

# BGP Communities



- BGP Communities are a means of tagging routes to ensure consistent filtering or route selection policy
- A community can be set on a group of prefixes to indicate that they should be treated the same way
- A BGP router can add Communities to routes in incoming and outgoing Updates or when redistributing
- A BGP router can filter routes in incoming or outgoing updates or select preferred routes based on communities

# Private BGP Communities



- As well as the 'Well Known' Internet, No-Advertise, No-Export and Local-AS communities, private communities can also be used
- These can be used within as AS, or between Autonomous Systems when their use is agreed on (what Community Number to use, and what action to take on it)

# Community Number Format



- A BGP community is a 32-bit number
- It can be displayed as either:
  - Full 32-bit number (0 - 4,294,967,295)
  - **'New Format'**: two 16-bit numbers (0-65535):(0-65535)
- The full number is shown by default in IOS output. To display in New Format:

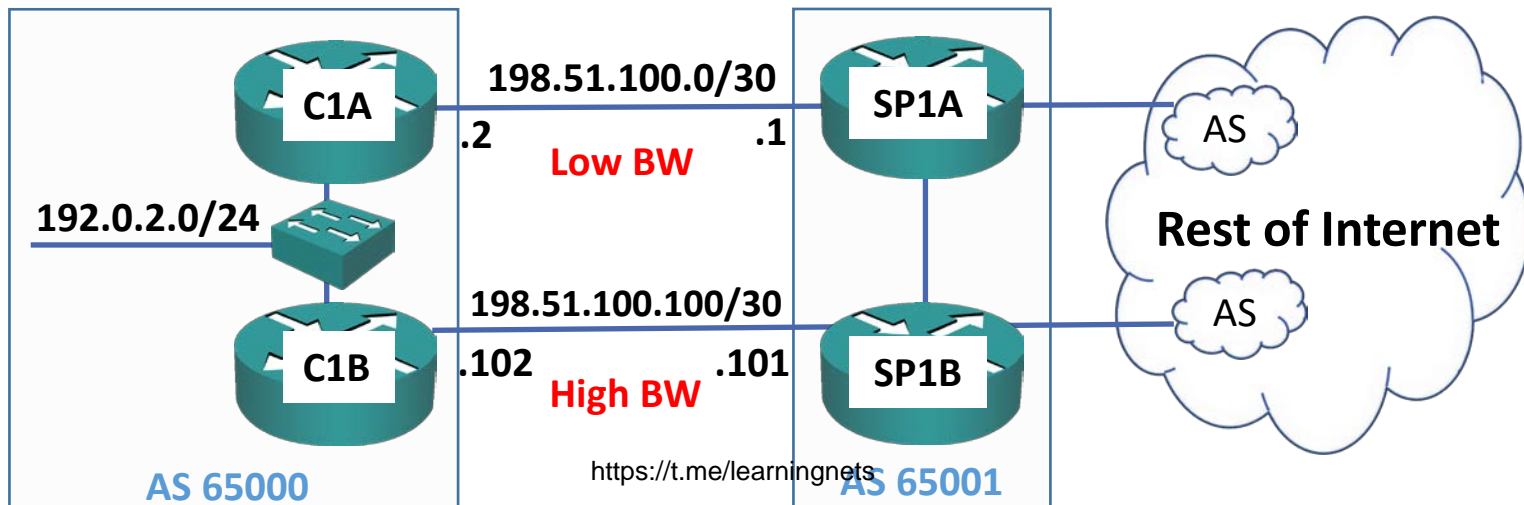
```
R1(config)# ip bgp-community new-format
```

# Community Number Format (Cont.)

- When New Format numbers are used, the first portion is typically the AS number of the organisation managing the community, and the 2<sup>nd</sup> portion is a unique number for each community.
- For example (communities managed by AS 65001):
  - 65001:1200 - prefixes to be assigned Local Preference 200 by neighbor AS
  - 65001:1300 - prefixes to be assigned Local Preference 300 by neighbor AS

