# Advanced CCIE SERVICE PROVIDER v3.0

www.MicronicsTraining.com

**Narbik Kocharians**
**CCIE #12410**
**R&S, Security, SP**

**Paul Negron**
**CCIE #14856**
**SP**

# Table of Content:

| EIGRP | | |
|---|---|---|
| Lab 1 EIGRP Configuration | 5 | Vol-I |
| Lab 2 Advanced EIGRP Stub Configuration | 76 | Vol-I |
| Lab 3 EIGRP & Default-information | 85 | Vol-I |
| Lab 4 EIGRP Filtering | 96 | Vol-I |
| ISIS | | |
| Lab 1 Basic ISIS configuration | 105 | Vol-I |
| Lab 2 Multi-Instance | 117 | Vol-I |
| Lab 3 Multi-Topology | 125 | Vol-I |
| Lab 4 Redistribution & Summarization | 137 | Vol-I |
| Lab 5 ISIS Efficiency Mechanism | 147 | Vol-I |
| Lab 6 Authentication | 162 | Vol-I |
| Lab 7 Advanced Configuration I | 167 | Vol-I |
| Lab 8 Advanced Configuration II | 174 | Vol-I |
| OSPF | | |
| Lab 1 Advertising Networks | 187 | Vol-I |
| Lab 2 Optimization of OSPF & Adjusting Timers | 190 | Vol-I |
| Lab 3 OSPF Authentication | 197 | Vol-I |
| Lab 4 OSPF Cost | 222 | Vol-I |
| Lab 5 OSPF Summarization | 227 | Vol-I |
| Lab 6 Virtual-links and GRE Tunnels | 234 | Vol-I |
| Lab 7 OSPF Stub, T/Stub, and NSSAs | 244 | Vol-I |
| Lab 8 OSPF Filtering | 254 | Vol-I |
| Lab 9 Additional OSPF Filtering | 280 | Vol-I |
| Lab 10 Redirecting Traffic in OSPF | 290 | Vol-I |
| Lab 11 Database Overload Protection | 295 | Vol-I |
| Lab 12 OSPF Non-Broadcast Networks | 300 | Vol-I |
| Lab 13 OSPF Broadcast Networks | 309 | Vol-I |
| Lab 14 OSPF Point-to-Point Networks | 313 | Vol-I |
| Lab 15 OSPF Point-to-Multipoint Networks | 317 | Vol-I |
| Lab 16 OSPF Point-to-Multi Network – II | 324 | Vol-I |
| Lab 17 OSPF P-to-M Non-Broadcast Networks | 331 | Vol-I |
| Lab 18 OSPF and NBMA | 337 | Vol-I |
| Lab 19 Forward Address Suppression | 345 | Vol-I |
| Lab 20 OSPF NSSA No-redistribution & Injection of Default Routes | 351 | Vol-I |
| BGP | | |
| Lab 1 Establishing Neighbor Adjacency | 345 | Vol-I |
| Lab 2 Route Reflectors | 362 | Vol-I |
| Lab 3 Conditional Adv. & Back door | 378 | Vol-I |

| | | |
|---|---|---|
| Lab 4 Route Dampening | 393 | Vol-I |
| Lab 5 Route Aggregation | 402 | Vol-I |
| Lab 6 The Community Attribute | 422 | Vol-I |
| Lab 7 BGP Cost Community | 438 | Vol-I |
| Lab 8 BGP & Load Balancing – I | 447 | Vol-I |
| Lab 9 BGP Load Balancing – II | 470 | Vol-I |
| Lab 10 BGP Unequal Cost Load Balancing | 474 | Vol-I |
| Lab 11 BGP Local Preference – I | 482 | Vol-I |
| Lab 12 BGP Local Preference – II | 493 | Vol-I |
| Lab 13 The AS-Path Attribute | 501 | Vol-I |
| Lab 14 The Weight Attribute | 509 | Vol-I |
| Lab 15 MED | 516 | Vol-I |
| Lab 16 Filtering Using ACLs & Prefix-lists | 532 | Vol-I |
| Lab 17 Regular Expressions | 542 | Vol-I |
| Lab 18 Advanced BGP Configurations | 559 | Vol-I |
| Lab 19 Administrative Distance | 569 | Vol-I |
| Lab 20 BGP Confederation | 577 | Vol-I |
| Lab 21 BGP Hiding Local AS Number | 582 | Vol-I |
| Lab 22 BGP Allowas-in | 590 | Vol-I |
| **Frame-relay** | | |
| Lab 1 Hub-n-Spoke Using Frame Map Statements | 5 | Vol-II |
| Lab 2 Hub-n-Spoke Frame-Relay Point-to-Point | 19 | Vol-II |
| Lab 3 Mixture of P2P and Multipoint | 24 | Vol-II |
| Lab 4 Multipoint Frame-Relay W/O Frame maps | 29 | Vol-II |
| Lab 5 Frame-Relay and Authentication | 35 | Vol-II |
| Lab 6 Frame-Relay End-to-End Keepalives | 44 | Vol-II |
| Lab 7 Tricky Frame-Relay Configuration | 59 | Vol-II |
| Lab 8 Frame-Relay Multilinking | 67 | Vol-II |
| Lab 9 Back-to-Back Frame-Relay connection | 74 | Vol-II |
| **Multicasting** | | |
| Lab 1 Configuring IGMP | 91 | Vol-II |
| Lab 2 Dense Mode | 109 | Vol-II |
| Lab 3 Static RP Configuration | 127 | Vol-II |
| Lab 4 Auto-RP | 141 | Vol-II |
| Lab 5 Auto-RP Filtering & Listener | 163 | Vol-II |
| Lab 6 Configuring BSR | 184 | Vol-II |
| Lab 7 Configuring MSDP | 198 | Vol-II |
| Lab 8 Anycast RP | 215 | Vol-II |
| Lab 9 MSDP/MP-BGP | 225 | Vol-II |
| Lab 10 Configuring SSM | 244 | Vol-II |
| Lab 11 Helper-Map | 255 | Vol-II |
| Lab 12 Bidirectional PIM | 262 | Vol-II |
| **Security** | | |
| Lab 1 Basic Router Security Configuration | 280 | Vol-II |

| | | |
|---|---|---|
| Lab 2 Standard Named Access List | 287 | Vol-II |
| Lab 3 Controlling Telnet Access and SSH | 291 | Vol-II |
| Lab 4 Extended Access List IP and ICMP | 298 | Vol-II |
| Lab 5 Extended Access List OSPF & EIGRP | 304 | Vol-II |
| Lab 6 Using MQC as a Filtering tool | 308 | Vol-II |
| Lab 7 Extended Access List With Established | 312 | Vol-II |
| Lab 8 Dynamic Access List | 315 | Vol-II |
| Lab 9 Reflexive Access-Lists | 325 | Vol-II |
| Lab 10 Access-list & Time Range | 331 | Vol-II |
| Lab 11 Black Hole Filtering | 335 | Vol-II |
| Lab 12 Configuring uRPF | 344 | Vol-II |
| Lab 13 Control Plane Policing | 350 | Vol-II |
| Lab 14 Attacks | 357 | Vol-II |
| **Syslog & IP Accounting** | | |
| Lab 1 Syslog | 369 | Vol-II |
| Lab 2 IP Accounting | 373 | Vol-II |
| Lab 3 IP SLA | 388 | Vol-II |
| Lab 4 Reliable Static Routing using IP SLA | 394 | Vol-II |
| Lab 5 Reliable Conditional Default Route Injection using IP SLA | 401 | Vol-II |
| **QoS** | | |
| Lab 1 Priority Queuing | 415 | Vol-II |
| Lab 2 Custom Queuing | 421 | Vol-II |
| Lab 3 WFQ | 425 | Vol-II |
| Lab 4 CBWFQ | 429 | Vol-II |
| Lab 5 CBWFQ – II | 435 | Vol-II |
| Lab 6 Converting Custom Queuing to CBWFQ | 437 | Vol-II |
| Lab 7 LLQ | 440 | Vol-II |
| Lab 8 Class Based Policing – I | 444 | Vol-II |
| Lab 9 CB Policing – II | 453 | Vol-II |
| Lab 10 WRED & CB WRED | 458 | Vol-II |
| Lab 11 RSVP | 463 | Vol-II |
| Lab 12 Match Access-Group | 469 | Vol-II |
| Lab 13 Match Destination & Source Add MAC | 474 | Vol-II |
| Lab 14 Match Input-Interface | 480 | Vol-II |
| Lab 15 Match FR-de & Packet Length | 483 | Vol-II |
| Lab 16 Match IP Precedence vs. Match Precedence | 491 | Vol-II |
| **Switching (ME 3400)** | | |
| Lab 1 Switching Port-types | 503 | Vol-II |
| Lab 2 Basic L2 Tunneling Mechanism | 513 | Vol-II |
| Lab 3 Provider Bridging (PB) | 520 | Vol-II |

| Subject | Page | Volume |
|---|---|---|
| **LDP** | | |
| Lab 1 Configuring Label Distribution Protocol | 5 | Vol-III |
| **VRF-LITE** | | |
| Lab 1 PBR Based on Source IP Address | 75 | Vol-III |
| **MPLS & L3VPNs** | | |
| Lab 1 Static & RIPv2 Routing in a VPN | 82 | Vol-III |
| Lab 2 OSPF Routing in a VPN | 111 | Vol-III |
| Lab 3 Backdoor links & OSPF | 131 | Vol-III |
| Lab 4 EIGRP Routing in a VPN | 147 | Vol-III |
| Lab 5 EIGRP SOO | 162 | Vol-III |
| Lab 5 BGP Routing in a VPN | 180 | Vol-III |
| Lab 6 Complex VPNs and Filters | 197 | Vol-III |
| **MPLS and NAT** | | |
| Lab 1 MPLS and NAT – I | 216 | Vol-III |
| Lab 2 MPLS and NAT – II | 229 | Vol-III |
| **Carrier Supporting Carrier** | | |
| Lab 1 Carrier Supporting Carrier | 239 | Vol-III |
| Lab 2 Inter-AS VPN | 252 | Vol-III |
| **VPN Extranet & Internet Access** | | |
| Lab 1 Internet Access Methods Using Separate VPN & Dual PVC | 265 | Vol-III |
| Lab 2 Static Internet Access Using Global Next-hop | 280 | Vol-III |
| **Multicast VPN (mVPN)** | | |
| Lab 1 Multicast VPN (mVPN) | 287 | Vol-III |
| **Layer 3 VPNs 6PE/6VPE** | | |
| Lab 1 6PE | 304 | Vol-III |
| Lab 2 6VPE | 316 | Vol-III |
| **Layer 2 VPNs** | | |
| Lab 1 Confguring MPLS & layer 2 VPNs (AToM) | 328 | Vol-III |
| Lab 2 L2TPv3 VPN | 339 | Vol-III |
| Lab 3 L2TPv3 Internetworking | 344 | Vol-III |
| **MPLS Traffic Engineering** | | |
| Lab 1 Confguring Static Tunnels | 352 | Vol-III |
| Lab 2 Dynamic Tunnels | 369 | Vol-III |
| Lab 3 Load Balancing | 383 | Vol-III |
| Lab 4 Fast Reroute backup Tunnels | 394 | Vol-III |
| Lab 5 MPLS QoS | 405 | Vol-III |
| **IOS-XR** | | |
| Lab 1 IOS-XR Quick Reference | 420 | Vol-III |
| **IOS-XR and Layer 3 VPNs** | | |

| | | |
|---|---|---|
| Lab 1 Static and RIPv2 Routing in a VPN | 430 | Vol-III |
| Lab 2 OSPF Routing in a VPN | 463 | Vol-III |
| Lab 3 Backdoor links and OSPF | 482 | Vol-III |
| Lab 4 Eigrp Routing in a VPN | 500 | Vol-III |
| Lab 5 BGP Routing ina VPN | 517 | Vol-III |
| IOS-XR Traffic Engineering | | |
| Lab 1 Configuring Static Tunnels | 529 | Vol-III |
| Lab 2 Dynamic Tunnels | 547 | Vol-III |
| VPWS and VPLS | | |
| Lab 1 VPWS | 559 | Vol-III |
| Lab 2 VPLS (EP-LAN) | 568 | Vol-III |
| Carrier Supporting Carrier (CSC) on IOS-XR | | |
| Lab 1 CSC | 576 | Vol-III |

# Advanced
# CCIE SERVICE PROVIDER
# v3.0

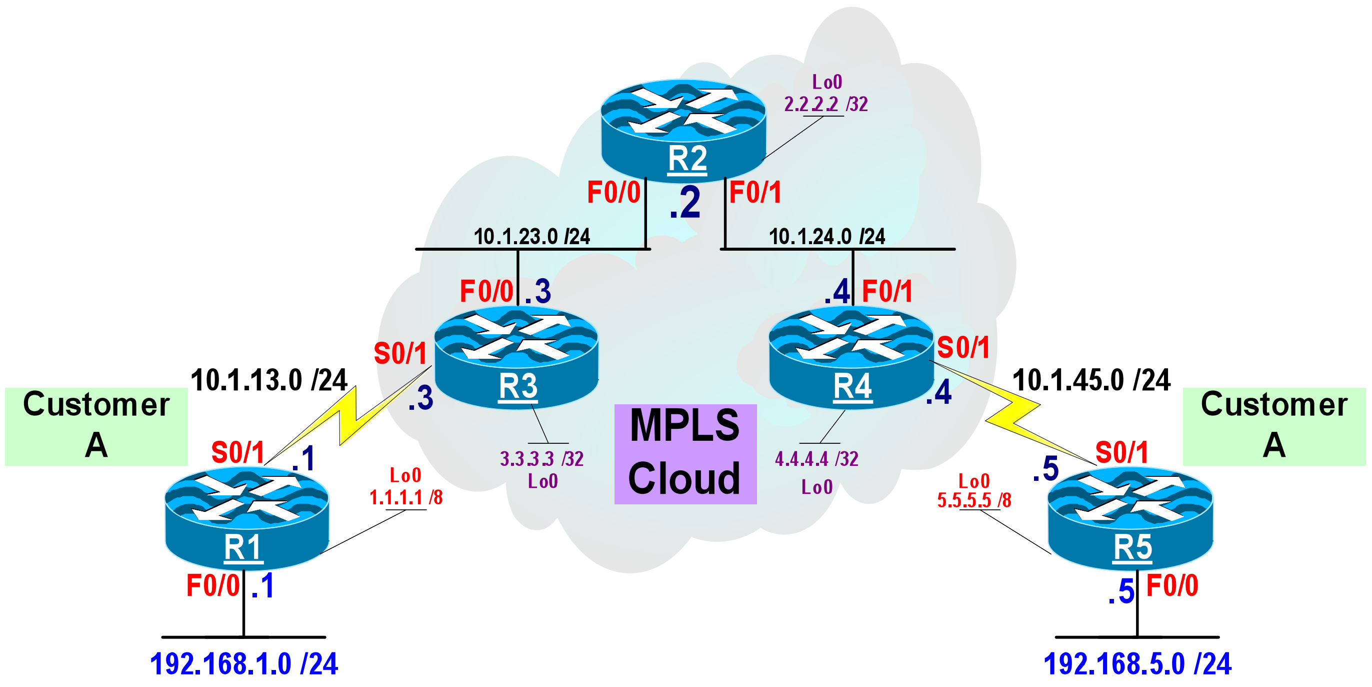
**www.MicronicsTraining.com**

**Narbik Kocharians**
**CCIE #12410**
**R&S, Security, SP**

**Paul Negron**
**CCIE #14856**
**SP**

# MPLS L3/VPNs

# Lab 1
# Static & RIPv2 routing in a VPN



## Lab Setup:

- The connections between R1 and R3, R4 and R5 should be configured with HDLC encapsulation. The clock rate should be set to 64000.
- Configure F0/0 of R1 in VLAN 100 and R5 in VLAN 500.
- Configure the F0/0 interface of R2 and R3 in VLAN 200.
- Configure the F0/1 interface of R2 and R4 in VLAN 300.
- Configure the rest of the routers according to the above diagram.

