config)#router eigrp 1
R2(config-router)#netw 200.1.1.2 0.0.0.0
```

On R3:

```
R3(config-if)#router bgp 300
R3(config-router)#neigh 23.1.1.2 remote 100
```

To verify the configuration:

On R2:

```
R2#sh ip rou eigrp | b Gate
Gateway of last resort is not set

      10.0.0.0/32 is subnetted, 1 subnets
D        10.1.1.1 [90/409600] via 200.1.1.1, 00:00:46, Ethernet0/1

R2#sh ip rou 10.1.1.1
Routing entry for 10.1.1.1/32
  Known via "eigrp 1", distance 90, metric 409600, type internal
  Redistributing via eigrp 1
  Last update from 200.1.1.1 on Ethernet0/1, 00:02:02 ago
  Routing Descriptor Blocks:
  * 200.1.1.1, from 200.1.1.1, 00:02:02 ago, via Ethernet0/1
      Route metric is 409600, traffic share count is 1
      Total delay is 6000 microseconds, minimum bandwidth is 10000 Kbit
      Reliability 255/255, minimum MTU 1500 bytes
      Loading 1/255, Hops 1
```

We can see that R2 took EIGRP over BGP because EIGRP's AD is lower. Let's check the BGP table:

```
R2#sh ip bgp | b Net
     Network          Next Hop            Metric LocPrf Weight Path
 r>i 10.1.1.1/32      100.1.1.1                0    100      0 i
```

We can see that we have a RIB failure, let's see why:

```
R2#sh ip bgp rib-failure
  Network            Next Hop                      RIB-failure   RIB-NH Matches
10.1.1.1/32        100.1.1.1       Higher admin distance              n/a
```

Well, it is due to a higher administrative distance. Did R3 receive the route?

On R3:

```
R3#sh ip bgp | b Net
     Network          Next Hop            Metric LocPrf Weight Path
 *>  10.1.1.1/32      23.1.1.2                               0 100 i
```

Yes, R3 did receive the route, even though the route is inactive. But why?

Let's see the next hop from EIGRP and BGP's perspective:

On R2:

```
R2#sh ip rou ei | b Gate
Gateway of last resort is not set

      10.0.0.0/32 is subnetted, 1 subnets
D        10.1.1.1 [90/409600] via 200.1.1.1, 00:06:36, Ethernet0/1


R2#sh ip bgp | b Net
     Network          Next Hop            Metric LocPrf Weight Path
 r>i 10.1.1.1/32      100.1.1.1                0    100      0 i
```

Even though there is a mismatch in the next hop, R2 advertises 10.1.1.1/32 to R3. Because by default, the inactive routes are advertised.

To suppress the inactive routes:

```
R2(config)#router bgp 100
R2(config-router)#bgp suppress-inactive
```

To verify the configuration:

On R3:

```
R3#sh ip bgp | b Net
R3#
```

Well, this worked.
The “BGP suppress-inactive” command suppressed the rib-failed routes only if the next-hop of BGP and the IGP are different. If they are not different, meaning that they are both pointing to the same IP address then the command has no affect.

This means that if the Next hop of BGP and the RIB entry match, the “BGP suppress-inactive” command will NOT suppress the route. Let’s add a static route that points to the same IP address as the BGP’s next hop and test:

On R2:

```
R2(config)#ip route 10.1.1.1 255.255.255.255 100.1.1.1
```

On R2, and R3:

```
Rx#clear ip bgp *
```

To verify the configuration:

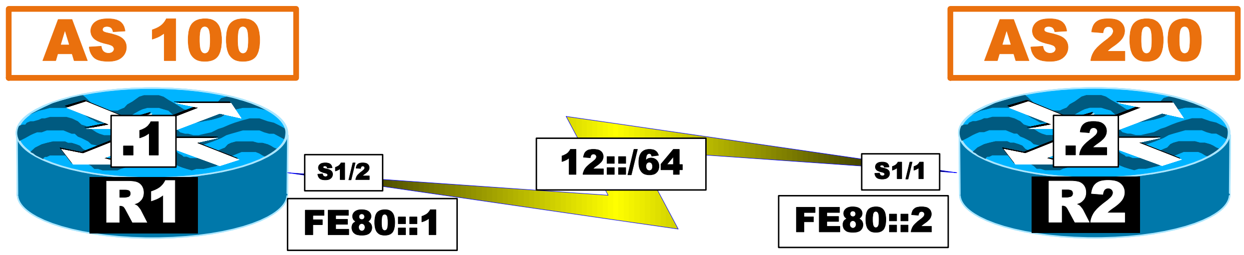
On R3:

```
R3#sh ip bgp | b Net
     Network          Next Hop            Metric LocPrf Weight Path
 *>  10.1.1.1/32      23.1.1.2                                0 100 i
```

Task 3

Erase the startup configuration on all routers and switches and reload them before proceeding to the next lab.

Lab 4 – Configuring IPv6 BGP



Task 1

Configure the routers based on the above topology.

On R1:

```
R1(config)#int s1/2
R1(config-if)#ipv6 enable
R1(config-if)#ipv6 address fe80::1 l
R1(config-if)#ipv6 address 12::1/64
R1(config-if)#no shut
```

On R2:

```
R2(config)#int s1/1
R2(config-if)#ipv6 enable
R2(config-if)#ipv6 address fe80::2 l
R2(config-if)#ipv6 address 12::2/64
R2(config-if)#no shut
```

To verify the configuration:

```
R2#ping 12::1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 12::1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 8/8/9 ms

R2#ping fe80::1
```

```
Output Interface: serial1/1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to FE80::1, timeout is 2 seconds:
Packet sent with a source address of FE80::2%Serial1/1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 8/8/9 ms
```

Task 2

Configure a BGP peer session between R1 and R2 using their IPv6 Global Unicast address.

On R1:

```
R1(config)#ipv6 unicast-routing

R1(config)#router bgp 100
R1(config-router)#no bgp default ipv4-unicast
R1(config-router)#bgp router-id 1.1.1.1
R1(config-router)#neigh 12::2 remote 200

R1(config-router)#address-family ipv6
R1(config-router-af)#neigh 12::2 activate
R1(config-router-af)#netw 1::/64
```

On R2:

```
R2(config)#ipv6 unicast-routing

R2(config)#router bgp 200
R2(config-router)#no bgp default ipv4-unicast
R2(config-router)#bgp router-id 2.2.2.2
R2(config-router)#neigh 12::1 remote 100

R2(config-router)#address-family ipv6
R2(config-router-af)#neigh 12::1 activate
R2(config-router-af)#netw 2::/64
```

You should see the following console message:

```
%BGP-5-ADJCHANGE: neighbor 12::1 Up
```

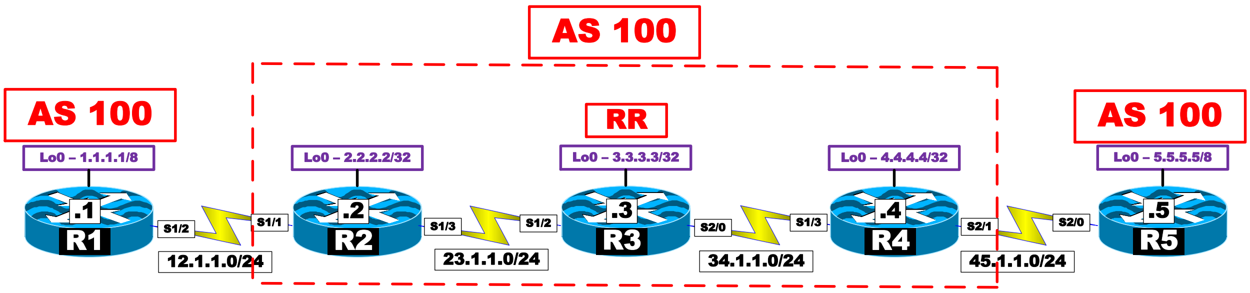
To verify the configuration:

```
R2#sh bgp ipv6 unicast summary | b Neigh

Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
12::1           4   100       7       8        3    0    0 00:03:07        1

R2#sh bgp ipv6 unicast | b Net

     Network          Next Hop            Metric LocPrf Weight Path
 *>  1::/64           12::1                    0             0 100 i
 *>  2::/64           ::                       0         32768 i

