



High-scale BGP Multipath and Load Balancing

This chapter describes high-scale BGP multipath and load balancing techniques to improve network scalability and efficiency.

High-scale BGP multipath and load balancing refers to advanced networking features that

- enable routers to distribute traffic across multiple paths to a destination
- support configurations for various Equal Cost Multipath (ECMP) levels, including 64-way, 128-way, and 256-way ECMP
- introduce hierarchical load balancing for even greater scalability, supporting up to 1,024 ECMPs or 256 Unequal Cost Multipath (UCMP) paths, and
- are crucial for high-traffic networks where efficient load balancing is necessary to optimize bandwidth utilization and improve network performance.

You can leverage ECMP with these scaling options:

- [64-way ECMP](#) configures up to 64 equal-cost next hops and load balances traffic over up to 64 LSPs.
- [128-way ECMP](#) allows the routers to configure up to 128 ECMP next hops for BGP (iBGP/eBGP) in IPv4/IPv6 and support 64 IGP multipaths.
- [256-way ECMP](#) configures up to 256 next hops for ECMP in IPv4/IPv6 and optimizes network bandwidth by load balancing traffic across parallel paths.
- [Hierarchical load balancing](#) model expands ECMP scalability beyond hardware limitations by splitting routes.
- [64-way multipath ECMP, on page 1](#)
- [128-way multipath ECMP , on page 2](#)
- [256-way multipath ECMP, on page 5](#)
- [Hierarchical load balancing, on page 9](#)

64-way multipath ECMP

64-way multipath ECMP is a BGP feature that

- supports configuration of up to 64 equal-cost multipath (ECMP) next hops for BGP, and

- enables overloaded routers to load balance traffic over up to 64 LSPs.

ECMP routing is a routing strategy that

- enables next-hop packet forwarding to a single destination over multiple best paths which tie for top place in routing metric calculations
- can be used with most routing protocols as it is a per-hop decision limited to a single router, and
- potentially offers substantial increases in bandwidth by load-balancing traffic over multiple paths.

128-way multipath ECMP

The 128-way multipath ECMP is a BGP feature that

- enables the router to support up to 128 BGP and 64 Interior Gateway Protocol (IGP) parallel multipaths to a destination
- enables you to configure up to 128 ECMP next hops for BGP in IPv4 and IPv6 on Cisco 8000 Series Routers, and
- allows up to 128-path ECMP in the global table for BGP, specifically for iBGP and eBGP prefixes.

Table 1: Feature History Table

Feature Name	Release Name	Description
128-way ECMP	Release 7.3.1	This feature enables the router to support up to 128 BGP and 64 IGP parallel multipath routes to a destination.

Restrictions for 128-way multipath ECMP

These restrictions apply to the 128-way multipath ECMP feature:

- Virtual Routing and Forwarding (VRF) and global routing tables in IGP support 64-path ECMP as the maximum for paths, with up to 128 paths supported when backup is configured.
- BGP and IGP in UCMP support 64-path ECMP.
- Label Distribution Protocol (LDP) receives support for only 64-way ECMP from the Routing Information Base (RIB). As a result, Label Switch Database (LSD) receives a maximum of 64 equal-cost multipaths from all clients.

Configure 128-multipath ECMP

Configure 128-multipath ECMP for iBGP, eBGP, or eiBGP to enable traffic load balancing across multiple paths.

You can configure eBGP and iBGP multipath to function together. However, if eiBGP multipath is already configured, you cannot configure iBGP or eBGP multipath.

Procedure

Step 1 Configure 128-multipath ECMP for the desired BGP type, iBGP, eBGP, or eiBGP.