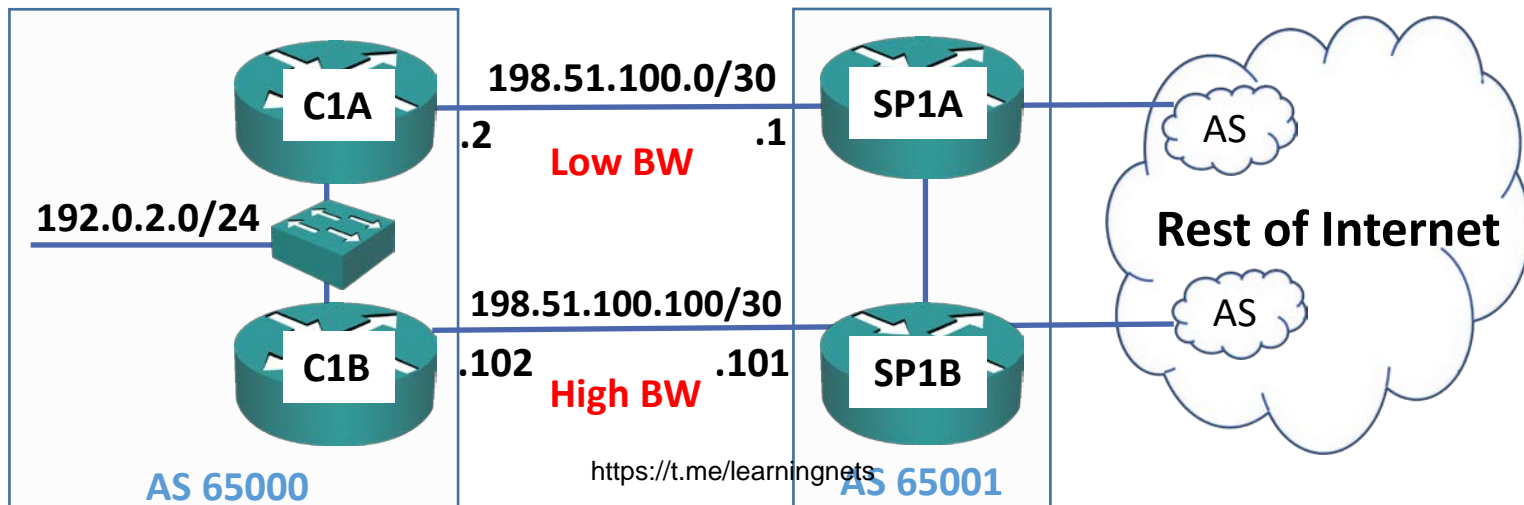
# Inbound Path Selection Example

- In the example below, AS 65000 would prefer AS 65001 to send traffic to its 192.0.2.0/24 network over the high bandwidth link. The customer could ask the Service Provider to set a higher Local Preference for the link
- Internet Service Providers use set policies and configuration templates
- They typically avoid custom configurations per customer because it would add considerable complexity to their administration and troubleshooting
- One option is for the customer to use MED to influence the incoming path



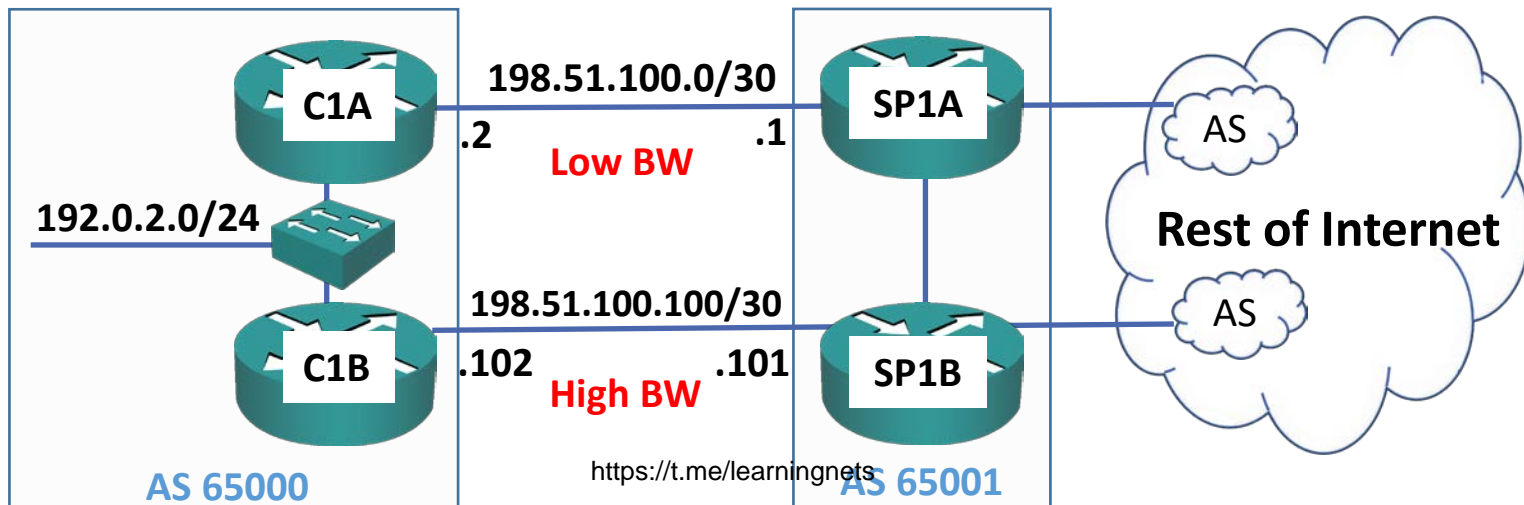
# Communities for Inbound Path Selection

- SP1 can tell their customers to use Community 65001:1200 (for example) to set a preferred path for traffic to their networks
- SP1 will set a higher Local Preference for routes with the Community set
- This allows a standard policy for all customers to influence inbound path selection to their own networks
- Customer 1 wants incoming traffic to the 192.0.2.0/24 network to be received on the high bandwidth link



# Inbound Path Selection

- With all other factors being equal, SP1A and SP1B will send traffic to 192.0.2.0/24 over their direct eBGP links to AS 65000:
  - Prefer highest weight (local to router).
  - Prefer highest local preference (global within AS).
  - Prefer route originated by the local router ('network' command or redistribution).
  - Prefer shortest AS path.
  - Prefer lowest origin code: IGP ('network') < EGP (legacy) < incomplete (redistributed).
  - Prefer lowest MED (exchanged between autonomous systems).
  - Prefer EBGP path over IBGP path.