## Task 1

Configure OSPF on Core MPLS routers (R2, R3 and R4), you should run OSPF area 0 on the F0/0 interface of R2 and R3, F0/1 interface of R2 and R4 and the loopback interfaces of these routers.

### On R2

```
R2(config)#router ospf 1
R2(config-router)#netw 10.1.23.2 0.0.0.0 area 0
R2(config-router)#netw 10.1.24.2 0.0.0.0 area 0
R2(config-router)#netw 2.2.2.2 0.0.0.0 area 0
```

### On R3

```
R3(config)#router ospf 1
R3(config-router)#netw 3.3.3.3 0.0.0.0 area 0
R3(config-router)#netw 10.1.23.3 0.0.0.0 are 0
```

### On R4

```
R4(config)#router ospf 1
R4(config-router)#netw 4.4.4.4 0.0.0.0 area 0
R4(config-router)#netw 10.1.24.4 0.0.0.0 area 0
```

### To verify the configuration:

### On R3

```
R3#Show ip route ospf | Inc O

O       2.2.2.2 [110/2] via 10.1.23.2, 00:00:24, FastEthernet0/0
O       4.4.4.4 [110/3] via 10.1.23.2, 00:00:24, FastEthernet0/0
O       10.1.24.0 [110/2] via 10.1.23.2, 00:00:24, FastEthernet0/0
```

### On R2

```
R2#Show ip route ospf | Inc O

O       3.3.3.3 [110/2] via 10.1.23.3, 00:01:03, FastEthernet0/0
O       4.4.4.4 [110/2] via 10.1.24.4, 00:01:03, FastEthernet0/1
```

### On R4

```
R4#Show ip route ospf | Inc O

O        2.2.2.2 [110/2] via 10.1.24.2, 00:01:41, FastEthernet0/1
O        3.3.3.3 [110/3] via 10.1.24.2, 00:01:41, FastEthernet0/1
O        10.1.23.0 [110/2] via 10.1.24.2, 00:01:41, FastEthernet0/1
```

## Task 2

Configure LDP between the core routers. These routers should use their loopback 0 interfaces as their LDP router ID; the core MPLS routers (R2, R3 and R4) should use the following label range:

R2 – 200 – 299
R3 – 300 – 399
R4 – 400 – 499

**On R3**

```
R3(config)#MPLS label protocol LDP

R3(config)#MPLS ldp router-id lo0

R3(config)#MPLS label range 300 399

R3(config)#int f0/0
R3(config-if)#MPLS IP
```

**On R2**

```
R2(config)#MPLS label protocol LDP

R2(config)#MPLS LDP router-id lo0

R2(config)#MPLS label range 200 299

R2(config)#int F0/0
R2(config-if)#MPLS IP

R2(config-if)#int F0/1
R2(config-if)#MPLS IP
```

**On R4**

```
R4(config)#MPLS label protocol LDP

R4(config)#MPLS LDP router-id lo0

R4(config)#MPLS label range 400 499

R4(config)#int F0/1
R4(config-if)#MPLS IP
```

**To verify the configuration:**

**On R4**

```
R4#Show mpls interface

Interface              IP            Tunnel   Operational
FastEthernet0/1        Yes (ldp)     No       Yes

R4#Show mpls ldp neighbor

    Peer LDP Ident: 2.2.2.2:0; Local LDP Ident 4.4.4.4:0
        TCP connection: 2.2.2.2.646 - 4.4.4.4.50597
        State: Oper; Msgs sent/rcvd: 10/9; Downstream
        Up time: 00:01:34
        LDP discovery sources:
          FastEthernet0/1, Src IP addr: 10.1.24.2
        Addresses bound to peer LDP Ident:
          10.1.23.2       10.1.24.2       2.2.2.2

R4#Show mpls ldp discovery all

Local LDP Identifier:
    4.4.4.4:0
    Discovery Sources:
    Interfaces:
        FastEthernet0/1 (ldp): xmit/recv
            LDP Id: 2.2.2.2:0

R4#Show mpls label range

Downstream Generic label region: Min/Max label: 400/499
```

**The default range for the labels is 16 to 100,000 on this platform. These number ranges are software and platform specific. Over 353,000 labels are supported on 6500's as of the writing of this work book.**

## On R2

```
R2#Show mpls interfaces

Interface              IP            Tunnel   Operational
FastEthernet0/0        Yes (ldp)     No       Yes
FastEthernet0/1        Yes (ldp)     No       Yes

R2#Show mpls ldp neighbor

    Peer LDP Ident: 3.3.3.3:0; Local LDP Ident 2.2.2.2:0
        TCP connection: 3.3.3.3.23184 - 2.2.2.2.646
        State: Oper; Msgs sent/rcvd: 12/13; Downstream
        Up time: 00:04:01
        LDP discovery sources:
          FastEthernet0/0, Src IP addr: 10.1.23.3
        Addresses bound to peer LDP Ident:
          10.1.23.3       10.1.13.3       3.3.3.3
    Peer LDP Ident: 4.4.4.4:0; Local LDP Ident 2.2.2.2:0
        TCP connection: 4.4.4.4.50597 - 2.2.2.2.646
        State: Oper; Msgs sent/rcvd: 11/13; Downstream
        Up time: 00:03:22
        LDP discovery sources:
          FastEthernet0/1, Src IP addr: 10.1.24.4
        Addresses bound to peer LDP Ident:
          10.1.24.4       10.1.45.4       4.4.4.4

R2#Show mpls ldp discovery all

Local LDP Identifier:
    2.2.2.2:0
    Discovery Sources:
    Interfaces:
        FastEthernet0/0 (ldp): xmit/recv
            LDP Id: 3.3.3.3:0
        FastEthernet0/1 (ldp): xmit/recv
            LDP Id: 4.4.4.4:0

R2#Show mpls label range

Downstream Generic label region: Min/Max label: 200/299
```

## On R3

```
R3#Show mpls interfaces
```

```
Interface              IP            Tunnel   Operational
FastEthernet0/0        Yes (ldp)     No       Yes

R3#Show mpls ldp neighbor

    Peer LDP Ident: 2.2.2.2:0; Local LDP Ident 3.3.3.3:0
        TCP connection: 2.2.2.2.646 - 3.3.3.3.23184
        State: Oper; Msgs sent/rcvd: 13/12; Downstream
        Up time: 00:04:01
        LDP discovery sources:
          FastEthernet0/0, Src IP addr: 10.1.23.2
        Addresses bound to peer LDP Ident:
          10.1.23.2       10.1.24.2       2.2.2.2

R3#Show mpls ldp discovery all

Local LDP Identifier:
    3.3.3.3:0
    Discovery Sources:
    Interfaces:
        FastEthernet0/0 (ldp): xmit/recv
            LDP Id: 2.2.2.2:0

R3#Show mpls label range

Downstream Generic label region: Min/Max label: 300/399
```

## To verify the LFIB of these routers:

**NOTE:**

**The labels displayed in the output of this lab may differ from your lab but the principle remains the same.**

## On R3

```
R3#Show mpls forwarding-table

Local  Outgoing    Prefix            Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
300    Pop tag     2.2.2.2/32        0          Fa0/0      10.1.23.2
301    201         4.4.4.4/32        0          Fa0/0      10.1.23.2
302    Pop tag     10.1.24.0/24      0          Fa0/0      10.1.23.2
```

**On R2**

```
R2#Show mpls forwarding-table

Local  Outgoing    Prefix            Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
200    Pop tag     3.3.3.3/32        0          Fa0/0      10.1.23.3
201    Pop tag     4.4.4.4/32        0          Fa0/1      10.1.24.4
```

**On R4**

```
R4#Show mpls forwarding-table

Local  Outgoing    Prefix            Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
400    Pop tag     2.2.2.2/32        0          Fa0/1      10.1.24.2
401    200         3.3.3.3/32        0          Fa0/1      10.1.24.2
402    Pop tag     10.1.23.0/24      0          Fa0/1      10.1.24.2
```

## Task 3

Configure MP-BGP between R3 and R4 as they represent the Provider Edge routers in this topology in AS 65001. Do not allow the BGP peers to share IPV4 routing information by default. The only bgp peering relationship should be VPNv4.

**On R3**

```
R3(config)#Router bgp 65001
R3(config-router)#NO BGP default ipv4-unicast
R3(config-router)#Neighbor 4.4.4.4 remote-as 65001
R3(config-router)#Neighbor 4.4.4.4 Update-source Lo0
```

**Note the exchange of IPv4 routes between BGP neighbors are enabled by default, which means the configured neighbors will NOT only establish a BGP session but they will also receive the advertised prefixes.**

**Since the "NO bgp default ipv4-unicast" is configured first (Before the neighbor commands), the local router will NOT establish a BGP peer session with any neighbor unless that neighbor is activated.**

**On R4**

```
R4(config)#Router bgp 65001
R4(config-router)#NO BGP default ipv4-unicast
R4(config-router)#Neighbor 3.3.3.3 remote-as 65001
R4(config-router)#Neighbor 3.3.3.3 Update-source Lo0
```

**To verify the configuration:**

**On R3**

```
R3#Show ip bgp
R3#

R3#Show ip bgp summary
R3#
```

**Note there is no IPv4 neighbor adjacency established between the two routers. Since the task states that these two BGP speakers should ONLY establish a VPNv4 peer session, the neighbors MUST be activated in the address-family VPNv4.**

**On R3**

```
R3(config)#Router bgp 65001
R3(config-router)#Address-family vpnv4 unicast
R3(config-router-af)#Neighbor 4.4.4.4 Activate
```

**On R4**

```
R4(config)#Router bgp 65001
R4(config-router)#Address-family vpnv4 unicast
R4(config-router-af)#Neighbor 3.3.3.3 Activate
```

**You should see the following console message:**

```
%BGP-5-ADJCHANGE: neighbor 3.3.3.3 Up
```

**To verify the configuration:**

**On R3**

```
R3#Show ip bgp VPNv4 all summary | B Neighbor

Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
4.4.4.4         4 65001      12      12        1    0    0 00:08:44        0
```

**On R4**

```
R4#Show ip bgp VPNv4 all summary | B Neighbor

Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
3.3.3.3         4 65001      13      13        1    0    0 00:09:57        0
```

**NOTE: The BGP speakers have ONLY established a VPNv4 peering.**

**On R3**

```
R4#Sh ip bgp summ | B Neigh
R4#
```

**On R4**

```
R4#Sh ip bgp summ | B Neigh
R4#
```

**On R2**

```
R2#Show ip bgp summary

% BGP not active
```

**Note R2 is NOT running BGP at all, and ONLY the edge routers are running BGP. Running BGP on the core routers (P) is NOT necessary; these routers ONLY perform label switching.**

## Task 4

Configure a **Virtual Routing Forwarding (VRF) Instance with a name of CA** (For Customer A), a **route-distinguisher (RD) of "1:10"** and a **route-target (RT) of "1:100"** on R3. On R4, the same route targets will be used for the vrf, but the **RD should be configured to be "1:20"** and the name of the VRF should be configured as **CB**.