- Configure 128-multipath ECMP in iBGP.

```
Router(config)#router bgp 100
Router(config-bgp)#bgp router-id 10.10.10.11
Router(config-bgp)#address-family ipv4 unicast
Router(config-bgp-af)#maximum-paths ibgp 128
Router(config-bgp-af)#exit
Router(config-bgp)#address-family ipv6 unicast
Router(config-bgp-af)#maximum-paths ibgp 128
Router(config-bgp-af)#commit
```

Running configuration:

```
Router# show run router bgp
router bgp 100
bgp router-id 10.10.10.11 address-family ipv4 unicast
maximum-paths ibgp 128
!
address-family ipv6 unicast maximum-paths ibgp 128
```

- Configure 128-multipath ECMP in eBGP.

```
Router(config)#router bgp 100
Router(config-bgp)#bgp router-id 10.10.10.11
Router(config-bgp)#address-family ipv4 unicast
Router(config-bgp-af)#maximum-paths ebgp 128
Router(config-bgp-af)#exit
Router(config-bgp)#address-family ipv6 unicast
Router(config-bgp-af)#maximum-paths ebgp 128
Router(config-bgp-af)#commit
```

Running configuration:

```
Router# show run router bgp
router bgp 100
bgp router-id 10.10.10.11 address-family ipv4 unicast
maximum-paths ebgp 128
!
address-family ipv6 unicast maximum-paths ebgp 128
```

- Configure 128-multipath ECMP in eiBGP.

```
Router(config)#router bgp 100
Router(config-bgp)#bgp router-id 10.10.10.11
Router(config-bgp)#address-family ipv4 unicast
Router(config-bgp-af)#maximum-paths eibgp 128
Router(config-bgp-af)#exit
Router(config-bgp)#address-family ipv6 unicast
Router(config-bgp-af)#maximum-paths eibgp 128
Router(config-bgp-af)#commit
```

Running configuration:

```
Router# show run router bgp
router bgp 100
bgp router-id 10.10.10.11 address-family ipv4 unicast
maximum-paths eibgp 128
```

```
!
address-family ipv6 unicast maximum-paths eibgp 128
```

Step 2 Verify the BGP multipath marking in the routing table, FIB, and RIB, and confirm the total number of IPv4 and IPv6 ECMP paths, as well as the platform's ECMP capabilities.

a) Verify the BGP multipath marking.

Example:

```
Router# show bgp 192.0.2.254/24
```

```
BGP routing table entry for 191.1.0.0/24 Versions:
Process bRIB/RIB SendTblVer
Speaker 11008 11008
Last Modified: Aug 14 13:59:39.403 for 00:00:05
Paths: (35 available, best #1)
Advertised IPv4 Unicast paths to peers (in unique update groups): 100.101.3.2
Path #1: Received by speaker 0
Advertised IPv4 Unicast paths to peers (in unique update groups): 100.101.3.2
Local, (received & used)
120.0.101.1 from 120.0.101.1 (120.0.101.1)
Origin IGP, localpref 100, valid, internal, best, group-best, multipath Received Path ID 0, Local
Path ID 1, version 7708
Path #2: Received by speaker 0
Advertised IPv4 Unicast paths to peers (in unique update groups): 100.101.3.2
Local, (received & used)
120.0.102.1 from 120.0.102.1 (120.0.102.1)
Origin IGP, localpref 100, valid, internal, multipath Received Path ID 0, Local Path ID 6, version
11008
.....
Path #128: Received by speaker 0
Advertised IPv4 Unicast paths to peers (in unique update groups): 100.101.3.2
Local, (received & used)

120.0.227.1 from 120.0.227.1 (120.0.227.1)
Origin IGP, localpref 100, valid, internal, multipath Received Path ID 0, Local Path ID 6, version
14008
```

b) Verify the BGP multipath marking in FIB.

Example:

```
Router# show cef 192.0.2.254/24
191.1.0.0/24, version 46115, internal 0x5000001 0x40 (ptr 0xd236928) [1], 0x0 (0xe715668), 0x0
(0x0)
Updated Aug 14 13:59:39.007
Prefix Len 24, traffic index 0, precedence n/a, priority 4
via 120.0.101.1/32, 3 dependencies, recursive, bgp-multipath [flags 0x6080] path-idx 0 NHID 0x0
[0xd236a00 0x0]
next hop 120.0.101.1/32 via 120.0.101.1/32
via 120.0.102.1/32, 3 dependencies, recursive, bgp-multipath [flags 0x6080] path-idx 1 NHID 0x0
[0xde9a6d0 0x0]
next hop 120.0.102.1/32 via 120.0.102.1/32
via 120.0.103.1/32, 3 dependencies, recursive, bgp-multipath [flags 0x6080] path-idx 2 NHID 0x0
[0xde9a010 0x0]
...
via 120.0.227.1/32, 3 dependencies, recursive, bgp-multipath [flags 0x6080] path-idx 127 NHID 0x0
[0xde9a010 0x0]
```