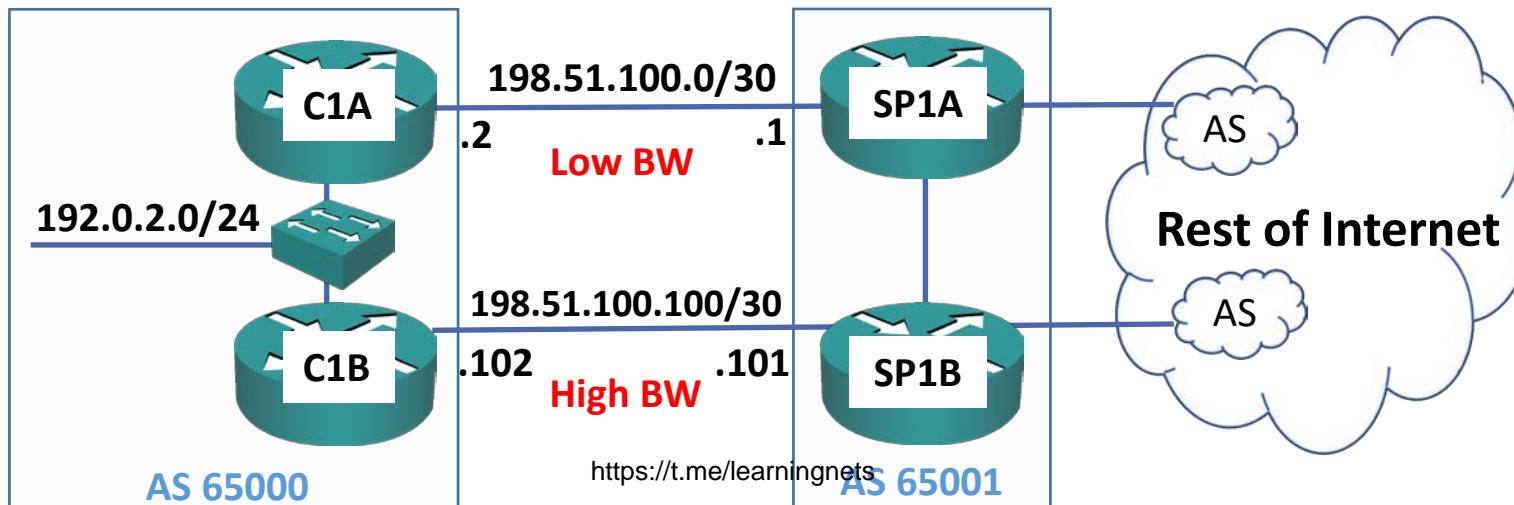


# Verification – show ip bgp (Before Policy)

```
SP1A#sh ip bgp
```

```
      Network                Next Hop                Metric LocPrf Weight Path
*i 192.0.2.0                 192.168.0.2            0      100      0 65000 i (SP1B)
*>                            198.51.100.2          0      100      0 65000 i (C1A)
```

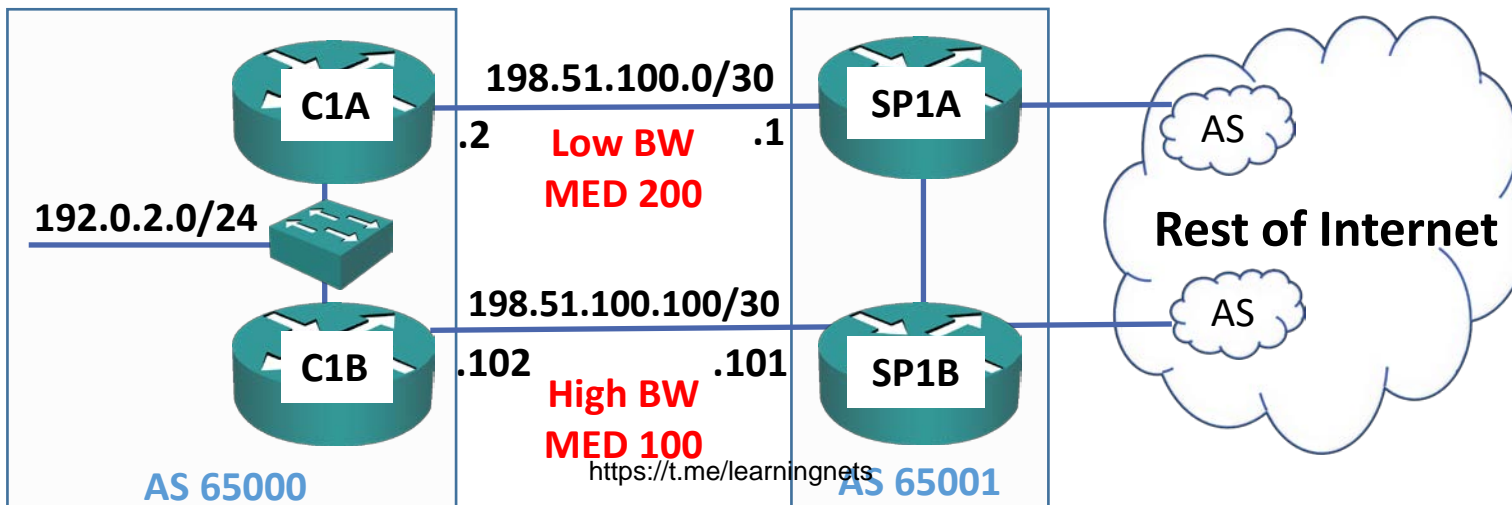
- Local Preference is sent to iBGP neighbors only. The eBGP route received from C1A shows a blank Local Preference, and is treated as the default value of 100