R2#sh ipv6 route bgp

IPv6 Routing Table - default - 6 entries
Codes: C - Connected, L - Local, S - Static, U - Per-user Static route
       B - BGP, HA - Home Agent, MR - Mobile Router, R - RIP
       H - NHRP, I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea
       IS - ISIS summary, D - EIGRP, EX - EIGRP external, NM - NEMO
       ND - ND Default, NDp - ND Prefix, DCE - Destination, NDr - Redirect
       O - OSPF Intra, OI - OSPF Inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2, la - LISP alt
       lr - LISP site-registrations, ld - LISP dyn-eid, a - Application
B   1::/64 [20/0]
     via FE80::1, Serial1/1

R2#sh ipv6 route 1::/64

Routing entry for 1::/64
  Known via "bgp 200", distance 20, metric 0, type external
  Route count is 1/1, share count 0
  Routing paths:
    FE80::1, Serial1/1
      MPLS label: nolabel
      Last updated 00:02:51 ago
```

Task 3

Erase the startup configuration on all routers and switches and reload them before proceeding to the next lab.

Lab 5
iBGP peer session between PE-CE



Task 1

Configure the routers in the above topology. DO NOT configure routing protocol.

On R1:

```
R1(config)#int s1/2
R1(config-if)#ip addr 12.1.1.1 255.255.255.0
R1(config-if)#no shu

R1(config-if)#int lo0
R1(config-if)#ip addr 1.1.1.1 255.0.0.0
```

On R2:

```
R2(config)#int lo0
R2(config-if)#ip addr 2.2.2.2 255.255.255.255

R2(config-if)#int s1/1
R2(config-if)#ip addr 12.1.1.2 255.255.255.0
R2(config-if)#no shu

R2(config-if)#int s1/3
R2(config-if)#ip addr 23.1.1.2 255.255.255.0
R2(config-if)#no shu
```

On R3:

```
R3(config)#int lo0
R3(config-if)#ip addr 3.3.3.3 255.255.255.255
```

```
R3(config-if)#int s1/2
R3(config-if)#ip addr 23.1.1.3 255.255.255.0
R3(config-if)#no shu

R3(config-if)#int s2/0
R3(config-if)#ip addr 34.1.1.3 255.255.255.0
R3(config-if)#no shu
```

On R4:

```
R4(config)#int lo0
R4(config-if)#ip addr 4.4.4.4 255.255.255.255

R4(config-if)#int s1/3
R4(config-if)#ip addr 34.1.1.4 255.255.255.0
R4(config-if)#no shu

R4(config-if)#int s2/1
R4(config-if)#ip addr 45.1.1.4 255.255.255.0
R4(config-if)#no shu
```

On R5:

```
R5(config)#int lo0
R5(config-if)#ip addr 5.5.5.5 255.0.0.0

R5(config-if)#int s2/0
R5(config-if)#ip addr 45.1.1.5 255.255.255.0
R5(config-if)#no shu
```

Task 2

Configure OSPF on all links of all routers including their loopback0 interfaces in the providers network.

On R2:

```
R2(config)#router ospf 1
R2(config-router)#netw 2.2.2.2 0.0.0.0 a 0
R2(config-router)#netw 23.1.1.2 0.0.0.0 a 0
```

On R3:

```
R3(config)#router ospf 1
R3(config-router)#netw 0.0.0.0 0.0.0.0 a 0
```

On R4:

```
R4(config)#router ospf 1
R4(config-router)#netw 34.1.1.4 0.0.0.0 a 0
R4(config-router)#netw 4.4.4.4 0.0.0.0 a 0
```

To verify the configuration:

```
R4#sh ip rou ospf | b Gate
Gateway of last resort is not set

      2.0.0.0/32 is subnetted, 1 subnets
O        2.2.2.2 [110/129] via 34.1.1.3, 00:00:37, Serial1/3
      3.0.0.0/32 is subnetted, 1 subnets
O        3.3.3.3 [110/65] via 34.1.1.3, 00:00:37, Serial1/3
      23.0.0.0/24 is subnetted, 1 subnets
O        23.1.1.0 [110/128] via 34.1.1.3, 00:00:37, Serial1/3
```

Task 3

Configure LDP on the routers in the core and set the RID to be based on their loopback0 interface.

On All Routers:

```
Rx(config)#mpls labe proto ldp
Rx(config)#mpls ldp router-id lo0 for
```

On R2:

```
R2(config)#router ospf 1
R2(config-router)#mpls ldp autoconfig area 0
```

On R3:

```
R3(config)#router ospf 1
R3(config-router)#mpls ldp autoconfig area 0
```

On R4:

```
R4(config)#router ospf 1
R4(config-router)#mpls ldp autoconfig area 0
```

To verify the configuration:

```
R4#sh mpls interfaces
Interface              IP            Tunnel   BGP Static Operational
Serial1/3              Yes (ldp)     No       No  No     Yes
```

On R3:

```
R3#sh mpls interfaces
Interface              IP            Tunnel   BGP Static Operational
Serial1/2              Yes (ldp)     No       No  No     Yes
Serial2/0              Yes (ldp)     No       No  No     Yes
```

On R2:

```
R2#sh mpls interfaces
Interface              IP            Tunnel   BGP Static Operational
Serial1/3              Yes (ldp)     No       No  No     Yes
```

Task 4

Configure iBGP peer session on the routers in the core. R3 should be configured as a route reflector.

On R2:

```
R2(config)#router bgp 100
R2(config-router)#neigh 3.3.3.3 remote 100
R2(config-router)#neigh 3.3.3.3 up lo0

R2(config-router)#address-family vpnv4
R2(config-router-af)#neigh 3.3.3.3 act
R2(config-router-af)#neigh 3.3.3.3 send-comm e
```

On R4:

```
R4(config)#router bgp 100
```

```
R4(config-router)#neigh 3.3.3.3 remote 100
R4(config-router)#neigh 3.3.3.3 up lo0

R4(config-router)#address-family vpnv4
R4(config-router-af)#neigh 3.3.3.3 act
R4(config-router-af)#neigh 3.3.3.3 send-community e
```

<u>On R3:</u>

```
R3(config)#router bgp 100
R3(config-router)#neigh 2.2.2.2 remote 100
R3(config-router)#neigh 2.2.2.2 up lo0

R3(config-router)#neigh 4.4.4.4 remote 100
R3(config-router)#neigh 4.4.4.4 up lo0

R3(config-router)#address-family vpnv4
R3(config-router-af)#neigh 2.2.2.2 act
R3(config-router-af)#neigh 2.2.2.2 route-reflector-client

R3(config-router-af)#neigh 4.4.4.4 act
R3(config-router-af)#neigh 4.4.4.4 route-reflector-client
```

<u>To verify the configuration:</u>

```
R3#sh ip bgp vpnv4 all summ | b Nei
Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
2.2.2.2         4   100       6       5        1    0    0 00:00:59        0
4.4.4.4         4   100       6       2        1    0    0 00:00:51        0

On R2:

R2#sh ip bgp vpnv4 all summ | b Nei
Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
3.3.3.3         4   100       7       8        1    0    0 00:03:07        0
```

Task 5

Configure VRFs on the PE routers (R2 and R4). Use an RD, names and route-target value of your choice.

<u>On R2:</u>

```
R2(config)#ip vrf 11
R2(config-vrf)#rd 1:10
R2(config-vrf)#route-target both 1:100

R2(config)#int s1/1
R2(config-if)#ip vrf for 11
R2(config-if)#ip addr 12.1.1.2 255.255.255.0
```

To verify the configuration:

```
R2#ping vrf 11 12.1.1.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 12.1.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 6/8/9 ms
```

On R4:

```
R4(config)#ip vrf 55
R4(config-vrf)#rd 1:50
R4(config-vrf)#route-target both 1:100

R4(config)#int s2/1
R4(config-if)#ip vrf for 55
R4(config-if)#ip addr 45.1.1.4 255.255.255.0
```

To verify the configuration:

```
R4#ping vrf 55 45.1.1.5
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 45.1.1.5, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 9/9/9 ms
```

Task 6

Configure an iBGP peer session between the customer routers (R1 and R5) and the PE routers (R2 and R4). The customer routers should advertise their lo0.

On R1:

```
R1(config)#router bgp 100
R1(config-router)#neigh 12.1.1.2 remote 100
R1(config-router)#netw 1.0.0.0
```

On R2:

```
R2(config)#router bgp 100
R2(config-router)#address-family ipv4 vrf 11
R2(config-router-af)#neigh 12.1.1.1 remote 100
```

On R5:

```
R5(config)#router bgp 100
R5(config-router)#neigh 45.1.1.4 remote 100
R5(config-router)#netw 5.0.0.0
```

On R4:

```
R4(config)#router bgp 100
R4(config-router)#address-family ipv4 vrf 55
R4(config-router-af)#neigh 45.1.1.5 remote 100
```

To verify the configuration:

On R1:

```
R1#sh ip bgp summ | b Nei
Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
12.1.1.2        4   100       6       7        2    0    0 00:01:57        0
```

On R2:

```
R2#sh ip route vrf 11 bgp | b Gate
Gateway of last resort is not set

B     1.0.0.0/8 [200/0] via 12.1.1.1, 00:02:48

R2#sh ip bgp vpnv4 all | b Net
     Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 1:10 (default for vrf 11)
 *>i 1.0.0.0          12.1.1.1                 0    100      0 i

R2#sh ip bgp vpnv4 all summ | b Nei
Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
3.3.3.3         4   100      31      33        2    0    0 00:24:44        0
12.1.1.1        4   100       9       8        2    0    0 00:04:00        1
```

On R3:

```
R3#sh ip bgp vpnv4 all
R3#
```

On R4:

```
R4#sh ip bgp vpnv4 all | b Net
     Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 1:50 (default for vrf 55)
 *>i 5.0.0.0          45.1.1.5                 0    100      0 i
```

Why R3 (The RR) does not see any of the BGP VPNv4 routes? The reason is because the customer is also running in the same AS, if the customer is also running in the same AS, then this feature must be enabled. Let's enable and verify:

On R2:

```
R2(config)#router bgp 100

R2(config-router)#address-family ipv4 vrf 11
R2(config-router-af)#neigh 12.1.1.1 internal-vpn-client
```

To verify the configuration:

```
R2#sh ip bgp vpnv4 all | b Net
     Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 1:10 (default for vrf 11)
 *>i 1.0.0.0          12.1.1.1                 0    100      0 i
 * i                  12.1.1.1                                 0 i
```

On R3:

```
R3#sh ip bgp vpnv4 all | b Net
     Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 1:10
 *>i 1.0.0.0          2.2.2.2                       100      0 i
```

On R4:

```
R4#sh ip bgp vpnv4 all | b Net
     Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 1:10
 *>i 1.0.0.0          2.2.2.2                       100      0 i
Route Distinguisher: 1:50 (default for vrf 55)
```

```
 *>i 1.0.0.0          2.2.2.2                  0    100      0 i
 *>i 5.0.0.0          45.1.1.5                 0    100      0 i
```

On R5:

```
R5#sh ip bgp | b Net
     Network          Next Hop            Metric LocPrf Weight Path
 *>  5.0.0.0          0.0.0.0                  0         32768 i
```

Now, we have to enable the same feature on R4 for VRF 55.

On R4:

```
R4(config)#router bgp 100

R4(config-router)#address-family ipv4 vrf 55
R4(config-router-af)#neigh 45.1.1.5 internal-vpn-client
```

To verify the configuration:

On R1:

```
R1#sh ip bgp | b Net
     Network          Next Hop            Metric LocPrf Weight Path
 *>  1.0.0.0          0.0.0.0                  0         32768 i
 *>i 5.0.0.0          12.1.1.2                 0    100      0 i
```

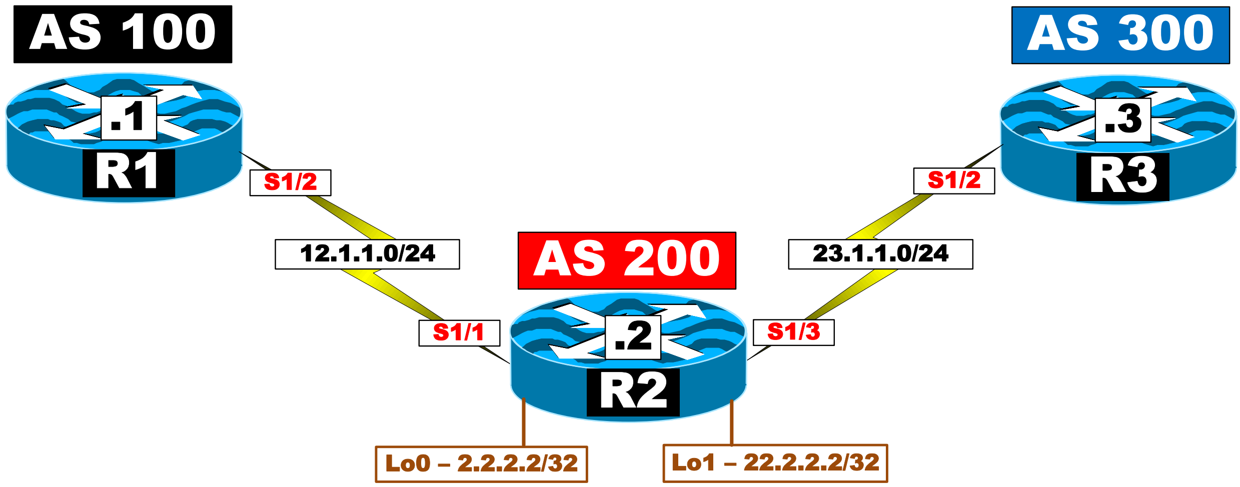
On R5:

```
R5#sh ip bgp | b Net
     Network          Next Hop            Metric LocPrf Weight Path
 *>i 1.0.0.0          45.1.1.4                 0    100      0 i
 *>  5.0.0.0          0.0.0.0                  0         32768 i

R5#ping 1.1.1.1 sou lo0
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 1.1.1.1, timeout is 2 seconds:
Packet sent with a source address of 5.5.5.5
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 32/34/36 ms
```

Task 7

Erase the startup configuration on all routers and switches and reload them before proceeding to the next lab.

Lab 6 – BGP Conditional Route Advertisement



R2 is receiving a default route from both AS 100 and AS 300. R2's connection to R3 is much faster than R2's connection to R1, therefore, AS 300 is the primary and AS 100 is the backup.

R2 needs the return traffic through R3, but this has to happen 100 percent of the time, in order to achieve this, R2 will ONLY advertise its loopback interfaces to R3 if R3 is up. If R3 is NOT up, then and only then R2 will advertise the two loopback interfaces to R1.

On R1:

```
R1(config)#router bgp 100
R1(config-router)#neigh 12.1.1.2 remot 200
R1(config-router)#neigh 12.1.1.2 default-originate
```

On R3:

```
R3(config)#router bgp 300
R3(config-router)#neigh 23.1.1.2 remot 200
R3(config-router)#neigh 23.1.1.2 default-originate
```

On R2:

```
R2(config)#router bgp 200
R2(config-router)#neigh 12.1.1.1 remot 100
R2(config-router)#neigh 23.1.1.3 remot 300
R2(config-router)#netw 2.2.2.2 mask 255.255.255.255
R2(config-router)#netw 22.2.2.2 mask 255.255.255.255
```

To verify the configuration:

On R1:

```
R1#sh ip bgp | b Netw
     Network          Next Hop            Metric LocPrf Weight Path
     0.0.0.0          0.0.0.0                                  0 i
 *>  2.2.2.2/32       12.1.1.2                 0             0 200 i
 *>  22.2.2.2/32      12.1.1.2                 0             0 200 i
```

On R3:

```
R3#sh ip bgp | b Netw
     Network          Next Hop            Metric LocPrf Weight Path
 *>  0.0.0.0          23.1.1.2                               0 200 100 i
                      0.0.0.0                                  0 i
 *>  2.2.2.2/32       23.1.1.2                 0             0 200 i
 *>  22.2.2.2/32      23.1.1.2                 0             0 200 i
```

Let's implement the scenario:

The first step is to identify the two networks on R2:

```
R2(config)#access-list 1 permit host 2.2.2.2
R2(config)#access-list 1 permit host 22.2.2.2
```

The second step is to advertise these two routes ONLY to the two ASes:

```
R2(config)#router bgp 200
R2(config-router)#neigh 12.1.1.1 distribute-list 1 out
R2(config-router)#neigh 23.1.1.3 distribute-list 1 out
```

The third step is to identify the ASN of R3, the primary AS:

```
R2(config)#ip as-path access-list 1 permit ^300$
```

The fourth step is to identify the default route:

```
R2(config)#access-list 2 permit 0.0.0.0 255.255.255.255
```

The fifth step is to configure two route-maps. One route-map will be referenced by the exist-map, and the second route-map will be referenced by the non-exist-map:

```
R2(config)#route-map ADVERTISE permit 10
R2(config-route-map)#match ip addr 1
```