c) Verify the BGP multipath marking in RIB.

Example:

```
Router# show route 192.0.2.254/24
```

```
Routing entry for 191.1.0.0/24
Known via "bgp 1", distance 200, metric 0, type internal Installed Aug 14 13:59:38.971 for 00:11:48
Routing Descriptor Blocks
120.0.101.1, from 120.0.101.1, BGP multi path Route metric is 0
120.0.102.1, from 120.0.102.1, BGP multi path Route metric is 0
...
120.0.227.1, from 120.0.227.1, BGP multi path Route metric is 0
```

- d) Verify the total number of IPv4 ECMP paths.

Example:

```
Router# show bgp ipv4 unicast 191.1.0.0/24 | i multipath | utility wc -l
128
```

- e) Verify the total number of IPv6 ECMP paths.

Example:

```
Router# show bgp ipv6 unicast 191:1::/64 | i multipath | utility wc -l
128
```

- f) Verify the IPv4 128-multipath ECMP.

Example:

```
Router# show route ipv4 191.1.0.0/24 | i multipath | utility wc -l
128
Router# show cef ipv4 191.1.0.0/24 | i multipath | utility wc -l
128
```

- g) Verify the IPv6 128-multipath ECMP.

Example:

```
Router# show route ipv6 191:1::/64 | i multi | utility wc -l
128
Router# show cef ipv6 191:1::/64 | i multipath | utility wc -l
128
```

- h) Verify the ECMP capabilities of the platform.

Example:

```
Router# show cef misc
Platform capabilities:

L3 loadbalancing levels: 2 L3 Hash buckets: 64
L3 recursive Hash buckets: 128 L3 Unequal cost hash buckets: 64
```

256-way multipath ECMP

256-way multipath ECMP is a BGP feature that

- enables configuration of up to 256 next hops for ECMP in BGP in both IPv4 and IPv6, and
- optimizes network bandwidth by load balancing traffic across parallel paths.

Equal-cost multipath routing (ECMP) is a network technique that allows data packets destined for a single location to be sent via several optimal routes. These routes are considered equally efficient based on routing criteria. This approach works with most network protocols because each router independently determines the best forwarding path. ECMP can enhance network capacity by distributing data flow across diverse routes. With the 256-path ECMP capability, a router can use up to 256 simultaneous connections to reach a given endpoint.

Cisco 8000 Series Routers allow configuration of up to 256 ECMP next hops for BGP in both IPv4 and IPv6. This maximum 256-path ECMP capability extends to the global BGP table, particularly for iBGP and eBGP prefixes.

Table 2: Feature History Table

Feature Name	Release Name	Description
256-way ECMP	Release 25.3.1	Introduced in this release on: Centralized Systems (8400 [ASIC: K100])(select variants only*) * This feature is now supported on Cisco 8404-SYS-D routers.
256-way ECMP	Release 25.1.1	Introduced in this release on: Fixed Systems (8700 [ASIC: K100]). This feature is now supported on Cisco 8712-MOD-M routers.
256-way ECMP	Release 24.4.1	Introduced in this release on: Fixed Systems (8200, 8700); Centralized Systems (8600); Modular Systems (8800 [LC ASIC: Q100, Q200, P100]) You can now configure up to 256 ECMP next hops for BGP in both IPv4 and IPv6. This optimizes network bandwidth by load balancing traffic across parallel paths.