# Verification – show ip bgp (Before Policy)

```
SP1B#sh ip bgp
```

```
      Network                Next Hop           Metric LocPrf  Weight  Path
*i  192.0.2.0                192.168.0.1       0      100      0 65000  i (SP1A)
*> 198.51.100.102           198.51.100.102   0      0        0 65000  i (C1B)
```



# Community Propagation



- Communities are stripped in all outgoing updates by default
- To send communities:

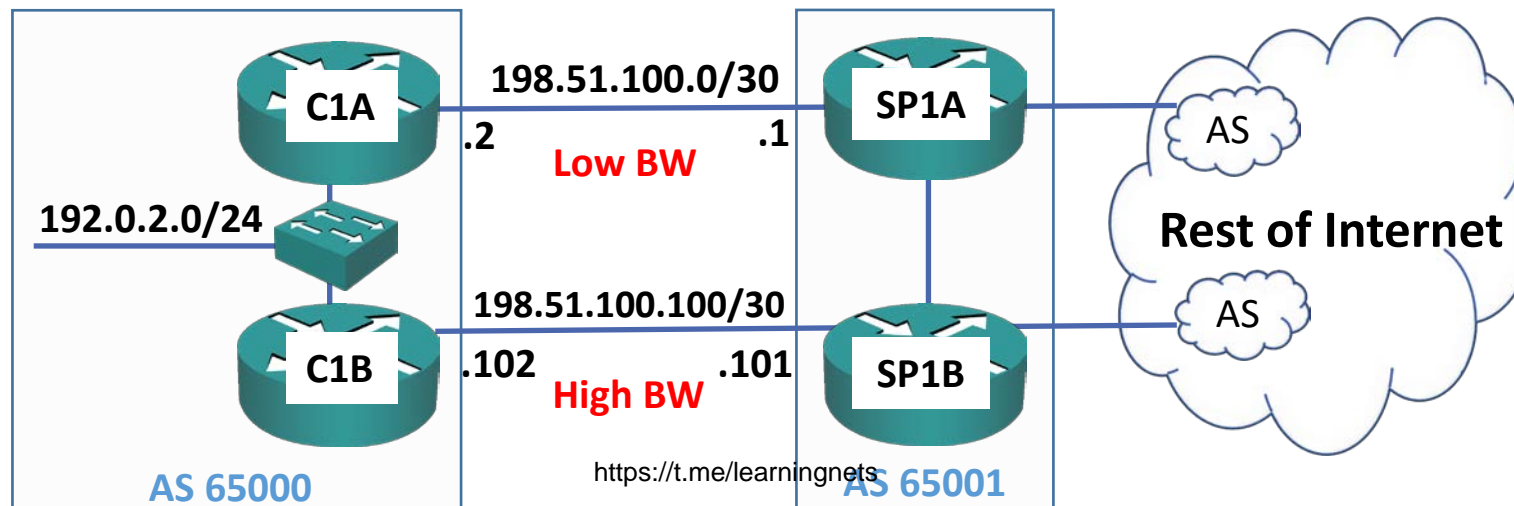
```
R1(config)# router bgp 65001
```

```
R1(config)# neighbor 1.1.1.1 send-community
```

# Enterprise Configuration



```
C1B(config)# route-map HIGH_BW permit 10
C1B(config-route-map)# set community 65001:1200
C1B(config)# router bgp 21
C1B(config-router)# neighbor 198.51.100.101 send-community
C1B(config-router)# neighbor 198.51.100.101 route-map HIGH_BW out
```



# Service Provider Configuration (All Edge Routers)



```
SP1B(config)#ip community-list 1 permit 65001:1200
```

```
SP1B(config)#route-map LOCAL_PREF permit 10
```