**The "IP vrf" command creates a new VRF and enters the global configuration mode for that specific VRF. The name of the VRF is locally significant and it's case sensitive. VRF is NOT operational unless the RD is defined.**

**The "RD" VRF configuration mode command is used to define and assign an RD to a VRF, remember that the VRF is NOT operational without an RD. RD is a 64 bit value used to transform 32 bit**

**customer IPv4 address, which is NON-Unique into a Unique 96 bit addresses called VPNv4. These addresses are ONLY exchanged between the PE routers and NEVER between the CE routers.**

**When the CE router sends an update to a PE router, the PE router prepends a 64 bit RD to the IPv4 address (32 bit address) resulting in a globally unique 96 bit address called VPNv4.**

**The PE router will then send the VPNv4 address/es via MP-BGP session to the other PE router/s. The receiving PE router strips the RD from the VPNv4 prefix, resulting in an IPv4 address.**

**Remember that the RD does NOT indicate which VRF a given prefix belongs to, it's used to make the VRF prefix/es unique within the cloud. RDs DO NOT identify the VPN.**

**Let's configure the VRFs and the RDs:**

## On R3

```
R3(config)#ip vrf CA
R3(config-vrf)#rd 1:10
```

## On R4

```
R4(config)#ip vrf CB
R4(config-vrf)#RD 1:20
```

**Since VRFs are locally significant and the RDs will keep the CE routes Unique within the cloud, and the receiving PE will strip off the 64 bit RD value, how does the receiving PE know which VRF does the IP address belong to? The answer is "Route-Target".**

**The "Route-target import|export" command defines the "RT"; An RT is a BGP extended community that indicates which routes should be exported or imported from MP-BGP into the VRF.**

**Basically RTs were introduced to support identifying a site that participates in more than one VPN.**

**The "route-target export" command specifies an RT is to be attached to every route exported from the local VRF to MP-BGP. Whereas, the "route-target import" command specifies an RT to be used as an import filter, ONLY routes matching the RT are imported into the VRF.
This implementation allows a route to have many imported or exported RTs, all to be attached to every imported or exported route.**

**Let's configure RTs:**

## On R3

```
R3(config)#ip vrf CA
R3(config-vrf)#route-target import 1:100
```

```
R3(config-vrf)#route-target export 1:100
```

**On R4**

```
R4(config)#ip vrf CB
R4(config-vrf)#route-target Both 1:100
```

**The "both" keyword is used to replace both "import" and "export" keywords.**

**But remember that the Rotue-Targets are BGP extended communities that are attached to the VPNv4 addresses, and communities are NOT sent unless they are configured to be sent.**

**Let's configure the PE routes to send the extended communities:**

**On R3**

```
R3(config)#Router bgp 65001
R3(config-router)#Address-family vpnv4 unicast
R3(config-router-af)#Neighbor 4.4.4.4 Send-community extended
```

**On R4**

```
R4(config)#Router bgp 65001
R4(config-router)#Address-family vpnv4 unicast
R4(config-router-af)#Neighbor 3.3.3.3 Send-community extended
```

**To verify the VRF Configuration:**

**On R4**

```
R4#Show ip vrf detail

VRF CB; default RD 1:20; default VPNID <not set>
  No interfaces
  Connected addresses are not in global routing table
  Export VPN route-target communities
    RT:1:100
  Import VPN route-target communities
    RT:1:100
  No import route-map
  No export route-map
  VRF label distribution protocol: not configured
  VRF label allocation mode: per-prefix
```

**Think of VRFs as VLANs; once a VLAN is configured, the appropriate interface/s are assigned to that**

**VLAN. The same philosophy applies to VRFs, you can think of VRFs as layer three VLANs, in this case, the VRFs are configured, and now the appropriate interface/s needs to be assigned to that VRF:**

## To associate an interface/s to a given VRF:

### On R4

```
R4(config)#int S0/1
R4(config-if)#IP vrf forwarding CB
```

**The above command associates an interface with the specified VRF; remember when this command is applied to a given interface, the IP address of that interface is removed and it should be reconfigured.**

**You should get the following console message:**

```
% Interface Serial0/1 IP address 10.1.45.4 removed due to enabling VRF CB

R4(config-if)#ip address 10.1.45.4 255.255.255.0
```

**Let's configure R3:**

### On R3

```
R3(config)#int S0/1
R3(config-if)#ip vrf forwarding CB

% Interface Serial0/1 IP address 10.1.13.3 removed due to enabling VRF CA

R3(config-if)#ip address 10.1.13.3 255.255.255.0
```

## To verify the configuration:

### On R4

```
R4#Show ip vrf detail

VRF CB; default RD 1:20; default VPNID <not set>
  Interfaces:
    Se0/1
  Connected addresses are not in global routing table
  Export VPN route-target communities
    RT:1:100

  Import VPN route-target communities
    RT:1:100
  No import route-map
  No export route-map
  VRF label distribution protocol: not configured
  VRF label allocation mode: per-prefix
```

**The above output begins with the name of the VRF, the default Route Distinguisher value and the VPNID which is normally not set by default. The VPNID is an extension used to further identify the VPN by using the customer "OUI" and "vpn index" that can be keyed in as a Hexadecimal or Decimal formatted number.**

**The interfaces that have the "VRF CB" applied will be listed under the Interfaces: list along with the warning that Connected addresses are no longer in the global routing table. It is a good practice to have at least 1 route target that the router will be both EXPORTING and IMPORTING. The 1:100 Route Target follows this basic design rule.**

**Currently there is no Export or Import route-maps applied. VRF-SELECT will be discussed later.**

## On R3

```
R3#Show ip vrf CA

Name                             Default RD          Interfaces
CA                               1:10                Se0/1

R3#Show ip vrf interfaces

Interface                  IP-Address      VRF                             Protocol
Se0/1                      10.1.13.3       CA                              up
```

## To verify the connectivity between PEs and the CEs:

## On R4:

```
R4#Ping 10.1.45.5

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.45.5, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```

**A regular ping will no longer work. A ping with no other key words will default to using the global routing table. The 10.1.45.0 /24 prefix is not accessible in the global routing table anymore. The ping must be added with the proper VRF keyword. Remember the IP address is in the VRF and NOT the global routing table.**

```
R4#Ping VRF CB 10.1.45.5

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.45.5, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/32 ms
```

**On R3:**

```
R3#Ping VRF CA 10.1.13.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.13.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/29 ms
```

## Task 5

Configure a static default route on each Customer router located in CA and CB; these static routes should be configured to point to their respective PE router (**R3** for R1 and **R4** for R5). The PE Routers (R3 and R4), should both be configured with a static route that reaches the loopback and F0/0 interface of the Customer router, R3 and R4 should be able to see both static routes in their BGP table.

### To configure a default route on the CEs:

**On R1**

```
R1(config)#ip route 0.0.0.0 0.0.0.0 10.1.13.3
```

**On R5**

```
R5(config)#ip route 0.0.0.0 0.0.0.0 10.1.45.4
```

**Just a regular default route from the customer's perspective. This method is one of the recommended methods that most Service Providers prefer when offering MPLS VPN as a service to a very small company.**

### To configure a static route on the PEs:

**Note the output of the following show command reveals that when the VRF forwarding was enabled on**

**S0/1 interface, the "Address-family IPv4 VRF CA" was added to the BGP configuration.**

## On R4

```
R4(config)#ip route vrf CB 5.0.0.0 255.0.0.0 10.1.45.5
R4(config)#ip route vrf CB 192.168.5.0 255.255.255.0 10.1.45.5
```

## On R3

```
R3(config)#ip route vrf CA 1.0.0.0 255.0.0.0 10.1.13.1
R3(config)#ip route vrf CA 192.168.1.0 255.255.255.0 10.1.13.1
```

## To verify and test the configuration:

## On R3

```
R3#Show ip route vrf CA | B Gateway
Gateway of last resort is not set

S     1.0.0.0/8 [1/0] via 10.1.13.1
      10.0.0.0/24 is subnetted, 1 subnets
C        10.1.13.0 is directly connected, Serial0/1
S     192.168.1.0/24 [1/0] via 10.1.13.1

R3#Ping vrf CA 1.1.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 1.1.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/28 ms

R3#Ping vrf CA 192.168.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/28 ms
```

## On R4

```
R4#Show ip route vrf CB | B Gateway
Gateway of last resort is not set

S     5.0.0.0/8 [1/0] via 10.1.45.5
S     192.168.5.0/24 [1/0] via 10.1.45.5
```

```
     10.0.0.0/24 is subnetted, 1 subnets
C       10.1.45.0 is directly connected, Serial0/1

R4#Ping vrf CB 5.5.5.5

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 5.5.5.5, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/29/32 ms

R4#Ping vrf CB 192.168.5.5

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.5.5, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/56/60 ms
```

**In the last step of this task, the static routes that were configured on the PE routers should be redistributed into BGP so the neighboring PE router can see the route in its BGP table. This redistribution should be done under another address-family called "IPv4 Unicast"; this address-family is created automatically when the VRF, RD and the RTs are configured; the following show command verifies this fact:**

**On R4**

```
R4#Sh run | B router bgp

router bgp 65001
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 neighbor 3.3.3.3 remote-as 65001
 neighbor 3.3.3.3 update-source Loopback0
 !
 address-family vpnv4
  neighbor 3.3.3.3 activate
  neighbor 3.3.3.3 send-community extended
 exit-address-family
 !
 address-family ipv4 vrf CB
  no synchronization
 exit-address-family
```

**On R3**

```
R3#Sh run | B router bgp
```

```
router bgp 65001
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 neighbor 4.4.4.4 remote-as 65001
 neighbor 4.4.4.4 update-source Loopback0
 !
 address-family vpnv4
  neighbor 4.4.4.4 activate
  neighbor 4.4.4.4 send-community extended
 exit-address-family
 !
 address-family ipv4 vrf CA
  no synchronization
 exit-address-family
```

**To configure the redistribution of the static routes:**

## On R3

```
R3(config)#Router bgp 65001
R3(config-router)#Address-family IPv4 VRF CA
R3(config-router-af)#Redistribute static
```

**Note the redistributed routes should be verified on R4, if the configuration is performed successfully, R4 should be able to see networks 1.0.0.0 /8 and 192.168.1.0 /24 in its vrf CB routing table.**

## On R4

```
R4#Show ip route vrf CB | B Gateway
Gateway of last resort is not set

B    1.0.0.0/8 [200/0] via 3.3.3.3, 00:02:51
S    5.0.0.0/8 [1/0] via 10.1.45.5
S    192.168.5.0/24 [1/0] via 10.1.45.5
     10.0.0.0/24 is subnetted, 1 subnets
C       10.1.45.0 is directly connected, Serial0/1
B    192.168.1.0/24 [200/0] via 3.3.3.3, 00:02:51
```

**NOTE: The redistributed routes show up on R4 as BGP routes. Let's configure R4 to redistribute its static routes:**

## On R4

```
R4(config)#Router bgp 65001
R4(config-router)#Address-family IPv4 vrf CB
```

```
R4(config-router-af)#Redistribute static
```

**To verify the configuration:**

**On R3**

```
R3#Show ip route vrf CA | b Gateway
Gateway of last resort is not set

S    1.0.0.0/8 [1/0] via 10.1.13.1
B    5.0.0.0/8 [200/0] via 4.4.4.4, 00:00:27
B    192.168.5.0/24 [200/0] via 4.4.4.4, 00:00:27
     10.0.0.0/24 is subnetted, 1 subnets
C       10.1.13.0 is directly connected, Serial0/1
S    192.168.1.0/24 [1/0] via 10.1.13.1
```

**To verify the routes in VPNv4:**

**On R3**

```
R3#Show ip bgp vpnv4 all | B Network

   Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 1:10 (default for vrf CA)
*> 1.0.0.0          10.1.13.1                0         32768 ?
*>i5.0.0.0          4.4.4.4                  0    100      0 ?
*> 192.168.1.0      10.1.13.1                0         32768 ?
*>i192.168.5.0      4.4.4.4                  0    100      0 ?
Route Distinguisher: 1:20
*>i5.0.0.0          4.4.4.4                  0    100      0 ?
*>i192.168.5.0      4.4.4.4                  0    100      0 ?
```

**On R4**

```
R4#Show ip bgp vpnv4 all | B Network

   Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 1:10
*>i1.0.0.0          3.3.3.3                  0    100      0 ?
*>i192.168.1.0      3.3.3.3                  0    100      0 ?
Route Distinguisher: 1:20 (default for vrf CB)
*>i1.0.0.0          3.3.3.3                  0    100      0 ?
*> 5.0.0.0          10.1.45.5                0         32768 ?
*>i192.168.1.0      3.3.3.3                  0    100      0 ?
*> 192.168.5.0      10.1.45.5                0         32768 ?
```

**One of the reasons to use different Route Distinguishers for ALL VRFs in the network is for clarity, especially when trying to figure out which VPN a route belongs to and where it's originated. As you can see, the "1:20" RD is local, but the 1.0.0.0 & 192.168.1.0 networks are present in both RDs, which means that 1.0.0.0 & 192.168.1.0 networks are participating in the same VPN. Even though the VPN is defined by the Route Target, it is still apparent that the prefixes are being imported and exported into both VRFs.**

## To test the configuration:

### On R1

```
R1#Ping 5.5.5.5 source 1.1.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 5.5.5.5, timeout is 2 seconds:
Packet sent with a source address of 1.1.1.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/56/60 ms

R1#Ping 5.5.5.5 source 192.168.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 5.5.5.5, timeout is 2 seconds:
Packet sent with a source address of 192.168.1.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 84/84/88 ms
```

### On R5

```
R5#Ping 1.1.1.1 source 5.5.5.5

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 = 56/56/57 ms

R5#Ping 1.1.1.1 source 192.168.5.5

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 192.168.5.5

!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 84/84/84 ms
```

**Note the source of the two ping commands are specified to be the loopback 0 & F0/0's IP address, if the source is NOT specified, the Ping will NOT be successful:**

## On R5

```
R5#Ping 1.1.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 1.1.1.1, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```

## On R1

```
R1#Ping 5.5.5.5

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 5.5.5.5, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```

**The ping failed because the source was NOT set based on the loopback0 or the F0/0 interface, if the source is not specified, the source IP address will be the closest interface, in this case the IP address of the S0/1 interface. In order to provide the capability of Pinging without specifying the source IP address, the PE routers should also redistribute the IP address of their S0/1 interface, this can be done using the "Redistribute Connected":**

## On R3

```
R3(config)#router bgp 65001
R3(config-router)#Address-family IPv4 vrf CA
R3(config-router-af)#Redistribute connected
```

## To verify the configuration:

## On R4

```
R4#Show ip route vrf CB | B Gateway
Gateway of last resort is not set

B    1.0.0.0/8 [200/0] via 3.3.3.3, 00:21:41
S    5.0.0.0/8 [1/0] via 10.1.45.5
S    192.168.5.0/24 [1/0] via 10.1.45.5
     10.0.0.0/24 is subnetted, 2 subnets
```