Restrictions for 256-way multipath ECMP

These restrictions apply to the 256-way multipath ECMP feature:

- BGP and IGP in UCMP support 256 path ECMP for IP, and 64 path ECMP for MPLS.
- LDP receives only 64 ECMP support from the RIB. As a result, LSD receives a maximum of 64 ECMPs from all clients.

Configure 256-way multipath ECMP

Configure 256-way multipath ECMP for iBGP, eBGP, or eiBGP to enable traffic load balancing across multiple paths.

You can configure eBGP and iBGP multipath to function together. However, if eiBGP multipath is already configured, you cannot configure iBGP or eBGP multipath.

Procedure

Step 1 Configure 256-multipath ECMP for the desired BGP type, iBGP, eBGP, or eiBGP.

a) Configure 256-multipath ECMP in iBGP.

Example:

```
Router(config)#router bgp 100
Router(config-bgp)#bgp router-id 10.10.10.11
Router(config-bgp)#address-family ipv4 unicast
Router(config-bgp-af)#maximum-paths ibgp 256
Router(config-bgp-af)#exit
Router(config-bgp)#address-family ipv6 unicast
Router(config-bgp-af)#maximum-paths ibgp 256
Router(config-bgp-af)#commit
```

b) Configure 256-multipath ECMP in eBGP.

Example:

```
Router(config)#router bgp 100
Router(config-bgp)#bgp router-id 10.10.10.11
Router(config-bgp)#address-family ipv4 unicast
Router(config-bgp-af)#maximum-paths ebgp 256
Router(config-bgp-af)#exit
Router(config-bgp)#address-family ipv6 unicast
Router(config-bgp-af)#maximum-paths ebgp 256
Router(config-bgp-af)#commit
```

c) Configure 256-multipath ECMP in eiBGP.

Example:

```
Router(config)#router bgp 100
Router(config-bgp)#bgp router-id 10.10.10.11
Router(config-bgp)#address-family ipv4 unicast
Router(config-bgp-af)#maximum-paths eibgp 256
Router(config-bgp-af)#exit
Router(config-bgp)#address-family ipv6 unicast
Router(config-bgp-af)#maximum-paths eibgp 256
Router(config-bgp-af)#commit
```

Step 2 Verify the BGP multipath marking.

Example:

```
Router# show bgp 10.0.2.254/24

BGP routing table entry for 10.1.0.0/24
Versions:
  Process          bRIB/RIB   SendTblVer
  Speaker          11008      11008
Last Modified: Aug 14 13:59:39.403 for 00:00:05
Paths: (35 available, best #1)
  Advertised IPv4 Unicast paths to peers (in unique update groups):
```

```

10.101.3.2
Path #1: Received by speaker 0
Advertised IPv4 Unicast paths to peers (in unique update groups):
10.101.3.2
Local, (received & used)
10.0.101.1 from 10.0.101.1 (10.0.101.1)
Origin IGP, localpref 100, valid, internal, best, group-best, multipath
Received Path ID 0, Local Path ID 1, version 7708
Path #2: Received by speaker 0
Advertised IPv4 Unicast paths to peers (in unique update groups):
10.101.3.2
Local, (received & used)
10.0.102.1 from 10.0.102.1 (10.0.102.1)
Origin IGP, localpref 100, valid, internal, multipath
Received Path ID 0, Local Path ID 6, version 11008
.....
Path #256: Received by speaker 0
Advertised IPv4 Unicast paths to peers (in unique update groups):
10.101.3.2
Local, (received & used)
10.0.227.1 from 10.0.227.1 (10.0.227.1)
Origin IGP, localpref 100, valid, internal, multipath
Received Path ID 0, Local Path ID 6, version 14008

```

Step 3 Verify the BGP multipath marking in FIB.