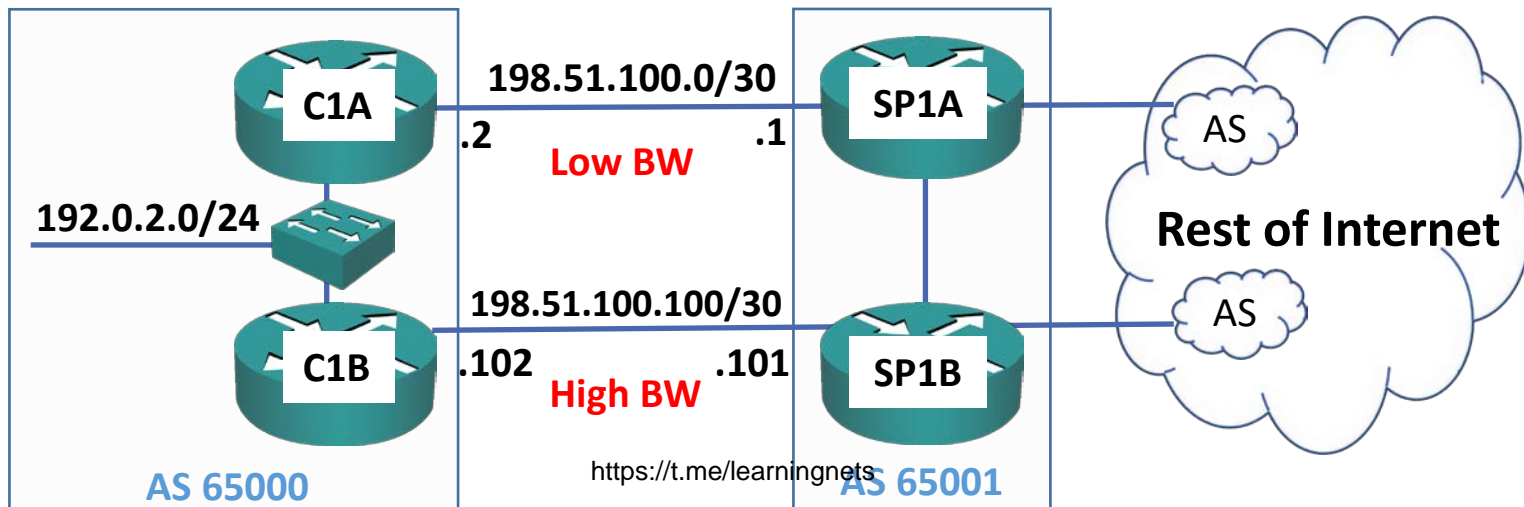
```
SP1B(config-route-map)#match community 1
```

```
SP1B(config-route-map)#set local-preference 200
```

```
SP1B(config-route-map)#route-map LOCAL_PREF permit 20
```

```
SP1B(config)#router bgp 65001
```

```
SP1B(config-router)#neighbor 198.51.100.102 route-map LOCAL_PREF in
```