```
B       10.1.13.0 [200/0] via 3.3.3.3, 00:00:26
C       10.1.45.0 is directly connected, Serial0/1
B    192.168.1.0/24 [200/0] via 3.3.3.3, 00:21:41
```

**On R4**

```
R4(config)#router bgp 65001
R4(config-router)#Address-family IPv4 vrf CB
R4(config-router-af)#Redistribute connected
```

**To verify the configuration:**

**On R3**

```
R3#Show ip route vrf CA | B Gateway
Gateway of last resort is not set

S    1.0.0.0/8 [1/0] via 10.1.13.1
B    5.0.0.0/8 [200/0] via 4.4.4.4, 00:17:30
B    192.168.5.0/24 [200/0] via 4.4.4.4, 00:17:30
     10.0.0.0/24 is subnetted, 2 subnets
C       10.1.13.0 is directly connected, Serial0/1
B       10.1.45.0 [200/0] via 4.4.4.4, 00:00:45
S    192.168.1.0/24 [1/0] via 10.1.13.1
```

**To test the connectivity:**

**On R1**

```
R1#Ping 5.5.5.5

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 5.5.5.5, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/57/60 ms
```

**On R5**

```
R5#Ping 1.1.1.1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 1.1.1.1, timeout is 2 seconds:
!!!!!
```

```
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/56/57 ms
```

**To verify the extended community attached to a given prefix:**

**On R4**

```
R4#Show ip bgp VPNv4 all 5.0.0.0

BGP routing table entry for 1:20:5.0.0.0/8, version 8
Paths: (1 available, best #1, table CB)
  Advertised to update-groups:
        1
 Local
  10.1.45.5 from 0.0.0.0 (4.4.4.4)
   Origin incomplete, metric 0, localpref 100, weight 32768, valid, sourced,
best
      Extended Community: RT:1:100
      mpls labels in/out 403/nolabel
```

**We have successfully verified that the route-target has been added to the routes as they are exported from the local VRF. This is important as NO vrf will be able to import this route if the extended community is not added to the prefix. The VPNv4 label has also been added to the prefix. The "403(in) and the nolabel(out)" shows that this router expects to see this route with label 403 when it recieves the data traffic and will remove the label before forwarding it to the Customer (R5) router.**

```
R4#Show ip bgp VPNv4 all labels

   Network          Next Hop      In label/Out label
Route Distinguisher: 1:10
   1.0.0.0          3.3.3.3         nolabel/303
   10.1.13.0/24     3.3.3.3         nolabel/305
   192.168.1.0      3.3.3.3         nolabel/304
Route Distinguisher: 1:20 (CB)
   1.0.0.0          3.3.3.3         nolabel/303
   5.0.0.0          10.1.45.5       403/nolabel
   10.1.13.0/24     3.3.3.3         nolabel/305
   10.1.45.0/24     0.0.0.0         405/aggregate(CB)
   192.168.1.0      3.3.3.3         nolabel/304
   192.168.5.0      10.1.45.5       404/nolabel
```

**This command displays the VPNv4 labels that are added to the routes. A common question that is often asked is "What does the aggregate keyword mean?" The aggregate keyword means that the route requires an IP lookup to determine the next hop of this packet. When we redistributed the connected routes, the far end does not know about this next hop due to the fact that this route is not in the VPN.**

**By redistributing the connected route, a normal ping from the interface closest to the destination can**

**be performed by R1 due to the fact that R5 now knows about the source of the packet.**

```
R4#Sh mpls forwarding-table vrf CB

Local  Outgoing    Prefix            Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
403    Untagged    5.0.0.0/8[V]      2080       Se0/1      point2point
404    Untagged    192.168.5.0/24[V] 520        Se0/1      point2point
405    Aggregate   10.1.45.0/24[V]   520
```

**On R3**

```
R3#Show mpls forwarding-table vrf CA

Local  Outgoing    Prefix            Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
303    Untagged    1.0.0.0/8[V]      2600       Se0/1      point2point
304    Untagged    192.168.1.0/24[V] 520        Se0/1      point2point
305    Aggregate   10.1.13.0/24[V]   520
```

## Task 6

Remove the Static routes and replace the current method of routing between the PE and Customers with RIPv2 Routing.

### Removing the Static Configuration on the CE routers:

**On R1**

```
R1(config)#NO ip route 0.0.0.0 0.0.0.0
```

**On R5**

```
R5(config)#NO ip route 0.0.0.0 0.0.0.0
```

### Removing the Static Configuration on the PE routers:

**On R3**

```
R3(config)#NO ip route vrf CA 1.0.0.0 255.0.0.0
R3(config)#NO ip route vrf CA 192.168.1.0 255.255.255.0
```

```
R3(config)#router bgp 65001
R3(config-router)# address ipv4 vrf CA
R3(config-router-af)#NO redistribute static
R3(config-router-af)#NO redistribute connected
```

**On R4**

```
R4(config)#NO ip route vrf CB 5.0.0.0 255.0.0.0
R4(config)#NO ip route vrf CB 192.168.2.0 255.255.255.0

R4(config)#router bgp 65001
R4(config-router)#address ipv4 vrf CB
R4(config-router-af)#NO redistribute static
R4(config-router-af)#NO redistribute connected
```

## Configuring RIPv2 routing on CE routers:

**On R1**

```
R1(config)#router rip
R1(config-router)#No au
R1(config-router)#ver 2
R1(config-router)#netw 10.0.0.0
R1(config-router)#netw 192.168.1.0
R1(config-router)#netw 1.0.0.0
```

**On R5**

```
R5(config)#router rip
R5(config-router)#No au
R5(config-router)#ver 2
R5(config-router)#network 10.0.0.0
R5(config-router)#netw 5.0.0.0
R5(config-router)#netw 192.168.5.0
```

## Configuring RIPv2 routing on PE routers:

**On R3**

```
R3(config)#router rip
R3(config-router)#ver 2
R3(config-router)#Address-family ipv4 vrf CA

R3(config-router-af)#Network 10.0.0.0
R3(config-router-af)#version 2
```

```
R3(config-router-af)#No au
```

**On R4**

```
R4(config)#router rip
R4(config-router)#ver 2

R4(config-router)#Address-family ipv4 vrf CB
R4(config-router-af)#Network 10.0.0.0
R4(config-router-af)#No au
R4(config-router-af)#ver 2
```

**To verify the configuration:**

**On R3**

```
R3#Show ip route vrf CA | B Gateway
Gateway of last resort is not set

R    1.0.0.0/8 [120/1] via 10.1.13.1, 00:00:20, Serial0/1
     10.0.0.0/24 is subnetted, 1 subnets
C       10.1.13.0 is directly connected, Serial0/1
R    192.168.1.0/24 [120/1] via 10.1.13.1, 00:00:20, Serial0/1
```

**On R4**

```
R4#Show ip route vrf CB | B Gateway
Gateway of last resort is not set

R    5.0.0.0/8 [120/1] via 10.1.45.5, 00:00:23, Serial0/1
R    192.168.5.0/24 [120/1] via 10.1.45.5, 00:00:23, Serial0/1
     10.0.0.0/24 is subnetted, 1 subnets
C       10.1.45.0 is directly connected, Serial0/1
```

**Note the routes are in the appropriate VRFs, the next step is to redistribute the routes.**

**The redistribution of RIPv2 into MP-BGP is necessary for the routes to show up on the other PE Router.**

**The redistribution of MP-BGP into RIP is necessary for the local router to translate the routes that have been received by the remote PE router so the CE router/s can see the routes.**

## Step One

**RIP routes are redistributed into the BGP for the specified VRF:**

## On R4

```
R4(config)#router bgp 65001
R4(config-router)#Address-family ipv4 vrf CB
R4(config-router-af)#Redistribute RIP
```

## To verify the configuration:

**Because of the redistribution, the PE router (R3) on the other side can now see the routes in its BGP table:**

## On R3

```
R3#Show ip bgp vpnv4 vrf CA | B Network

   Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 1:10 (default for vrf CA)
*>i5.0.0.0          4.4.4.4                  1    100      0 ?
*>i10.1.45.0/24     4.4.4.4                  0    100      0 ?
*>i192.168.5.0      4.4.4.4                  1    100      0 ?
```

## Step Two

**In this step the routes are redistributed into RIP, so the CE router will have them in its routing table:**

## On R3

**When the routes are redistributed into RIP, a metric of 5 is assigned; this is done on purpose so they can easily be identified in the routing table of the Customer router (R1):**

```
R3(config)#router rip
R3(config-router)#Address-family ipv4 vrf CA
R3(config-router-af)#redistribute BGP 65001 metric 5
```

## To verify the configuration:

## On R1

**Note R1 (The CE router) has the routes in its routing table:**

```
R1#Show ip route rip | Inc R

R    5.0.0.0/8 [120/5] via 10.1.13.3, 00:00:08, Serial0/1
R    192.168.5.0/24 [120/5] via 10.1.13.3, 00:00:08, Serial0/1
```

```
R       10.1.45.0 [120/5] via 10.1.13.3, 00:00:08, Serial0/1
```

**The same needs to be done in reverse order from R3 to R4 and verified on R5:**

## On R3

```
R3(config)#router bgp 65001
R3(config-router)#Address-family ipv4 vrf CA
R3(config-router-af)#Redistribute RIP
```

## On R4

```
R4(config)#Router rip
R4(config-router)#Address-family ipv4 vrf CB
R4(config-router-af)#Redistribute BGP 65001 metric 5
```

## To verify the configuration:

## On R5

```
R5#Show ip route rip | Inc R

R    1.0.0.0/8 [120/5] via 10.1.45.4, 00:00:05, Serial0/1
R       10.1.13.0 [120/5] via 10.1.45.4, 00:00:05, Serial0/1
R    192.168.1.0/24 [120/5] via 10.1.45.4, 00:00:05, Serial0/1
```

**To see the full picture, the following show commands verifies network 1.0.0.0 /8 from R1 all the way to the upstream router R5; this is called the control plane.**

## On R1

```
R1#Show ip route 1.0.0.0

Routing entry for 1.0.0.0/8
  Known via "connected", distance 0, metric 0 (connected, via interface)
  Redistributing via rip
  Advertised by rip
  Routing Descriptor Blocks:
  * directly connected, via Loopback0
      Route metric is 0, traffic share count is 1

R1#Show ip rip database 1.0.0.0 255.0.0.0

1.0.0.0/8    directly connected, Loopback0
```

**On R3**

```
R3#Show ip route vrf CA | B Gateway
Gateway of last resort is not set

R    1.0.0.0/8 [120/1] via 10.1.13.1, 00:00:04, Serial0/1
B    5.0.0.0/8 [200/1] via 4.4.4.4, 00:29:36
B    192.168.5.0/24 [200/1] via 4.4.4.4, 00:29:36
     10.0.0.0/24 is subnetted, 2 subnets
C       10.1.13.0 is directly connected, Serial0/1
B       10.1.45.0 [200/0] via 4.4.4.4, 00:29:36
R    192.168.1.0/24 [120/1] via 10.1.13.1, 00:00:04, Serial0/1

R3#Show ip route vrf CA 1.0.0.0

Routing entry for 1.0.0.0/8
  Known via "rip", distance 120, metric 1
  Redistributing via bgp 65001, rip
  Advertised by bgp 65001
  Last update from 10.1.13.1 on Serial0/1, 00:00:11 ago
  Routing Descriptor Blocks:
  * 10.1.13.1, from 10.1.13.1, 00:00:11 ago, via Serial0/1
      Route metric is 1, traffic share count is 1

R3#Show ip bgp vpnv4 vrf CA | B Network

   Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 1:10 (default for vrf CA)
*> 1.0.0.0          10.1.13.1                1         32768 ?
*>i5.0.0.0          4.4.4.4                  1    100      0 ?
*> 10.1.13.0/24     0.0.0.0                  0         32768 ?
*>i10.1.45.0/24     4.4.4.4                  0    100      0 ?
*> 192.168.1.0      10.1.13.1                1         32768 ?
*>i192.168.5.0      4.4.4.4                  1    100      0 ?

R3#Show ip bgp VPNV4 vrf CA 1.0.0.0/8

BGP routing table entry for 1:10:1.0.0.0/8, version 32
Paths: (1 available, best #1, table CA)
  Advertised to update-groups:
        1
  Local
    10.1.13.1 from 0.0.0.0 (3.3.3.3)

      Origin incomplete, metric 1, localpref 100, weight 32768, valid, sourced,
best
      Extended Community: RT:1:100
      mpls labels in/out 303/nolabel
```

**Note the Origin code is (incomplete) because the route was redistributed and therefore, the origin of the route is unknown. The Weight attribute is 32768, because the local router is advertising the route in BGP. The extended community is identified, this is the configured route-target, and the last line states that if the local router receives a packet with a label of 303 it will remove the label and forward it.**

## On R4

```
R4#Show ip bgp vpnv4 vrf CB 1.0.0.0

BGP routing table entry for 1:20:1.0.0.0/8, version 32
Paths: (1 available, best #1, table CB)
  Not advertised to any peer
  Local, imported path from 1:10:1.0.0.0/8
    3.3.3.3 (metric 3) from 3.3.3.3 (3.3.3.3)
      Origin incomplete, metric 1, localpref 100, valid, internal, best
      Extended Community: RT:1:100
      mpls labels in/out nolabel/303

R4#Show ip bgp vpnv4 vrf CB | B Network

   Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 1:20 (default for vrf CB)
*>i1.0.0.0          3.3.3.3                  1    100      0 ?
*> 5.0.0.0          10.1.45.5                1         32768 ?
*>i10.1.13.0/24     3.3.3.3                  0    100      0 ?
*> 10.1.45.0/24     0.0.0.0                  0         32768 ?
*>i192.168.1.0      3.3.3.3                  1    100      0 ?
*> 192.168.5.0      10.1.45.5                1         32768 ?