Example:

Router# **show cef 10.0.2.254/24**

```

10.1.0.0/24, version 46115, internal 0x5000001 0x40 (ptr 0xd236928) [1], 0x0 (0xe715668), 0x0 (0x0)
Updated Aug 14 13:59:39.007
Prefix Len 24, traffic index 0, precedence n/a, priority 4
via 10.0.101.1/32, 3 dependencies, recursive, bgp-multipath [flags 0x6080]
path-idx 0 NHID 0x0 [0xd236a00 0x0]
next hop 120.0.101.1/32 via 120.0.101.1/32
via 10.0.102.1/32, 3 dependencies, recursive, bgp-multipath [flags 0x6080]
path-idx 1 NHID 0x0 [0xde9a6d0 0x0]
next hop 10.0.102.1/32 via 10.0.102.1/32
via 10.0.103.1/32, 3 dependencies, recursive, bgp-multipath [flags 0x6080]
path-idx 2 NHID 0x0 [0xde9a010 0x0]
...
via 10.0.227.1/32, 3 dependencies, recursive, bgp-multipath [flags 0x6080]
path-idx 255 NHID 0x0 [0xde9a010 0x0]

```

Step 4 Verify the BGP multipath marking in RIB.

Example:

Router# **show route 10.0.2.254/24**

```

Routing entry for 10.1.0.0/24
Known via "bgp 1", distance 200, metric 0, type internal
Installed Aug 14 13:59:38.971 for 00:11:48
Routing Descriptor Blocks
10.0.101.1, from 10.0.101.1, BGP multi path
Route metric is 0
10.0.102.1, from 10.0.102.1, BGP multi path
Route metric is 0
...
10.0.227.1, from 10.0.227.1, BGP multi path
Route metric is 0

```

Step 5 Verify the total number of IPv4 and IPv6 ECMP paths.

Example:

```
Router# show bgp ipv4 unicast 10.1.0.0/24 | i multipath | utility wc -l
256
```

```
Router# show bgp ipv6 unicast 2001:DB8::1 | i multipath | utility wc -l
256
```

Step 6 Verify the ECMP capabilities of the platform.

Example:

```
Router# show cef misc
Platform capabilities:
-----
L3 loadbalancing levels: 2
L3 Hash buckets: 64
L3 recursive Hash buckets: 256
L3 IP route Hash buckets: 256
L3 Unequal cost hash buckets: 64
L3 IP Unequal cost hash buckets: 256
```

Hierarchical load balancing

Hierarchical load balancing is a network load balancing model that

- allows you to configure up to 1,024 ECMPs or 256 UCMPs to reach a destination
- achieves this by splitting routes into multiple hierarchical ECMPs or UCMPs, and
- enables ECMP to expand beyond hardware limitations, providing greater scalability.

Table 3: Feature History Table

Feature Name	Release Name	Description
Configure 1024 Equal Cost Multi-Path or 256 Unequal Cost Multi-Path using Hierarchical Load Balancing	Release 7.3.3	In earlier releases, you could configure only up to 128 ECMPs. This feature now allows you to configure up to 1,024 ECMPs or 256 UCMPs to reach a destination. You can achieve this by splitting the routes into multiple hierarchical-based ECMPs or UCMPs. This enables ECMP to expand beyond the hardware limitation of only 512 ECMPs, providing greater scalability.

Routes are split into multiple hierarchical routes to expand the capabilities into two levels of hierarchical ECMPs or UCMPs. You can enable the router to support 1,024 ECMPs or 256 UCMPs by using hierarchical-load-balancing model. Forwarding Information Base (FIB) splits routes based on your configuration, such as Autonomous System (AS) number of each path, group size, and your specific scenario.

As a prerequisite, Cisco Express Forwarding (CEF) must be enabled on all participating routers, since load balancing requires CEF. In unequal cost multipath (UCMP) load-balancing, a weight is associated with each next hop and traffic is distributed across the next hops in proportion to their weight. In ECMP, the route to a destination has multiple next hops and traffic is equally distributed.

Support was previously limited to 128 ECMPs. Starting from Release 7.3.3, the support is extended to 1,024 ECMPs or 256 UCMPs. At any given time, you can configure either 1,024 ECMPs or 256 UCMPs.

Restrictions and guidelines for hierarchical load balancing

- The hierarchical-load-balancing model works only for IP paths, which enhances eBGP interface peering scenario.
- This model does not apply to any path that includes an MPLS label or SRv6 information, because such information prevents CEF from converting flat load-balancing to a hierarchical-load-balancing model.
- CEF must be enabled on all participating routers.