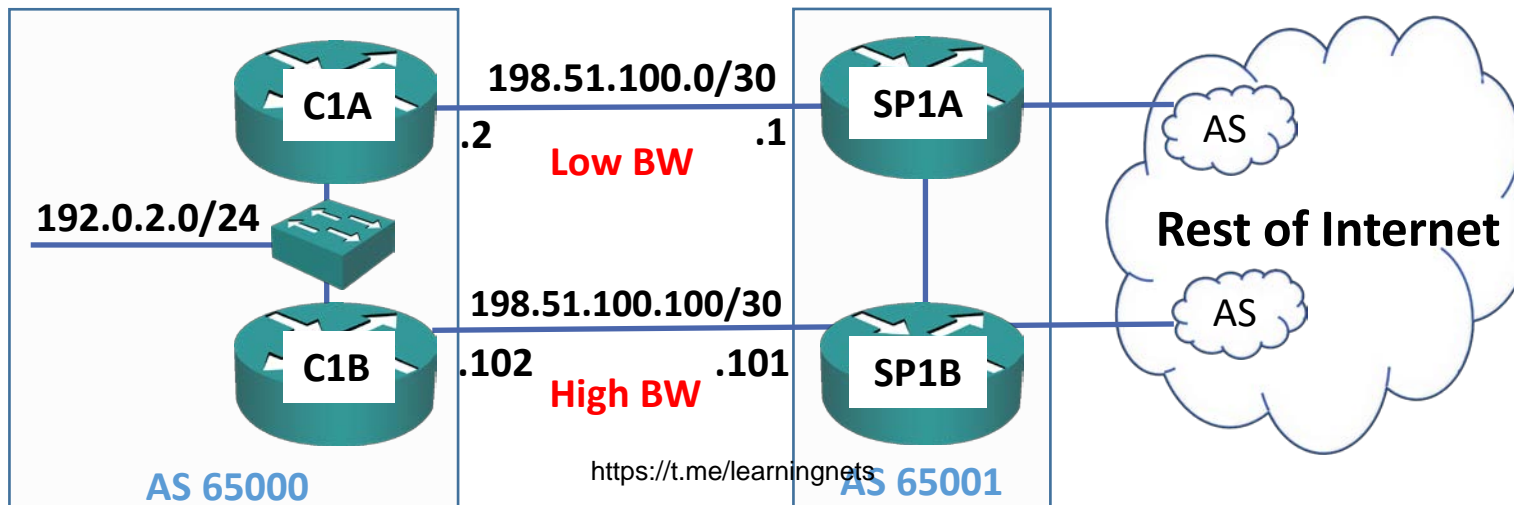


# Verification – show ip bgp (Before Policy)

```
SP1A#sh ip bgp
```

```
      Network                Next Hop                Metric LocPrf Weight Path
*i 192.0.2.0                 192.168.0.2            0     100     0 65000 i (SP1B)
*>                            198.51.100.2           0     100     0 65000 i (C1A)
```

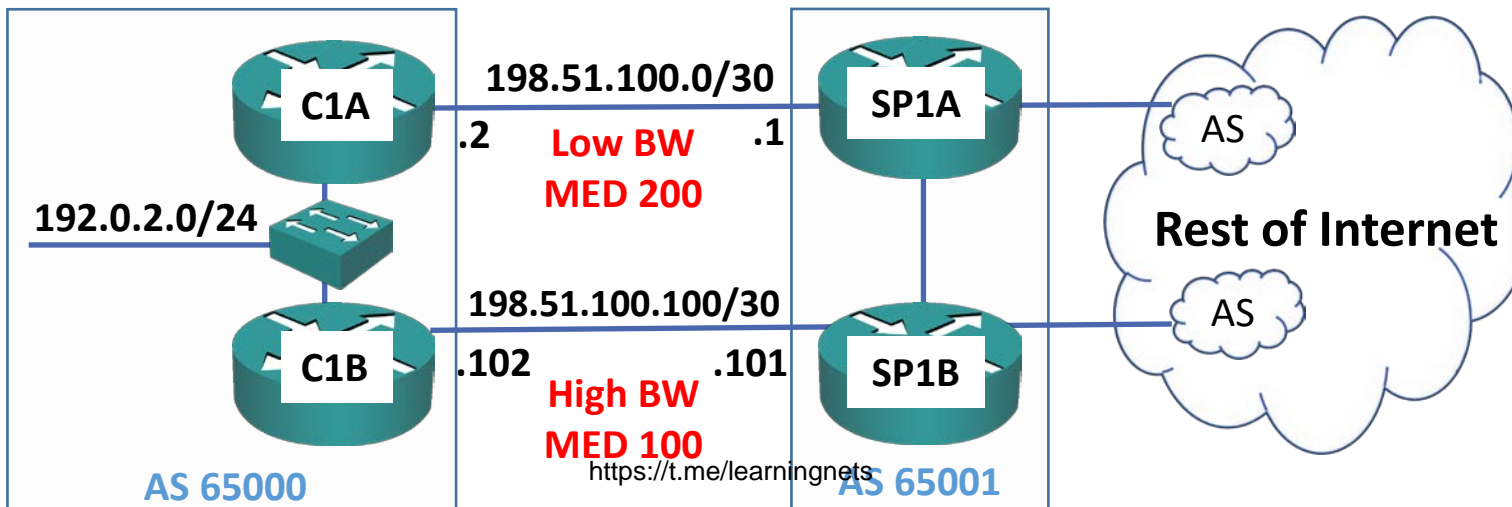
- Local Preference is sent to iBGP neighbors only. The eBGP route received from C1A shows a blank Local Preference, and is treated as the default value of 100