R4#Show ip route vrf CB | B Gateway
Gateway of last resort is not set

B    1.0.0.0/8 [200/1] via 3.3.3.3, 00:11:50
R    5.0.0.0/8 [120/1] via 10.1.45.5, 00:00:05, Serial0/1
R    192.168.5.0/24 [120/1] via 10.1.45.5, 00:00:05, Serial0/1
     10.0.0.0/24 is subnetted, 2 subnets
B       10.1.13.0 [200/0] via 3.3.3.3, 00:11:50
C       10.1.45.0 is directly connected, Serial0/1
B    192.168.1.0/24 [200/1] via 3.3.3.3, 00:11:50

R4#Sh ip route vrf CB 1.0.0.0

Routing entry for 1.0.0.0/8
  Known via "bgp 65001", distance 200, metric 1, type internal
  Redistributing via rip
  Advertised by rip metric 5
  Last update from 3.3.3.3 00:13:11 ago
  Routing Descriptor Blocks:
  * 3.3.3.3 (Default-IP-Routing-Table), from 3.3.3.3, 00:13:11 ago
      Route metric is 1, traffic share count is 1
      AS Hops 0
```

**On R5**

```
R5#Show ip route rip

R    1.0.0.0/8 [120/5] via 10.1.45.4, 00:00:04, Serial0/1
R       10.1.13.0 [120/5] via 10.1.45.4, 00:00:04, Serial0/1
R    192.168.1.0/24 [120/5] via 10.1.45.4, 00:00:04, Serial0/1
```

**To test the configuration:**

**On R5**

```
R5#Ping 1.1.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 1.1.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/56/57 ms
```

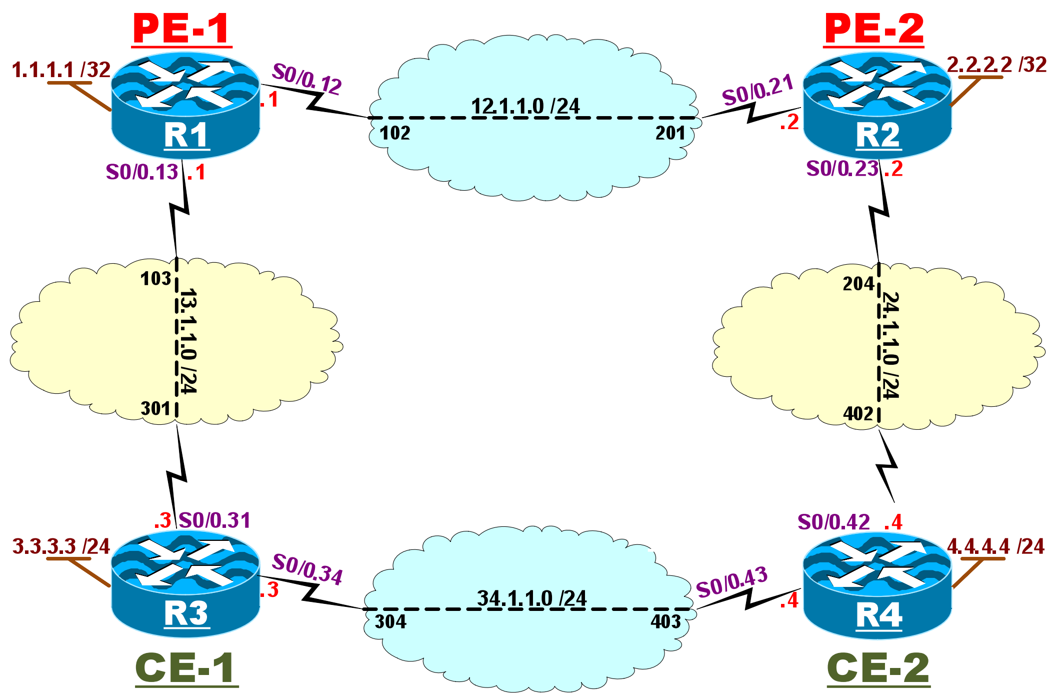
**On R1**

```
R1#Ping 5.5.5.5

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 5.5.5.5, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/56/60 ms
```

## Task 7

**Stop all routers in the console window and exit. Stop the Server and press the "Erase Start" launcher before proceeding.**

# LAB 2 - EIGRP-SOO



| Router | Interface | IP address |
|---|---|---|
| R1 | Loopback 0<br>S0/0.13<br>S0/0.12 | 1.1.1.1 /32<br>13.1.1.1 /24<br>12.1.1.1 /24 |
| **R2** | **Loopback 0**<br>**S0/0.21**<br>**S0/0.24** | **2.2.2.2 /32**<br>**12.1.1.2 /24**<br>**24.1.1.2 /24** |
| R3 | Lo0<br>S0/0.31<br>S0/0.34 | 3.3.3.3/ 32<br>13.1.1.3 /24<br>34.1.1.3 /24 |
| **R4** | **Lo0**<br>**S0/0.42**<br>**S0/0.43** | **4.4.4.4 /32**<br>**24.1.1.4 /24**<br>**34.1.1.4 /24** |

## Lab Setup:

All frame-relay interfaces must be configured in a point-to-point manner.

## Task 1

Configure OSPF area 0 on S0/0.12, and S0/0.21 sub-interfaces of R1 and R2 and their Loopback 0 interfaces. The Loopback interfaces of R3 and R4, the S0/0.34 and S0/0.43 of R3 and R4 should be configured in Eigrp 100.

### On R1

```
R1(config)#Router ospf 1
R1(config-router)#router-id 0.0.0.1
R1(config-router)#Netw 1.1.1.1 0.0.0.0 area 0
R1(config-router)#Netw 12.1.1.1 0.0.0.0 area 0
```

### On R2

```
R2(config)#Router ospf 1
R2(config-router)#router-id 0.0.0.2
R2(config-router)#Netw 2.2.2.2 0.0.0.0 area 0
R2(config-router)#Netw 12.1.1.2 0.0.0.0 area 0
```

### To verify the configuration:

### On R1

```
R1#Show ip ospf neighbor

Neighbor ID     Pri   State           Dead Time   Address         Interface
0.0.0.2           0   FULL/  -        00:00:36    12.1.1.2        Serial0/0.12
```

### On R3

```
R3(config)#Router eigrp 100
R3(config-router)#No au
R3(config-router)#Network 34.1.1.3 0.0.0.0
R3(config-router)#Network 3.3.3.3 0.0.0.0
```

### On R4

```
R4(config)#Router eigrp 100
R4(config-router)#No au
R4(config-router)#Network 34.1.1.4 0.0.0.0
R4(config-router)#Network 4.4.4.4 0.0.0.0
```

**To verify the configuration:**

**On R3**

```
R3#Show ip route eigrp | i D

D       4.4.4.0 [90/2297856] via 34.1.1.4, 00:02:11, Serial0/0.34
```

## Task 2

Configure the core routers (R1 and R2) to support MPLS VPN using AS 65012 using their loopback 0 interfaces. Use the following parameters for VRF configuration:

| | |
|---|---|
| VRF Name | TST |
| RD on R1 | 1:10 |
| RD on R2 | 2:10 |
| Route-target | 34:34 |
| PE-CE Routing Protocol | Eigrp 100 |

Ensure full connectivity between customer's (R3 and R4) routes.

**On R1**

```
R1(config)#Router bgp 65012
R1(config-router)#No au
R1(config-router)#Neighbor 2.2.2.2 remote-as 65012
R1(config-router)#Neighbor 2.2.2.2 update-source Lo0

R1(config-router)#Address-family vpnv4 unicast
R1(config-router-af)#Neighbor 2.2.2.2 activate
R1(config-router-af)#Neighbor 2.2.2.2 send-community extended
```

**On R2**

```
R2(config)#router bgp 65012
R2(config-router)#No au
```

```
R2(config-router)#Neighbor 1.1.1.1 remote-as 65012
R2(config-router)#Neighbor 1.1.1.1 update-source Lo0

R2(config-router)#Address-family vpnv4 unicast
R2(config-router-af)#Neighbor 1.1.1.1 activate
R2(config-router-af)#Neighbor 1.1.1.1 send-community extended
```

**To verify the configuration:**

**On R2**

```
R2#Show bgp ipv4 unicast summary | B Neighbor

Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
1.1.1.1         4 65012       9       9        1    0    0 00:01:44        0
```

**On R1**

```
R1(config)#MPLS label protocol ldp
R1(config)#MPLS ldp router-id Lo0 force

R1(config)#IP vrf TST
R1(config-vrf)#rd 1:10
R1(config-vrf)#route-target both 34:34

R1(config)#Int S0/0.12
R1(config-subif)#MPLS IP

R1(config)#Int S0/0.13
R1(config-subif)#IP vrf forwarding TST
R1(config-subif)#IP address 13.1.1.1 255.255.255.0

R1(config)#Router eigrp 100
R1(config-router)#Address-family ipv4 vrf TST
R1(config-router-af)#autonomous-system 100
R1(config-router-af)#No au
R1(config-router-af)#Netw 13.1.1.1 0.0.0.0
R1(config-router-af)#Redistribute bgp 65012 metric 1 1 1 1 1

R1(config)#Router bgp 65012
R1(config-router)#Address-family ipv4 vrf TST
R1(config-router-af)#redistribute eigrp 100
```

**On R2**

```
R2(config)#MPLS label protocol ldp
R2(config)#MPLS ldp router-id Lo0 force

R2(config)#IP vrf TST
R2(config-vrf)#rd 2:10
R2(config-vrf)#route-target both 34:34

R2(config)#Int S0/0.21
R2(config-subif)#MPLS IP

R2(config)#Int S0/0.24
R2(config-subif)#IP vrf forwarding TST
R2(config-subif)#IP address 24.1.1.2 255.255.255.0

R2(config)#Router eigrp 100
R2(config-router)#Address-family ipv4 vrf TST
R2(config-router-af)#autonomous-system 100
R2(config-router-af)#No au
R2(config-router-af)#Network 24.1.1.2 0.0.0.0
R2(config-router-af)#redistribute bgp 65012 metric 1 1 1 1 1

R2(config)#Router bgp 65012
R2(config-router)#Address-family ipv4 vrf TST
R2(config-router-af)#redistribute eigrp 100
```

## On R3

```
R3(config)#Router eigrp 100
R3(config-router)#Network 13.1.1.3 0.0.0.0
```

## On R4

```
R4(config)#Router eigrp 100
R4(config-router)#Network 24.1.1.4 0.0.0.0
```

## To test and verify the configuration:

## On R3

```
R3#Show ip route Eigrp

     4.0.0.0/24 is subnetted, 1 subnets
D       4.4.4.0 [90/2297856] via 34.1.1.4, 00:01:12, Serial0/0.34
     24.0.0.0/24 is subnetted, 1 subnets
D       24.1.1.0 [90/2681856] via 34.1.1.4, 00:01:12, Serial0/0.34
```

```
                 [90/2681856] via 13.1.1.1, 00:01:12, Serial0/0.31
```

**On R4**

```
R4#Show ip route Eigrp

     3.0.0.0/24 is subnetted, 1 subnets
D       3.3.3.0 [90/2297856] via 34.1.1.3, 00:00:23, Serial0/0.43
     13.0.0.0/24 is subnetted, 1 subnets
D       13.1.1.0 [90/2681856] via 34.1.1.3, 00:00:23, Serial0/0.43
                 [90/2681856] via 24.1.1.2, 00:00:23, Serial0/0.42
```

**NOTE: The cost to Network 13.1.1.0/24 through R3 is identical to the cost through the MPLS cloud, this is because the configured metric under router bgp is NOT looked at, and the cost R4 to R2, and R1 to R3 for network 13.1.1.0/24 is calculated. But the cost of 3.3.3.0/24 network is lower through R3 than the MPLS cloud, this is because "the delay of the serial link to R3, plus the delay of the loopback 0 interface of R3 divided by ten is lower than, the delay of the serial link to R2, plus the delay of the serial link from R1 to R3 plus the delay of the loopback 0 interface divided by ten.**
**But if the link to R3 is down, the network is reachable through the MPLS cloud, let's test this:**

**On R3**

```
R3#Show ip route eigrp | i 4.4.4.0

D       4.4.4.0 [90/2297856] via 34.1.1.4, 00:30:57, Serial0/0.34

R3#Show int lo0 | Inc DLY

  MTU 1514 bytes, BW 8000000 Kbit/sec, DLY 5000 usec,
```

**Let's calculate the cost:**

**Basically it's 256(BW + DLY), but to calculate BW, the reference BW is divided by the slowest BW along the path to the detination, therefore, BW will be:**

**10,000,000/1544=6476.68 → Drop the fractions, so 6476 is the BW used in the formula.**

**The delay is calculated based on sum of all delays divided by ten:**

```
R3#Show int s0/0.34 | Inc DLY

  MTU 1500 bytes, BW 1544 Kbit/sec, DLY 20000 usec,
```

**Therefore:**

**20000+5000=25000/10=2500**

**Therefore:**

**256(2500+6476)=2,297,856**

**Now……let's shut down the Serial link to R4:**

## On R3

```
R3(config)#Int S0/0.34
R3(config-subif)#Shut

R3#Show ip route eigrp | i 4.4.4.0

D       4.4.4.0 [90/2809856] via 13.1.1.1, 00:00:24, Serial0/0.31
```

**Let's calculate the metric through the cloud:**

```
R3#Show int s0/0.31 | Inc DLY

  MTU 1500 bytes, BW 1544 Kbit/sec, DLY 20000 usec,
```

**Since the frame-relay sub-interfaces from R3 to R1, and R2 to R4 is the same, the BW and the DLY values will be identical.**

**Therefore:**

**To calculate the delay:**
**20,000+20,000+5000=45000/10=4500**

**We know that the lowest BW along the path to network 4.4.4.0/24 is 1544, therefore:**
**256(4500+6476)=2,809,856**

## Task 3

Configure the appropriate routers to eliminate the local routes from being learned from the backbone.
DO NOT configure R3 or R4, access-lists, or prefix-list to accomplish this task.

**In this case an extended community of Site of origin (SoO) can be used, SoO is set in a route-map and the route-map is referenced by the "ip vrf sitemap" interface configuration command.**

**Let's configure this extended community:**

## On R1

**NOTE: This extended community is also a 64 bit value (32bits:32bits):**