How hierarchical load balancing works

Summary

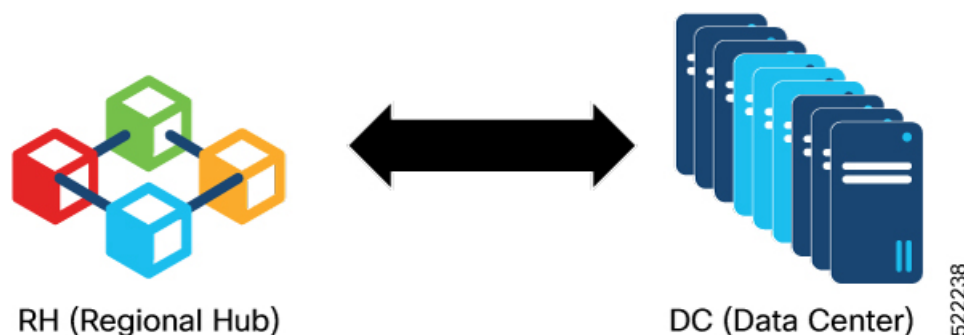
The key components involved in the process are:

- BGP: Downloads neighbor AS information for each path.
- Routing Information Base (RIB): Receives path information from BGP and passes it to FIB.
- FIB: Splits routes and groups paths into hierarchical levels based on configuration and received information.

Hierarchical load balancing is a network load balancing model that extends the capabilities of ECMP and UCMP by splitting and grouping routes into multiple hierarchical levels. This process involves the cooperation of BGP, RIB, and FIB to manage path information and distribute traffic efficiently.

Workflow

Figure 1: Sample topology of hierarchical load balancing



These stages describe hierarchical load balancing.

1. BGP downloads neighbor AS information for each path.

2. BGP forwards this path information to the RIB.
3. The RIB then passes this information to the FIB for regrouping.
4. The FIB splits routes based on the configured group size, AS number of each path, and the specific scenario.
5. The router-specific FIB groups these paths into two hierarchical levels.

Result

The sample topology shows the regional hub connected with 32 data centers with eBGP interface peering. Each data center has 32 interface connections, which adds up to a total of 1,024 ECMPs.

By enabling a hierarchical load-balancing UCMP group size of 256 within a data center, you can scale 256 UCMP eBGP paths on top-of-rack switching using weights.

Configure hierarchical load balancing

Enable and configure hierarchical load balancing to support a higher number of ECMP or UCMP paths for efficient traffic distribution.

Hierarchical load balancing allows you to scale beyond traditional ECMP limits by splitting and grouping routes. To use this feature, enable it and define route policies for UCMP.

Before you begin

Ensure CEF is enabled on all participating routers.

Procedure

Step 1 Run the **cef hierarchical-load-balancing** command to enable hierarchical load balancing with 1024 paths in ECMP mode.

For ECMP mode, the FIB splits or groups paths according to the path attribute of the AS number. Paths to the same destination must be configured on the same AS. Because of hardware limitations, the maximum number of remote AS numbers should be fewer than 128.

- If paths are less than the min-path, FIB uses native forwarding mode.
- If paths are greater than or equal to the min-path, FIB converts the forwarding chain to hierarchical forwarding.

Example:

```
Router(config)#router bgp 100
Router(config-bgp)#bgp router-id 10.10.10.11
Router(config-bgp)#address-family ipv4 unicast
Router(config-bgp-af)#maximum-paths ebgp 1024
Router(config-bgp-af)#maximum-paths unique-nexthop-check-disable
Router(config-bgp-af)#exit
Router(config-bgp)#address-family ipv6 unicast
Router(config-bgp-af)#maximum-paths ebgp 1024
Router(config-bgp-af)#maximum-paths unique-nexthop-check-disable
Router(config-bgp-af)#exit
Router(config-bgp)#exit
Router(config)#cef hierarchical-load-balancing ecmp min-paths 128
```

Note

Reload the router so that the **cef hierarchical-load-balancing ecmp min-paths** command takes effect.