# Verification – show ip bgp (Before Policy)

```
SP1B#sh ip bgp
```

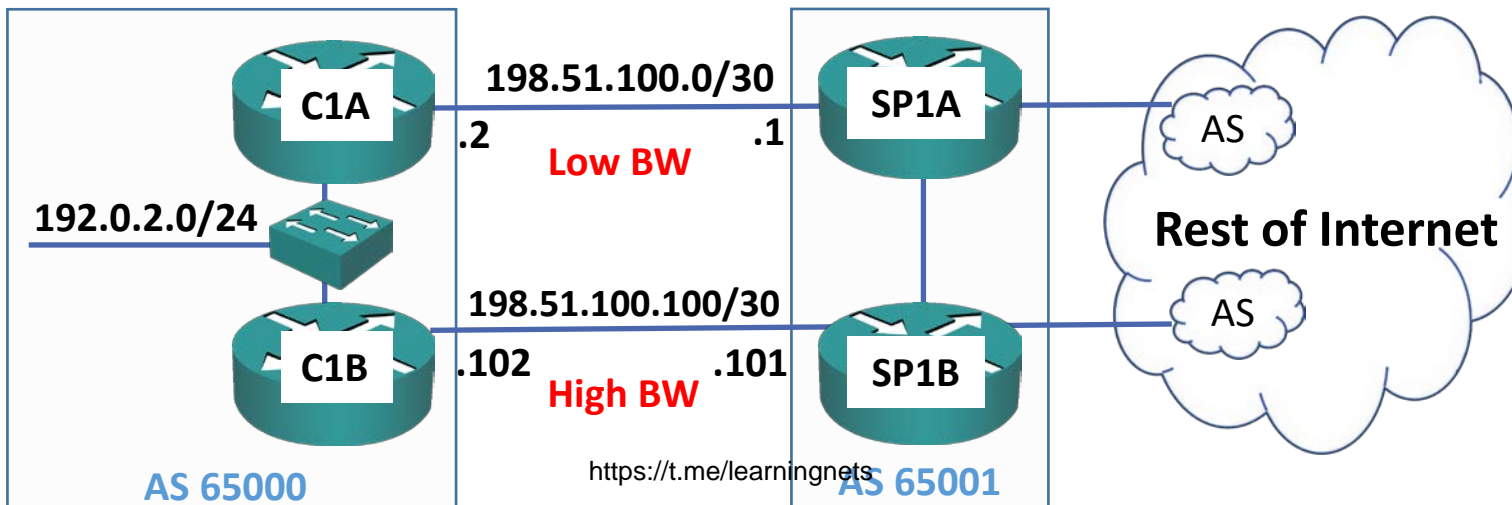
```
      Network                Next Hop           Metric LocPrf  Weight  Path
 *i  192.0.2.0              192.168.0.1       0      100     0 65000  i (SP1A)
 *> 198.51.100.102         198.51.100.102   0      0      0 65000  i (C1B)
```



# Verification – show ip bgp (After Policy)

```
SP1A#sh ip bgp
```

Network	Next Hop	Metric	LocPrf	Weight	Path
*i> 192.0.2.0	192.168.0.2	0	200	0	65000 i (SP1B)
*	198.51.100.2	0		0	65000 i (C1A)

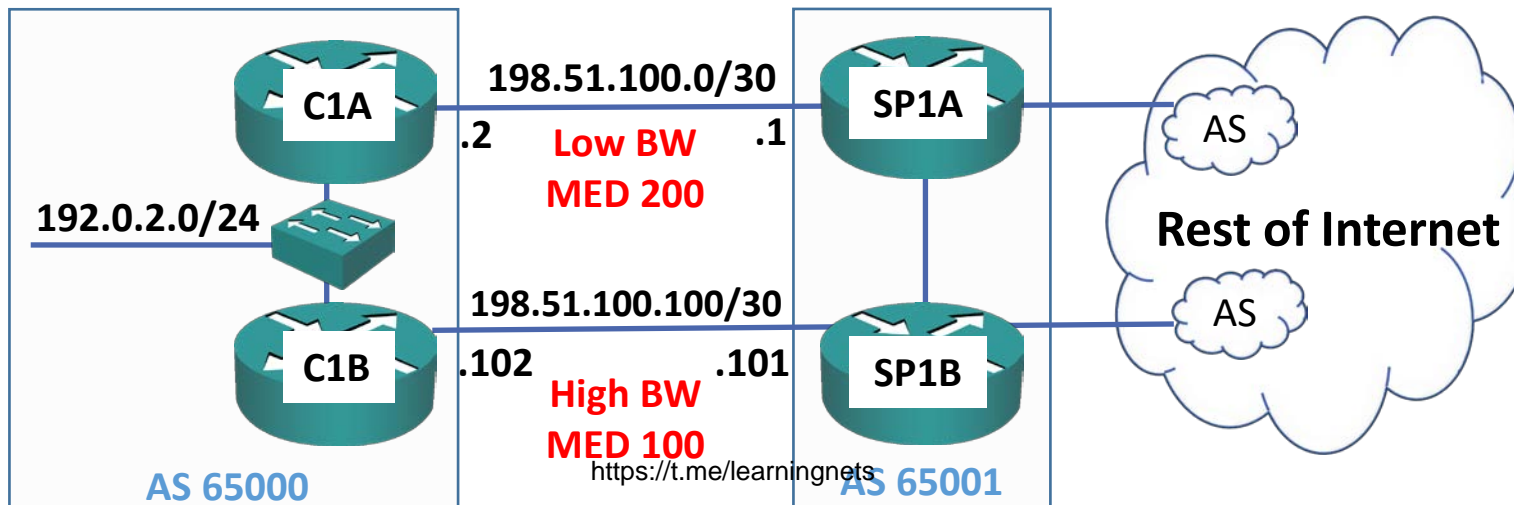


# Verification – show ip bgp (After Policy)

```
SP1B#sh ip bgp
```

```
      Network          Next Hop          Metric LocPrf  Weight  Path
* >  192.0.2.0        198.51.100.102      0       200       0 65000 i (C1B)
```

- SP1A's preferred path to the 192.0.2.0/24 network is via SP1B
- Loop prevention stops the path from being advertised back to SP1B by SP1A