```
R1(config)#Route-map SOO permit 10
R1(config-route-map)#set extcommunity soo 1:111

R1(config)#Int S0/0.13
R1(config-subif)#IP vrf sitemap SOO
```

**NOTE: Since the extended community is applied to an interface and the route-map is not referencing an ACL or an IP prefix-list, this extended community is applied to all routes received through the S0/0.13 sub-interface: Let's verify:**

**R1 assigns the extended community to all routes it receives from R3, and advertises them to its IBGP neighbor:**

## On R2

```
R2#Show ip eigrp vrf TST topology 13.1.1.0/24

IP-EIGRP (AS 100): Topology entry for 13.1.1.0/24
  State is Passive, Query origin flag is 1, 1 Successor(s), FD is 2169856
  Routing Descriptor Blocks:
  1.1.1.1, from VPNv4 Sourced, Send flag is 0x0
      Composite metric is (2169856/0), Route is Internal (VPNv4 Sourced)
      Vector metric:
        Minimum bandwidth is 1544 Kbit
        Total delay is 20000 microseconds
        Reliability is 255/255
        Load is 1/255
        Minimum MTU is 1500
        Hop count is 0
      Extended Community: SoO:1:111
```

**R2 advertises it to R4:**

## On R4

```
R4#Show ip eigrp topology 13.1.1.0/24

IP-EIGRP (AS 100): Topology entry for 13.1.1.0/24
  State is Passive, Query origin flag is 1, 2 Successor(s), FD is 2681856
  Routing Descriptor Blocks:
  24.1.1.2 (Serial0/0.42), from 24.1.1.2, Send flag is 0x0
```

```
      Composite metric is (2681856/2169856), Route is Internal
      Vector metric:
        Minimum bandwidth is 1544 Kbit
        Total delay is 40000 microseconds
        Reliability is 255/255
        Load is 1/255
        Minimum MTU is 1500
        Hop count is 1
      Extended Community: SoO:1:111
  34.1.1.3 (Serial0/0.43), from 34.1.1.3, Send flag is 0x0
      Composite metric is (2681856/2169856), Route is Internal
      Vector metric:
        Minimum bandwidth is 1544 Kbit
        Total delay is 40000 microseconds
        Reliability is 255/255
        Load is 1/255
        Minimum MTU is 1500
        Hop count is 1
```

**Let's configure R2:**

## On R2

```
R2(config)#Route-map SOO permit 10
R2(config-route-map)#set extcommunity soo 2:222

R2(config)#Int S0/0.24
R2(config-subif)#IP vrf sitemap SOO
```

## To verify the configuration:

## On R1

```
R1#Show ip eigrp vrf TST topology 24.1.1.0/24

IP-EIGRP (AS 100): Topology entry for 24.1.1.0/24
  State is Passive, Query origin flag is 1, 1 Successor(s), FD is 2169856
  Routing Descriptor Blocks:
  2.2.2.2, from VPNv4 Sourced, Send flag is 0x0
      Composite metric is (2169856/0), Route is Internal (VPNv4 Sourced)
      Vector metric:
        Minimum bandwidth is 1544 Kbit
        Total delay is 20000 microseconds
        Reliability is 255/255
        Load is 1/255
        Minimum MTU is 1500
        Hop count is 0
      Extended Community: SoO:2:222
```

**On R3**

```
R3#Show ip eigrp topology 24.1.1.0/24

IP-EIGRP (AS 100): Topology entry for 24.1.1.0/24
  State is Passive, Query origin flag is 1, 2 Successor(s), FD is 2681856
  Routing Descriptor Blocks:
  13.1.1.1 (Serial0/0.31), from 13.1.1.1, Send flag is 0x0
      Composite metric is (2681856/2169856), Route is Internal
      Vector metric:
        Minimum bandwidth is 1544 Kbit
        Total delay is 40000 microseconds
        Reliability is 255/255
        Load is 1/255
        Minimum MTU is 1500
        Hop count is 1
      Extended Community: SoO:2:222
  34.1.1.4 (Serial0/0.34), from 34.1.1.4, Send flag is 0x0
      Composite metric is (2681856/2169856), Route is Internal
      Vector metric:
        Minimum bandwidth is 1544 Kbit
        Total delay is 40000 microseconds
        Reliability is 255/255
        Load is 1/255
        Minimum MTU is 1500
        Hop count is 1
```

**Let's test the configuration:**

**On R3**

```
R3#Show ip route Eigrp

     4.0.0.0/24 is subnetted, 1 subnets
D       4.4.4.0 [90/2297856] via 34.1.1.4, 00:03:45, Serial0/0.34
     24.0.0.0/24 is subnetted, 1 subnets
D       24.1.1.0 [90/2681856] via 34.1.1.4, 00:03:29, Serial0/0.34
                 [90/2681856] via 13.1.1.1, 00:03:29, Serial0/0.31
```

**This did not work. Well.. the reason it did not work is because the extended community values configured on the PEs did not match, when a PE receives a route from another PE, it checks the SoO**

**extended community value of the received route, if it matches to what it has configured locally for the same route, it will not advertise it. In this case, the extended community tags have different values (1:111 on R1 and 2:222 on R2), let's change the route-map configured on R1 and R2 such that the extended community values match.**

## On R1

**In this case an extended community value of 1:121 is configured, but you can choose any extended community value, as long as they match.**

```
R1(config)#Route-map SOO permit 10
R1(config-route-map)#No set extcommunity soo 1:111
R1(config-route-map)#set extcommunity soo 1:121
```

## On R2

```
R2(config)#Route-map SOO permit 10
R2(config-route-map)#No set extcommunity soo 2:222
R2(config-route-map)#set extcommunity soo 1:121
```

## Let's verify the configuration:

## On R3

```
R3#Show ip route eigrp

     4.0.0.0/24 is subnetted, 1 subnets
D       4.4.4.0 [90/2297856] via 34.1.1.4, 00:02:06, Serial0/0.34
     24.0.0.0/24 is subnetted, 1 subnets
D       24.1.1.0 [90/2681856] via 34.1.1.4, 00:00:18, Serial0/0.34
```

## On R4

```
R4#Show ip route eigrp

     3.0.0.0/24 is subnetted, 1 subnets
D       3.3.3.0 [90/2297856] via 34.1.1.3, 00:02:19, Serial0/0.43
     13.0.0.0/24 is subnetted, 1 subnets
D       13.1.1.0 [90/2681856] via 34.1.1.3, 00:00:50, Serial0/0.43
```

## Task 4

After configuring the previous task, it is obvious that there is no redundancy. If the CE routers (R3 and R4) loose their directly connected link, they will have no reachability to each other's routes. Configure the appropriate router/s based on the following policy:

If the link between R3 and R4 is up, they should use each other as the next hop to reach the routes that they are advertising.

If the link between R3 and R4 is down, they should go through the cloud to reach each other's routes.

You should configure and test two different solutions to accomplish this task:

**Before we begin to test the two methods, let's change the extended community value configured on R2 back to 2:222:**

### On R2

```
R2(config)#Route-map SOO permit 10
R2(config-route-map)#No set extcommunity soo 1:121
R2(config-route-map)#set extcommunity soo 2:222
```

### To test the configuration:

### On R3

```
R3#Show ip route Eigrp

     4.0.0.0/24 is subnetted, 1 subnets
D       4.4.4.0 [90/2297856] via 34.1.1.4, 00:00:15, Serial0/0.34
     24.0.0.0/24 is subnetted, 1 subnets
D       24.1.1.0 [90/2681856] via 34.1.1.4, 00:00:15, Serial0/0.34
                 [90/2681856] via 13.1.1.1, 00:00:15, Serial0/0.31
```

**One way to accomplish this task is to manipulate the metric in Eigrp, to see the delay value before changing:**

```
R3#Show int s0/0.31 | Inc DLY

  MTU 1500 bytes, BW 1544 Kbit/sec, DLY 20000 usec,
```

**Let's change the delay value:**

### On R3

```
R3(config-if)#Int S0/0.31
R3(config-subif)#delay 4000

R3#Clear ip eigrp neighbor
```

**To test and verify the configuration**

**On R3**

```
R3#Show int s0/0.31 | Inc DLY

  MTU 1500 bytes, BW 1544 Kbit/sec, DLY 40000 usec,

R3#Show ip route Eigrp

     4.0.0.0/24 is subnetted, 1 subnets
D       4.4.4.0 [90/2297856] via 34.1.1.4, 00:00:51, Serial0/0.34
     24.0.0.0/24 is subnetted, 1 subnets
D       24.1.1.0 [90/2681856] via 34.1.1.4, 00:00:51, Serial0/0.34
```

**This forced the traffic through directly connected neighbor R4. To test redundancy, the S0/0.34 sub-interface is shutdown and the routing table of the CE routers is checked:**

**On R3**

```
R3(config)#Int S0/0.34
R3(config-subif)#Shut
```

**On R4**

```
R4(config)#Int S0/0.43
R4(config-subif)#shut
```

**On R3**

```
R3#Show ip route Eigrp

     4.0.0.0/24 is subnetted, 1 subnets
D       4.4.4.0 [90/3321856] via 13.1.1.1, 00:01:56, Serial0/0.31
     24.0.0.0/24 is subnetted, 1 subnets
D       24.1.1.0 [90/3193856] via 13.1.1.1, 00:01:57, Serial0/0.31

R3#Ping 4.4.4.4

Type escape sequence to abort.
```

```
Sending 5, 100-byte ICMP Echos to 4.4.4.4, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 160/160/164 ms

R3#Ping 24.1.1.4

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 24.1.1.4, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 160/164/176 ms
```

**Let's remove the metric and test redundancy using the second method:**

## On R4

```
R4(config)#Int S0/0.43
R4(config-subif)#No shut
```

## On R3

```
R3(config)#Int S0/0.34
R3(config-subif)#No shut

R3(config)#Int S0/0.31
R3(config-subif)#delay 2000

R3#Clear ip eigrp neighbor

R3#Show ip route Eigrp

     4.0.0.0/24 is subnetted, 1 subnets
D       4.4.4.0 [90/2297856] via 34.1.1.4, 00:00:15, Serial0/0.34
     24.0.0.0/24 is subnetted, 1 subnets
D       24.1.1.0 [90/2681856] via 34.1.1.4, 00:00:15, Serial0/0.34
                 [90/2681856] via 13.1.1.1, 00:00:15, Serial0/0.31
```

**In this scenario EEM is used, EEM is configured such that as long as 34.1.1.4 is up, it will apply the filter, meaning that it will set the extended community value to match the one that is configured on R1.**

**But if the link is down, meaning that the IP SLA feature can no longer send and receive ICMP ECHOes, it will change the extended community value to 2:222, and sicne this value doesn't match the one configured on R1, it will NOT filter the route and the routes advertised by R3 will be available through the MPLS cloud.**

## Let's configure this method:

## On R2

```
R2(config)#ip sla 1
R2(config-ip-sla)#icmp-echo 34.1.1.4 source-interface Serial0/0.24
R2(config-ip-sla-echo)#timeout 500
R2(config-ip-sla-echo)#frequency 5
R2(config-ip-sla-echo)#vrf TST

R2(config)#ip sla schedule 1 life forever start-time now

R2(config)#track 2 rtr 1 reachability

R2(config)#event manager applet tst
R2(config-applet)#event track 2 state down
R2(config-applet)#action 1.1 cli command "enable"
R2(config-applet)#action 1.2 cli command "config t"
R2(config-applet)#action 1.3 cli command "route-map SOO permit 10"
R2(config-applet)#action 1.4 cli command "NO set extcommunity soo 1:121"
R2(config-applet)#action 1.5 cli command "set extcommunity soo 2:222"

R2(config)#event manager applet tst1
R2(config-applet)#event track 2 state up
R2(config-applet)#action 1.1 cli command "en"
R2(config-applet)#action 1.2 cli command "config t"
R2(config-applet)#action 1.3 cli command "route-map SOO permit 10"
R2(config-applet)#action 1.4 cli command "NO set extcommunity soo 2:222"
R2(config-applet)#action 1.5 cli command "set extcommunity soo 1:121"
```

## To test the configuration:

**NOTE: The S0/0.43 sub-interface is up, and therefore, the operation of IP SLA is successful, therefore, R3 and R4 should reach each other's networks using each other as the next hop:**

## On R3

```
R3#Show ip route Eigrp

     4.0.0.0/24 is subnetted, 1 subnets
D       4.4.4.0 [90/2297856] via 34.1.1.4, 00:01:38, Serial0/0.34
     24.0.0.0/24 is subnetted, 1 subnets
D       24.1.1.0 [90/2681856] via 34.1.1.4, 00:00:54, Serial0/0.34
```

## On R4

```
R4#Show ip route Eigrp
```

```
     3.0.0.0/24 is subnetted, 1 subnets
D       3.3.3.0 [90/2297856] via 34.1.1.3, 00:01:59, Serial0/0.43
     13.0.0.0/24 is subnetted, 1 subnets
D       13.1.1.0 [90/2681856] via 34.1.1.3, 00:01:47, Serial0/0.43
```

**NOTE: The tag matches the one configured on R1:**

## On R1

```
R1#Show ip eigrp vrf TST topology 24.1.1.0/24 | s 2.2.2.2

  2.2.2.2, from VPNv4 Sourced, Send flag is 0x0
      Composite metric is (2169856/0), Route is Internal (VPNv4 Sourced)
      Vector metric:
        Minimum bandwidth is 1544 Kbit
        Total delay is 20000 microseconds
        Reliability is 255/255
        Load is 1/255
        Minimum MTU is 1500
        Hop count is 0
      Extended Community: SoO:1:121
```

**Let's shutdown the S0/0.43 sub-interface of R4 and verify the result:**

## On R4

```
R4(config)#Int S0/0.43
R4(config-subif)#Shut
```

**You should see the following console messages on R2:**