Step 2 Configure group size and route policy for UCMP links.

Define an extended community bandwidth routing policy that uses weight values to control traffic on UCMP links.

For example, the system applies the ratio *AS number to Weight number* on the egress of BGP. The FIB uses group size to split or group paths into hierarchical forwarding mode.

- The group size value must be within the hardware limitation of 64 in case of IGP level load balancing.
- The recommended group size is 64 if all paths are ECMP.
- If paths have weight attributes, the recommended group size is only 32.

Example:

```
Router(config)# cef hierarchical-load-balancing ucmp group-size 32

Router(config)#route-policy BW1
Router(config-rpl)#set extcommunity bandwidth (800:10000)
Router(config-rpl)#end-policy
Router(config)#route-policy BW2
Router(config-rpl)#set extcommunity bandwidth (800:20000)
Router(config-rpl)#end-policy
Router(config)#route-policy BW3
Router(config-rpl)#set extcommunity bandwidth (800:40000)
Router(config-rpl)#end-policy
Router(config)#route-policy BW4
Router(config-rpl)#set extcommunity bandwidth (800:80000)
Router(config-rpl)#end-policy
Router(config)#commit
```

Note

Reload the router so that the **cef hierarchical-load-balancing ucmp group-size** command takes effect.

Step 3 Verify CEF configuration for hierarchical load balancing.

- Verify the BGP IPv4 unicast prefix status.

Example:

```
Router#show bgp ipv4 unicast 1.4.0.0/16
BGP routing table entry for 1.4.0.0/16
Versions:
Process bRIB/RIB SendTblVer
Speaker 3 3
Last Modified: Jan 31 13:14:07.023 for 08:53:38
Paths: (1 available, best #1)
Advertised IPv4 Unicast paths to update-groups (with more than one peer): 0.5
Path #1: Received by speaker 0
Advertised IPv4 Unicast paths to update-groups (with more than one peer): 0.5
Local
100.100.100.200 (metric 30) from 100.100.100.101 (100.100.100.200)
Origin incomplete, metric 0, localpref 100, valid, internal, best, group-best Received Path ID 1,
Local Path ID 1, version 3
Originator: 100.100.100.200, Cluster list: 100.100.100.101
```

- Verify the CEF configuration status for ECMP.

Example:

```

Router# show cef 200.1.0.0 detail
200.1.0.0/24, version 92168105, internal 0x5000001 0x40 (ptr 0xaa001058) [1], 0x0 (0x0), 0x0 (0x0)
Updated Sep 8 23:04:13.489
Prefix Len 24, traffic index 0, precedence n/a, priority 4
gateway array (0x9232e858) reference count 596, flags 0x2010, source rib (7), 0 backups
[1 type 3 flags 0x48441 (0xa9df9098) ext 0x0 (0x0)] LW-LDI[type=0, refc=0, ptr=0x0, sh-ldi=0x0]
gateway array update type-time 1 Sep 8 23:04:13.490
LDI Update time Sep 8 23:04:13.490 Weight distribution:
slot 0, weight 5120, normalized_weight 15
slot 1, weight 5120, normalized_weight 15
slot 2, weight 5120, normalized_weight 15
slot 3, weight 5120, normalized_weight 15

Level 1 - Load distribution: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3

```

c) Verify the CEF configuration status for UCMF.