# Setting Communities



- Communities are set via Route Map
- You can add multiple communities to the same route, for example 2 different communities to indicate it is located in North America and it should have Local Preference 200
- By default, setting a community value overwrites any previously set communities
- Use the 'additive' keyword to retain existing communities and set multiple communities

```
R1(config)# ip prefix-list LIST1 permit 198.51.100.220/30
```

```
R1(config)# route-map DEMO permit 10
```

```
R1(config-route-map)# match ip address prefix-list LIST1
```

```
R1(config-route-map)# set community 65001:11 65001:52 additive
```

```
R1(config)# route-map DEMO permit 20
```

```
R1(config)# router bgp 65001
```

```
R1(config-router)# neighbor 192.168.0.2 send-community
```

```
R1(config-router)# neighbor 203.0.113.1 route-map DEMO in
```

# Matching Communities

- Communities are matched via Community List and Route Map
- Community Lists match on Community Number or 'Well Known' Community name
- They are evaluated from top to bottom like an ACL
- The example below would match a route with communities 65100:70 or 65100:80, and any other communities attached (for example a route with communities 65100:80 and 65001:105 would match)

```
R5(config)#ip community-list 1 permit 65100:70
```

```
R5(config)#ip community-list 1 permit 65100:80
```

```
R5(config)#route-map LOCAL_PREF permit 10
```

```
R5(config-route-map)#match community 1
```

```
R5(config-route-map)#set local-preference 200
```

```
R5(config-route-map)#route-map LOCAL_PREF permit 20
```

```
R5(config)#router bgp 65100
```

```
R5(config-router)#neighbor 192.0.2.10 route-map LOCAL_PREF in
```

# Matching Communities



- If there are multiple communities in a line, the route must have all of them to match
- With the 'exact' option, all communities attached to the route have to match
- The example below would match a route with communities '65100:15, 65100:20 and 65100:30', or with '65100:40' only, **but not** '65100:10, 65100:20 and 65100:30', or '65100:40 and 65100:50'

```
R5(config)#ip community-list 1 deny 65100:10
R5(config)#ip community-list 1 permit 65100:20 65100:30
R5(config)#ip community-list 1 permit 65100:40 exact
R5(config)#route-map LOCAL_PREF permit 10
R5(config-route-map)#match community 1
R5(config-route-map)#set local-preference 200
R5(config-route-map)#route-map LOCAL_PREF permit 20
R5(config)#router bgp 65100
R5(config-router)#neighbor 192.0.2.10 route-map LOCAL_PREF in
```

# Expanded and Named Community Lists

- Standard community lists use standard text
- Expanded community lists match based on Regular Expressions
- Standard community lists use numbers 1 – 99
- Expanded community lists use numbers 100 – 199
- Named community lists can also be used, with no limit in number

```
R1(config)#ip community-list ?  
  <1-99>      Community list number (standard)  
  <100-500>   Community list number (expanded)  
  expanded   Add an expanded community-list entry  
  standard   Add a standard community-list entry
```

# Verification – show ip bgp <prefix>



- Shows any communities attached to a specific route



```
R1#show ip bgp 203.0.113.0
BGP routing table entry for 203.0.113.0/24, version 23
Paths: (1 available, best #1, table default)
  Advertised to update-groups:
    2          3          5
Refresh Epoch 2
65012
  198.51.100.6 from 198.51.100.6 (192.168.1.2)
    Origin IGP, metric 0, localpref 200, valid, external, best
    Community: 4259906736
    rx pathid: 0, tx pathid: 0x0
```

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

```
R1#show ip bgp 203.0.113.0
BGP routing table entry for 203.0.113.0/24, version 23
Paths: (1 available, best #1, table default)
  Advertised to update-groups:
    2          3          5
Refresh Epoch 2
65012
  198.51.100.6 from 198.51.100.6 (192.168.1.2)
    Origin IGP, metric 0, localpref 200, valid, external, best
    Community: 65001:1200
    rx pathid: 0, tx pathid: 0x0
```

# Verification – show ip bgp community

- Shows all routes that have specified community

```
R1#sh ip bgp community 65001:1200
BGP table version is 23, local router ID is 192.168.0.5
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
               t secondary path,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found
```

	Network	Next Hop	Metric	LocPrf	Weight	Path
*>	203.0.113.0	198.51.100.6	0	200	0	65012 i

# Verification – show ip bgp community-list

- Shows all routes that match specified community-list

```
R1#sh ip bgp community-list 1
BGP table version is 23, local router ID is 192.168.0.5
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
               t secondary path,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found
```

	Network	Next Hop	Metric	LocPrf	Weight	Path
*>	203.0.113.0	198.51.100.6	0	200	0	65012 i

# Extended Communities



- Standard communities are 4 octets long
- Extended communities are 8 octets long and carry a specialised 'Type' field
- A common use is in MPLS VPN, where BGP is used to propagate Service Provider's customers routes
- Extended communities are used for the Route Targets which indicate where routes should be propagated. So Customer 1 routes are shared between Customer 1's offices, and Customer 2 routes are shared between Customer 2's offices

```
neighbor ip send-community [standard (default) | extended | both]
```

- Other commands use `extcommunity`
- Eg `sh ip bgp extcommunity-list`