```
%TRACKING-5-STATE: 2 rtr 1 reachability Down->Up
%TRACKING-5-STATE: 2 rtr 1 reachability Up->Down
```

## On R1

```
R1#Show ip eigrp vrf TST topology 24.1.1.0/24 | s 2.2.2.2

  2.2.2.2, from VPNv4 Sourced, Send flag is 0x0
      Composite metric is (2169856/0), Route is Internal (VPNv4 Sourced)
      Vector metric:
        Minimum bandwidth is 1544 Kbit
        Total delay is 20000 microseconds
        Reliability is 255/255
        Load is 1/255
        Minimum MTU is 1500
        Hop count is 0
      Extended Community: SoO:2:222
```

**On R3**

```
R3#Show ip route Eigrp

     4.0.0.0/24 is subnetted, 1 subnets
D       4.4.4.0 [90/2809856] via 13.1.1.1, 00:01:17, Serial0/0.31
     24.0.0.0/24 is subnetted, 1 subnets
D       24.1.1.0 [90/2681856] via 13.1.1.1, 00:01:17, Serial0/0.31

R3#Ping 4.4.4.4

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 4.4.4.4, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 160/160/160 ms
```

**On R4**

```
R4#Show ip route Eigrp

     34.0.0.0/24 is subnetted, 1 subnets
D       34.1.1.0 [90/3193856] via 24.1.1.2, 00:02:03, Serial0/0.42
     3.0.0.0/24 is subnetted, 1 subnets
D       3.3.3.0 [90/2809856] via 24.1.1.2, 00:02:03, Serial0/0.42
     13.0.0.0/24 is subnetted, 1 subnets
D       13.1.1.0 [90/2681856] via 24.1.1.2, 00:02:03, Serial0/0.42

R4#Ping 3.3.3.3

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 3.3.3.3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 160/160/164 ms
```

**Let's No shut the S0/0.43 sub-interface and see the result:**

**On R4**

```
R4(config)#Int s0/0.43
R4(config-subif)#No shut
```

**On R1**

```
R1#Show ip eigrp vrf TST topology 24.1.1.0/24 | s 2.2.2.2

  2.2.2.2, from VPNv4 Sourced, Send flag is 0x0
      Composite metric is (2169856/0), Route is Internal (VPNv4 Sourced)
      Vector metric:
        Minimum bandwidth is 1544 Kbit
        Total delay is 20000 microseconds
        Reliability is 255/255
        Load is 1/255
        Minimum MTU is 1500
        Hop count is 0
      Extended Community: SoO:1:121
```

**On R3**

```
R3#Show ip route Eigrp

     4.0.0.0/24 is subnetted, 1 subnets
D       4.4.4.0 [90/2297856] via 34.1.1.4, 00:01:00, Serial0/0.34
     24.0.0.0/24 is subnetted, 1 subnets
D       24.1.1.0 [90/2681856] via 34.1.1.4, 00:00:27, Serial0/0.34
```

**You can see that EEM can be very useful, in this case you could have configured another IP SLA and an EEM on R3 that looked at the results of the IP SLA that constantly sent ICMP-ECHO to R4's IP address and shutdown its interface (S0/0.34) if 34.1.1.4 was NOT up.**

## Task 5

**Stop all routers in the console window and exit. Stop the Server and press the "Erase Start" launcher before proceeding.**

# Advanced
# CCIE SERVICE PROVIDER
# v3.0

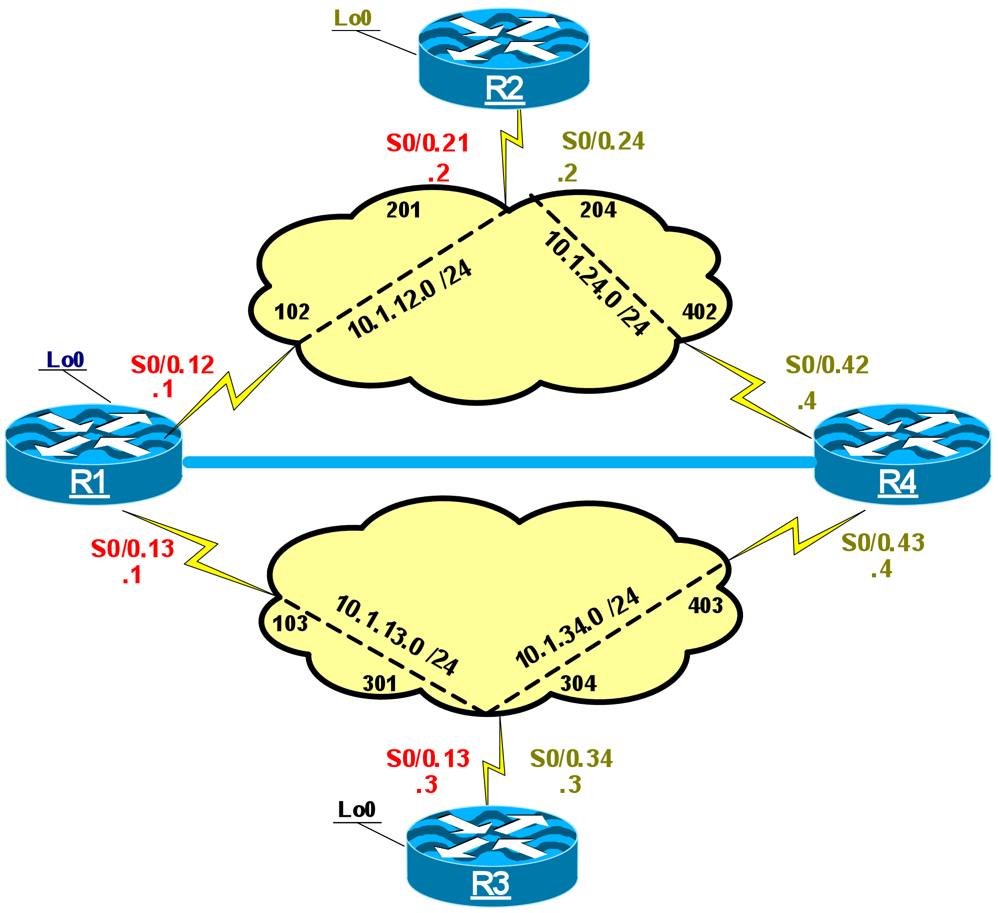
**www.MicronicsTraining.com**

**Narbik Kocharians**
**CCIE #12410**
**R&S, Security, SP**

**Paul Negron**
**CCIE #14856**
**SP**

## Traffic Engineering

# Lab 1 – Configuring Static Tunnels



## IP Addressing:

| Router | Interface | IP address |
|---|---|---|
| R1 | Lo0<br>S0/0.12<br>S0/0.13<br>F0/0 | 1.1.1.1 /32<br>10.1.12.1 /24<br>10.1.13.1 /24<br>10.1.14.1/ 24 |
| R2 | Lo0<br>S0/0.21<br>S0/0.23 | 2.2.2.2 /32<br>10.1.12.2 /24<br>10.1.24.2 /24 |
| R3 | Lo0<br>S0/0.31<br>S0/0.34 | 3.3.3.3/ 32<br>10.1.13.3 /24<br>10.1.34.3 /24 |
| R4 | Lo0<br>F0/0<br>S0/0.42<br>S0/0.43 | 4.4.4.4 /32<br>10.1.14.4/ 24<br>10.1.24.4 /24<br>10.1.34.4 /24 |

## Lab Setup:

The connections between R1 to R2, R1 to R3, R2 to R4, and R3 to R4 are to be setup in a Frame-Relay Point-to-Point manner

## Task 1

Configure an OSPF routed core using process "1". Support mpls traffic engineering throughout the core topology.

### Configure basic traffic Engineering on ALL Core Devices:

### On R1

```
R1(config)#mpls traffic-eng tunnels
```
**(This command allows MPLS-TE globally)**

**Configure on all core routers R1, R2, R3 and R4 with this command.**

```
R1(config)#router ospf 1
R1(config)#network 0.0.0.0 0.0.0.0 area 0
R1(config-router)#mpls traffic-eng area 0
R1(config-router)#mpls traffic-eng router-id loopback 0
```

**OSPF needs to recognize and use the OSPF Opaque LSA (type 10) to include the Traffic Engineering paths that will be configured. This command must also be configured on ALL core routers.**

```
R1(config)#interface s0/0.12
R1(config-if)#mpls traffic-eng tunnels
R1(config-if)#ip rsvp bandwidth 60

R1(config)#interface f0/0
R1(config-if)#mpls traffic-eng tunnels
R1(config-if)#ip rsvp bandwidth 60

R1(config-if)#interface s0/0.13
R1(config-if)#mpls traffic-eng tunnels
R1(config-if)#ip rsvp bandwidth (we leave this option with the default)
```

**ALL interfaces that may carry tunnel traffic at any time must be configured with the "mpls traffic-eng tunnels" command on the interface as well. If this command is not issued at this level, the router will not participate in MPLS Traffic Engineering path calculation (PCALC).**

**ALL interfaces that may carry tunnel traffic at any time must be configured with the "ip rsvp bandwidth" command on the interface that could be a potential path. The router will not participate in MPLS Traffic Engineering path calculation (PCALC) from the head-end of the tunnel if not configured. The default value, if not specified, will be 75% of the "bandwidth" command configured on the interface. 1.544 Mbps is the default bandwith of any of the serial interfaces. RSVP will use this value to subtract any requested bandwidth by a tunnel.**

**The three configurations: "mpls traffic-eng tunnels"(global), "mpls traffic-eng tunnels"(interface), and "ip rsvp bandwith" must all be used on all core interfaces to participate in MPLS Traffic Engineering, even if there will be no tunnels built from the router itself.**

## On R2

```
R2(config)#mpls traffic-eng tunnels

R2(config)#router ospf 1
R2(config)#network 0.0.0.0 0.0.0.0 area 0
R2(config-router)#mpls traffic-eng area 0
R2(config-router)#mpls traffic-eng router-id loopback 0

R2(config)#interface s0/0.21
R2(config-if)#mpls traffic-eng tunnels
R2(config-if)#ip rsvp bandwidth

R2(config-if)#interface s0/0.24
R2(config-if)#mpls traffic-eng tunnels
R2(config-if)#ip rsvp bandwidth
```

## On R3

```
R3(config)#mpls traffic-eng tunnels

R3(config)#router ospf 1
R3(config)#network 0.0.0.0 0.0.0.0 area 0
R3(config-router)#mpls traffic-eng area 0
R3(config-router)#mpls traffic-eng router-id loopback 0

R3(config)# interface s0/0.31
R3(config-if)#mpls traffic-eng tunnels
R3(config-if)#ip rsvp bandwidth

R3(config-if)#interface s0/0.34
R3(config-if)#mpls traffic-eng tunnels
R3(config-if)#ip rsvp bandwidth
```

**On R4**

```
R4(config)#mpls traffic-eng tunnels

R4(config)#router ospf 1
R4(config)#network 0.0.0.0 0.0.0.0 area 0
R4(config-router)#mpls traffic-eng area 0
R4(config-router)#mpls traffic-eng router-id loopback 0

R4(config)#interface s0/0.42
R4(config-if)#mpls traffic-eng tunnels
R4(config-if)#ip rsvp bandwidth

R4(config)#interface f0/0
R4(config-if)#mpls traffic-eng tunnels
R4(config-if)#ip rsvp bandwidth

R4(config-if)#interface s0/0.43
R4(config-if)#mpls traffic-eng tunnels
R4(config-if)#ip rsvp bandwidth
```

## Verifying basic traffic Engineering:

**On R1**

```
R1#Sh mpls traffic-eng link-management igp-neighbors

Link ID::  Fa0/0
    Neighbor ID:  4.4.4.4 (area: ospf area 0, IP: 10.1.14.4)
Link ID::  Se0/0.12
```

```
    Neighbor ID:  2.2.2.2 (area: ospf area 0, IP: 10.1.12.2)
Link ID::  Se0/0.13
    Neighbor ID:  3.3.3.3 (area: ospf area 0, IP: 10.1.13.3)
```

**MPLS traffic engineering has been correctly configured under the core interfaces or we would not be able to see neighbors.**

```
R1#Sh ip ospf database | be Type-10

                Type-10 Opaque Link Area Link States (Area 0)

Link ID         ADV Router      Age         Seq#            Checksum    Opaque ID
1.0.0.0           1.1.1.1       183         0x80000002      0x004347     0
1.0.0.0           2.2.2.2       148         0x80000002      0x00CCB9     0
1.0.0.0           3.3.3.3       112         0x80000002      0x004932     0
1.0.0.0           4.4.4.4       97          0x80000002      0x00CCE4     0
1.0.0.1           1.1.1.1       31          0x80000001      0x0038B6     1
1.0.0.1           2.2.2.2       31          0x80000001      0x008945     1
1.0.0.1           3.3.3.3       108         0x80000002      0x001C3E     1
1.0.0.1           4.4.4.4       93          0x80000002      0x00A50F     1
1.0.0.2           1.1.1.1       23          0x80000001      0x00AE77     2
1.0.0.2           4.4.4.4       75          0x80000001      0x00ACD5     2
```

**The type 10 opaque LSA is used to identify all of the LSA's learned from the MPLS traffic-engineering topology. The highlighted paths all share the same link ID's which represents the tunnel path and all of the LSA's learned from that path.**

```
R1#Sh ip rsvp int

interface     allocated  i/f max  flow max        sub max
F0/0          0          60K      60K             0
Se0/0         0          0        0               0
Se0/0.12      0          60K      60K             0
Se0/0.13      0          1158K    1158K           0
```

**RSVP has been configured under each of the core interfaces as well. The default-reserved bandwidth is 75% of the physical bandwidth.  Default would be Se0/0.13 = 1544000 – 386000 (25%) = 1158K. Since we set the bandwidth to restrict to 60K on the other interfaces, it overrides the default.**

**Configure R1 s0/0.13 with "ip rsvp bandwidth 60" command before proceeding!!**

## Task 2

Configure MPLS-TE to establish basic connectivity from R1 to R4. R1 should not follow the default routing path. The path through R2 should be favored as a static path with the following attributes. Priority = 4, all traffic uses the tunnel to 4.4.4.4 and bandwidth = 30K.

## Verify Current Routing Path:

### On R1