Example:

```

Router# show cef 200.1.0.0 detail
200.1.0.0/24, version 6044, internal 0x5000001 0x40 (ptr 0xa64e7738) [1], 0x0 (0x0), 0x0 (0x0)
Prefix Len 24, traffic index 0, precedence n/a, priority 4
gateway array (0x92f14758) reference count 5001, flags 0x2010, source rib (7), 0 backups [1 type
 3 flags 0x40441 (0xa4d331d8) ext 0x0 (0x0)]
LW-LDI[type=0, refc=0, ptr=0x0, sh-ldi=0x0]
gateway array update type-time 3 Nov 26 16:32:49.195
LDI Update time Nov 26 16:32:49.195 Weight distribution:
slot 0, weight 2560, normalized_weight 2
slot 1, weight 2560, normalized_weight 2
slot 2, weight 2560, normalized_weight 2
slot 3, weight 2560, normalized_weight 2
slot 4, weight 1280, normalized_weight 1
slot 5, weight 1280, normalized_weight 1
slot 6, weight 1280, normalized_weight 1
slot 7, weight 1280, normalized_weight 1
Level 1 - Load distribution: 0 0 1 1 2 2 3 3 4 5 6 7
[0] via 241.0.58.138/32, recursive
[1] via 241.0.58.138/32, recursive
[2] via 241.0.58.134/32, recursive
[3] via 241.0.58.134/32, recursive
[4] via 241.0.58.130/32, recursive
[5] via 241.0.58.130/32, recursive
[6] via 241.0.58.126/32, recursive
[7] via 241.0.58.126/32, recursive
[8] via 241.0.58.122/32, recursive
[9] via 241.0.58.118/32, recursive
[10] via 241.0.58.114/32, recursive
[11] via 241.0.58.110/32, recursive
via 241.0.58.138/32, 4 dependencies, recursive [flags 0x0] path-idx 0 NHID 0x0 [0x930b65e0 0x0]
next hop VRF - 'iid', table - 0xe0000203 next hop 241.0.58.138/32 via 241.0.58.138/32
Load distribution: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
27 28 29 30 31 (refcount 1)
Hash OK Interface Address
0 Y TenGigE0/0/0/21/3.1 18.1.0.1
1 Y TenGigE0/0/0/21/3.2 18.1.1.1
2 Y TenGigE0/0/0/21/3.3 18.1.2.1
3 Y TenGigE0/0/0/21/3.4 18.1.3.1
4 Y TenGigE0/0/0/21/3.5 18.1.4.1
5 Y TenGigE0/0/0/21/3.6 18.1.5.1
6 Y TenGigE0/0/0/21/3.7 18.1.6.1
7 Y TenGigE0/0/0/21/3.8 18.1.7.1
8 Y TenGigE0/0/0/21/3.9 18.1.8.1
9 Y TenGigE0/0/0/21/3.10 18.1.9.1
10 Y TenGigE0/0/0/21/3.11 18.1.10.1

```

```

11 Y TenGigE0/0/0/21/3.12 18.1.11.1
12 Y TenGigE0/0/0/21/3.13 18.1.12.1
13 Y TenGigE0/0/0/21/3.14 18.1.13.1
14 Y TenGigE0/0/0/21/3.15 18.1.14.1
15 Y TenGigE0/0/0/21/3.16 18.1.15.1
16 Y TenGigE0/0/0/21/3.17 18.1.16.1
17 Y TenGigE0/0/0/21/3.18 18.1.17.1
18 Y TenGigE0/0/0/21/3.19 18.1.18.1
19 Y TenGigE0/0/0/21/3.20 18.1.19.1
20 Y TenGigE0/0/0/21/3.21 18.1.20.1
21 Y TenGigE0/0/0/21/3.22 18.1.21.1
22 Y TenGigE0/0/0/21/3.23 18.1.22.1
23 Y TenGigE0/0/0/21/3.24 18.1.23.1
24 Y TenGigE0/0/0/21/3.25 18.1.24.1
25 Y TenGigE0/0/0/21/3.26 18.1.25.1
26 Y TenGigE0/0/0/21/3.27 18.1.26.1
27 Y TenGigE0/0/0/21/3.28 18.1.27.1
28 Y TenGigE0/0/0/21/3.29 18.1.28.1
29 Y TenGigE0/0/0/21/3.30 18.1.29.1
30 Y TenGigE0/0/0/21/3.31 18.1.30.1
31 Y TenGigE0/0/0/21/3.32 18.1.31.1

```

Hierarchical load balancing is enabled and configured, allowing for expanded ECMP or UCMP capabilities.