```
R2(config)#route-map Non-Exist-Map permit 10
R2(config-route-map)#match as-path 1
R2(config-route-map)#match ip addr 2
```

In the above configuration two route-maps we are configured, saying Advertise what I have referenced by ACL 1, which are the two loopback interfaces (2.2.2.2/32, and 22.2.2.2/32). If what is matched in the non-exist-map does not exist, meaning that we are no longer receiving a default route from AS 300.

The Sixth and the final step is to apply the condition in BGP to R1 in AS 100:

```
R2(config)#router bgp 200
R2(config-router)#neigh 12.1.1.1 advertise-map ADVERTISE non-
exist-map Non-Exist-Map
```

To test the configuration:

On R1:

```
R1#sh ip bgp | b Netw
     Network          Next Hop            Metric LocPrf Weight Path
     0.0.0.0          0.0.0.0                                 0 i
```

On R3:

```
R3(config)#int s1/2
R3(config-if)#shut
```

On R2:

```
R2#sh ip bgp neigh 12.1.1.1 advertised-routes

Total number of prefixes 0
```

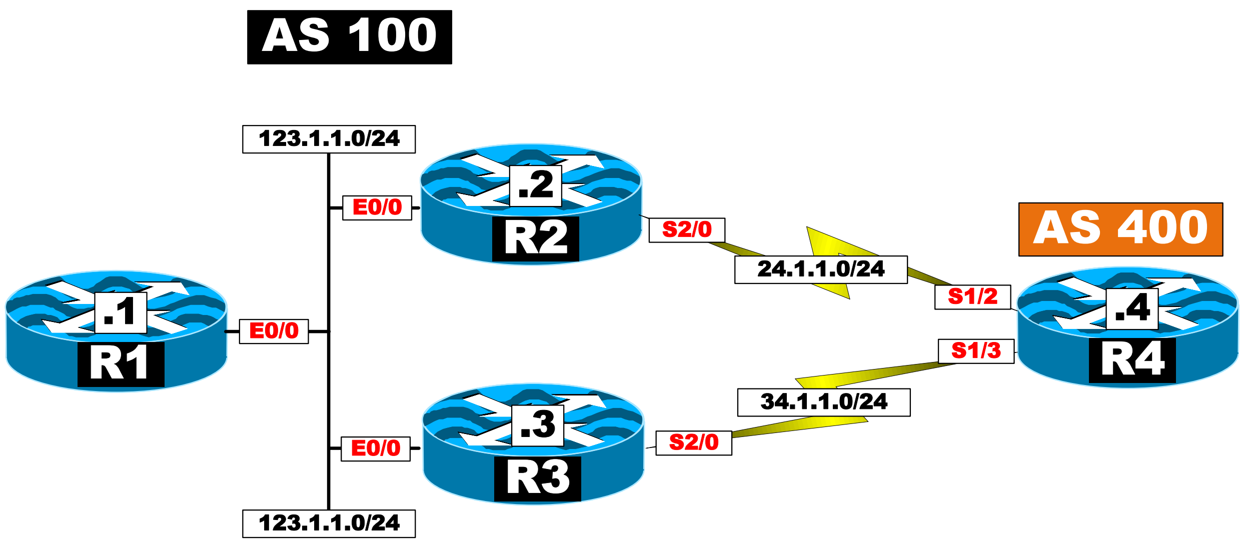
On R1:

```
R1#sh ip bgp | b Netw
     Network          Next Hop            Metric LocPrf Weight Path
     0.0.0.0          0.0.0.0                                 0 i
 *>  2.2.2.2/32       12.1.1.2                 0              0 200 i
 *>  22.2.2.2/32      12.1.1.2                 0              0 200 i
```

We can also do the following show command:

```
Show ip bgp neighbor 12.1.1.1 | i Condition-map
```

Lab 7 – BGP Unequal-cost Load sharing



The bandwidth on the link between R2 – R4 is 10240, and the bandwidth of the link between R3 – R4 is 30720.

R4 should perform unequal cost load sharing to reach the loopback interface of R1.

On R1:

```
R1#sh run int e0/0 | b interface
interface Ethernet0/0
 ip address 123.1.1.1 255.255.255.0
end
```

On R2:

```
R2#sh run int e0/0 | b interface
interface Ethernet0/0
 ip address 123.1.1.2 255.255.255.0
end
```

```
R2#sh run int s2/0 | b interface
interface Serial2/0
 bandwidth 10240
 ip address 24.1.1.2 255.255.255.0
 serial restart-delay 0
end
```

On R3:

```
R3#sh run int e0/0 | b interface
interface Ethernet0/0
 ip address 123.1.1.3 255.255.255.0
end

R3#sh run int s2/0 | b interface
interface Serial2/0
 bandwidth 30720
 ip address 34.1.1.3 255.255.255.0
 serial restart-delay 0
end
```

On R4:

```
R4#sh run int s1/2 | b interface
interface Serial1/2
 bandwidth 10240
 ip address 24.1.1.4 255.255.255.0
 serial restart-delay 0
end

R4#sh run int s1/3 | b interface
interface Serial1/3
 bandwidth 30720
 ip address 34.1.1.4 255.255.255.0
 serial restart-delay 0
end
```

Now, let's configure BGP:

On R1:

```
R1#sh run | s router bgp
router bgp 100
 bgp log-neighbor-changes
```

```
 network 1.1.1.1 mask 255.255.255.255
 neighbor 123.1.1.2 remote-as 100
 neighbor 123.1.1.3 remote-as 100
```

On R2:

```
router bgp 100
 bgp log-neighbor-changes
 neighbor 24.1.1.4 remote-as 400
 neighbor 123.1.1.1 remote-as 100
 neighbor 123.1.1.1 next-hop-self
 neighbor 123.1.1.3 remote-as 100
 neighbor 123.1.1.3 next-hop-self
```

On R3:

```
router bgp 100
 bgp log-neighbor-changes
 neighbor 34.1.1.4 remote-as 400
 neighbor 123.1.1.1 remote-as 100
 neighbor 123.1.1.1 next-hop-self
 neighbor 123.1.1.2 remote-as 100
 neighbor 123.1.1.2 next-hop-self
```

On R4:

```
router bgp 400
 bgp log-neighbor-changes
 bgp dmzlink-bw
 network 4.4.4.4 mask 255.255.255.255
 neighbor 24.1.1.2 remote-as 100
 neighbor 24.1.1.2 dmzlink-bw
 neighbor 34.1.1.3 remote-as 100
 neighbor 34.1.1.3 dmzlink-bw
 maximum-paths 2
```

To verify the configuration:

On R4:

```
R4#sh ip bgp 1.1.1.1

BGP routing table entry for 1.1.1.1/32, version 4

Paths: (2 available, best #2, table default)
```

```
Multipath: eBGP
Flag: 0x800
  Advertised to update-groups: (Pending Update Generation)
     2
  Refresh Epoch 1
  100
    34.1.1.3 from 34.1.1.3 (123.1.1.3)
      Origin IGP, localpref 100, valid, external, multipath(oldest)
      DMZ-Link Bw 3840 kbytes
      rx pathid: 0, tx pathid: 0
  Refresh Epoch 1
  100
    24.1.1.2 from 24.1.1.2 (123.1.1.2)
      Origin IGP, localpref 100, valid, external, multipath, best
      DMZ-Link Bw 1280 kbytes
      rx pathid: 0, tx pathid: 0x0

R4#sh ip rou 1.1.1.1

Routing entry for 1.1.1.1/32
  Known via "bgp 400", distance 20, metric 0
  Tag 100, type external
  Last update from 24.1.1.2 00:01:02 ago
  Routing Descriptor Blocks:
  * 34.1.1.3, from 34.1.1.3, 00:01:02 ago
      Route metric is 0, traffic share count is 3
      AS Hops 1
      Route tag 100
      MPLS label: none
    24.1.1.2, from 24.1.1.2, 00:01:02 ago
      Route metric is 0, traffic share count is 1
      AS Hops 1
      Route tag 100
      MPLS label: none

R4#sh ip bgp | b Network

     Network          Next Hop            Metric LocPrf Weight Path
 *>  1.1.1.1/32       24.1.1.2                               0 100 i
 *m                   34.1.1.3                               0 100 i
 *>  4.4.4.4/32       0.0.0.0                  0         32768 i
 *m  11.1.1.1/32      24.1.1.2                               0 100 i
 *>                   34.1.1.3                               0 100 i
```