```
R1#Trace 4.4.4.4

Type escape sequence to abort.
Tracing the route to 4.4.4.4

  1 10.1.14.4 4 msec * 4 msec

R1#Sh ip cef exact-route 1.1.1.1 4.4.4.4

1.1.1.1         -> 4.4.4.4         : FastEthernet0/0 (next hop 10.1.14.4)
```

**The current path is currently R1-R4 through the F0/0 interface.**

## Change the Current Routing Path:

```
R1(config)#interface tunnel 0
R1(config-if)#ip unnumbered loopback 0
```

**A good practice to not waste an ip address on the tunnel interface.**

```
R1(config-if)#tunnel destination 4.4.4.4
```

**The destination of the tunnel = the traffic-eng router-id of the last hop router**

```
R1(config-if)#tunnel mode mpls traffic-eng
```

**The tunnel mode must be changed from the default point-to-point tunnel mode, to mpls traffic engineering to hav RSVP create labels.**

```
R1(config-if)#tunnel mpls traffic-eng autoroute announce
```

**This command ensures that all traffic is routed into the tunnel and announced into ospf updates.**

```
R1(config-if)#tunnel mpls traffic-eng  priority 4 4
```

**The tunnel priority will use the SETUP priority (the first number) and compare with another tunnel's HOLD priority ( the second number in the command). Both numbers are usually set to be the same to avoid confusion but may be different to establish true priority.**

```
R1(config-if)#tunnel mpls traffic-eng bandwidth 30
R1(config-if)#tunnel mpls traffic-eng path-option 1 explicit name SLOW_LINK
R1(config-if)#exit
```

**The tunnel bandwidth of "30" is in kbps and is what the tunnel requires to be setup. The tunnel should still remain down until the explicit path is created.**

```
R1(config)#ip explicit-path name SLOW_LINK enable
R1(config)#next-address 10.1.12.2
R1(config)#next-address 10.1.24.4
```

**The path should come up after the last hop is configured**

## Verify the Traffic Engineered path Information

### On R1

```
R1#Sh mpls traffic-eng tunnels brief

Signalling Summary:
    LSP Tunnels Process:            running
    RSVP Process:                   running
    Forwarding:                     enabled
    Periodic reoptimization:        every 3600 seconds, next in 2149
seconds
    Periodic auto-bw collection:    disabled

TUNNEL NAME                      DESTINATION      UP IF     DOWN IF   STATE/PROT
R1_t0                            4.4.4.4          -         Se0/0.12  up/up
Displayed 1 (of 1) heads, 0 (of 0) midpoints, 0 (of 0) tails
```

**Notice that this is the "head" of the tunnel. It does not represent any other tunnel as a midpoint or a tail. The LSP Tunnel and RSVP processes show a running state.**

**Forwarding should be enabled or the top two processes do not matter.**

**Periodic reoptimazation allows for the tunnels to recalculate the best paths automatically over a specific amount of time (3600 seconds). There will be another update in 2149 seconds.**

**Periodic auto-bw collection is currently disabled and is used when automatically attempting to reroute based on utilization. This will be covered in another section.**

**The tunnel name, "R1_t0", can sometimes be referenced in the command line. The destination of the tunnel is the TE Router-ID of the last hop TE router R4.**

**The "UP IF" is a term used to describe the Downstream peer (next hop router) from a Data Plane perspective. The "DOWN IF" is "Serial 0/0.12". The router output locks the field in this output to a certain size. In some cases, the interface name is too long for the space. The interface can be seen with the "show mpls traffic-eng tunnels" command.**

**The State and Protocol are treated as layer 1 and Layer 2 for the physical interface. Possible states are "*up/up*"- the physical line and the line protocol are operational.**

***up/down*- the physical layer is good for the tunnel but the line protocol has been broken as in the example of a sub-interface or physical interface failure. The tunnel interface is fine from a logical and physical layer view.**

```
R1#Sh mpls traffic-eng tunnels

Name: R1_t0                          (Tunnel0) Destination: 4.4.4.4
  Status:
    Admin: up        Oper: up     Path: valid       Signalling: connected
    path option 1, type explicit SLOW_LINK (Basis for Setup, path weight 128)

  Config Parameters:
    Bandwidth: 30       kbps (Global)  Priority: 4  4   Affinity: 0x0/0xFFFF
    Metric Type: TE (default)
    AutoRoute:  enabled   LockDown: disabled  Loadshare: 30       bw-based
    auto-bw: disabled

  InLabel  :  -
  OutLabel : Serial0/0.12, 16
  RSVP Signalling Info:
       Src 1.1.1.1, Dst 4.4.4.4, Tun_Id 0, Tun_Instance 4
    RSVP Path Info:
      My Address: 1.1.1.1
      Explicit Route: 10.1.12.2 10.1.24.4 4.4.4.4
      Record Route:  NONE
      Tspec: ave rate=30 kbits, burst=1000 bytes, peak rate=30 kbits
    RSVP Resv Info:
      Record Route:  NONE
      Fspec: ave rate=30 kbits, burst=1000 bytes, peak rate=30 kbits
  History:
    Tunnel:

      Time since created: 2 hours, 17 minutes
      Time since path change: 2 hours, 15 minutes
    Current LSP:
      Uptime: 2 hours, 15 minutes
```

**The name, destination have already been covered.**

**The "*Admin: up*", "*Oper: up*, *Path: Valid* and *Signalling*: *connected* fields, make up extremly useful ouput for troubleshooting purposes. By the end of the labs, you should be able to use the "Path valid" and "Signalling" output to its' fullest extent.**

**The "path option 1" describes the number of the path-option declared by configuration that the tunnel has chosen. "type Explicit SLOW_LINK" was the explicit map configured earlier. The "(basis for Setup, setup weight 128)" field is igp/te cost = 64 + 64 = 128. The Router uses the TE metric by default for path selection process and the te metric matches the igp metric by default.**

**The section Config Parameters section covers "Bandwidth: 30". This reflects the bandwidth required by the tunnel to be established in kbps and it is taken from the (Global) pool of bandwidth from the interface.**

**The "setup priority" is listed here for quick reference.**

**Tunnel Affinity is covered later but is noted here as a 32 bit value if needed.**

**"Metric Type is "TE" by default as covered in the explanation of the basis for setup.**

**"Autoroute- enable" was the chosen method to route traffic into the tunnel. The "Lockdown-disabled" feature does not allow the tunnel to move from the current path and is disabled by default.**

**The "Load-share- disabled" can be used instead of specifying the bandwidth manually. It will load share the traffic between tunnels if more than one tunnel exist and has this feature enabled as well. There is further explanation of (bw-based) protocols later in the lesson. "auto-bw- disabled" field can be used in conjunction with manual bandwidth statements but is disabled by default.**

**The "Inlabel" displays the label as it applies to the data plane. A label that arrives from an peer is swapped with the "OutLabel" that may be imposed initially, as in this case. The "Outlabel" displays the label that is forwarded toward the route origination point and specifies the interface (Serial 0/0.12) and the label (16) that is used to accomplish this task.**

**The "RSVP signalling Info:" displays the SouRCe and DeSTination of the tunnel RSVP configured tunnel. The tunnel ID is "0" and the "Tun_Instance" will change anytime the tunnel is torn down.**

**The "RSVP Path Info:" list the address of the router and the "Explicit Route" of the hop-by-hop path from R1 to R4 through R2. "Record Route" will be covered later.**
**The Tspec , or Traffic specification is used to enforce policing on the traffic using the tunnel to the**

**Tspec value specified in kbps. The Fspec or Flow specification is the value that will be used in RESerVation (RESV) messages by a receiver node to request QOS parameters.**

**The History of the tunnel can prove very useful in troubleshooting to establish patterns of success or**

**failure. The "Tunnel:" is the time since the tunnel was created and how long since the last path change.**

**The "Current LSP:" output lists how long the current tunnel has been operational.**

**The "Prior LSP:" shows the path option in question and the "Removal Trigger" list what the potential problem or cause of the last LSP change or failure. The prior instance" is listed between the brackets after the path option.**

```
R1#Sh ip rsvp int

interface     allocated  i/f max  flow max    sub max
F0/0          0          60K      60K         0
Se0/0.12      30K        60K      60K         0
Se0/0         0          0        0           0
Se0/0.13      0          60K      60K         0
```

**The interface through R2 is reserving the bandwidth as 30K.**

## Control and Data Plane:

### On R1

```
R1#Sh mpls ldp parameters

Protocol version: 1
No label generic region for downstream label distribution
Session hold time: 180 sec; keep alive interval: 60 sec
Discovery hello: holdtime: 15 sec; interval: 5 sec
Discovery targeted hello: holdtime: 90 sec; interval: 10 sec
Downstream on Demand max hop count: 255
Downstream on Demand Path Vector Limit: 255
LDP for targeted sessions
LDP initial/maximum backoff: 15/120 sec
LDP loop detection: off
```

**R1 is performing "Downstream on Demand" learning of labels. This means that R1 will need to receive a label concerning a Downstream Network first before allocating a label for a given destination.**

### On R4

```
R4#Sh mpls traffic-eng tunnels int  | b InLabel |Out

  InLabel  : Serial0/0.42, implicit-null
  OutLabel : -
```

**R1 sends an RSVP "PATH message" that request for a label concerning the tunnel.**
**The request passes through R2 and eventually reaches R4. Once the request is received for the tunnel, it is assigned a label of "implicit null" and will send this label to R2. The "OutLabel" represents the action that will be taken in the event it receives the "InLabel" context. In this case, this is the termination point so there should be no Outlabel Context.**

## On R2

```
R2#Sh mpls traffic-eng tunnels int  | b InLabel |Out

  InLabel  : Serial0/0.21, 16
  OutLabel : Serial0/0.24, implicit-null

R2#Sh mpls traffic-eng tun brie | b TUNNEL

TUNNEL NAME            DESTINATION     UP IF      DOWN IF   STATE/PROT
R1_t0                  4.4.4.4         Se0/0.21   Se0/0.24    up/up
Displayed 0 (of 0) heads, 1 (of 1) midpoints, 0 (of 0) tails

R2#Sh mpls forwarding-table | b Local

Local  Outgoing    Prefix          Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id    switched   interface
16     Pop tag     1.1.1.1 0 [4]     0        Se0/0.24   point2point
```

**Control Plane: R2 assignes a local tag of "16" for the tunnel when it receives the implicit null tag from R4. The tag could be different in your output.**

**Data Plane: R2 is waiting for label "16" on the InLabel interface s0/0.21 and forwarding the route with the implicit null tag (label "3") out of "OutLabel" int s0/0.24.**

## On R1

```
R1#Sh mpls traffic-eng tunnels int  | b InLabel |Out

 InLabel  : -
 OutLabel : Serial0/0.12, 16
```

**Control Plane: R1 learns the "16 label from R2.**
**Data Plane: Label "16" will be imposed if R1 receives a packet destined to use the tunnel.**

```
R1#Trace 4.4.4.4

Type escape sequence to abort.
Tracing the route to 4.4.4.4
```

```
  1 10.1.12.2 [MPLS: Label 16 Exp 0] 48 msec 28 msec 4 msec
  2 10.1.24.4 4 msec *  4 msec

R1#Sh ip cef exact-route 1.1.1.1 4.4.4.4

1.1.1.1         -> 4.4.4.4        : Tunnel0 (next hop 0.0.0.0)
```

**R1 uses the slower path due to the TE configuration on the R1 tunnel head.**

## Task 3

Configure a second tunnel (tunnel ”1”) on R1 that uses the same path option configured in Task 2. Provide an additional path option 2 MEDIUM_LINK on both tunnels. This path option should use the path through R3.

### On R1

```
R1(config)#interface Tunnel0
R1(config-if)#tunnel mpls traffic-eng path-option 2 explicit name MEDIUM_LINK
```

**We add the second path option on tunnel 0**

```
R1(config)#interface Tunnel1
R1(config-if)#ip unnumbered Loopback0
R1(config-if)#tunnel destination 4.4.4.4
R1(config-if)#tunnel mode mpls traffic-eng
R1(config-if)#tunnel mpls traffic-eng autoroute announce
R1(config-if)#tunnel mpls traffic-eng priority 4 4
R1(config-if)#tunnel mpls traffic-eng bandwidth  30
R1(config-if)#tunnel mpls traffic-eng path-option 1 explicit name SLOW_LINK
R1(config-if)#tunnel mpls traffic-eng path-option 2 explicit name MEDIUM_LINK
```

**The second tunnel is created with similar options for simplicity. The path will not come up until at least one of the explicit paths is configured.**

```
R1(config)#ip explicit-path name MEDIUM_LINK
R1(cfg-ip-expl-path)#next-address 10.1.13.3

Explicit Path name MEDIUM_LINK:

    1: next-address 10.1.13.3
R1(cfg-ip-expl-path)#next-address 10.1.34.4
```

```
Explicit Path name MEDIUM_LINK:
    1: next-address 10.1.13.3
    2: next-address 10.1.34.4

R1(cfg-ip-expl-path)#end
```

**The tunnel comes up immediately. 2 paths have been created. We need to see what tunnel is being used and analyze the constraints.**

```
R1#Sh mpls traffic-eng tunnels name R1_t0

Name: R1_t0                        (Tunnel0) Destination: 4.4.4.4
  Status:
    Admin: up       Oper: up     Path: valid       Signalling: connected

    path option 1, type explicit SLOW_LINK (Basis for Setup, path weight 128)
    path option 2, type explicit MEDIUM_LINK

  Config Parameters:
    Bandwidth: 30       kbps (Global)  Priority: 4  4   Affinity: 0x0/0xFFFF
    Metric Type: TE (default)
    AutoRoute:  enabled   LockDown: disabled  Loadshare: 30      bw-based
    auto-bw: disabled

  InLabel  :  -
  OutLabel : Serial0/0.12, 16
  RSVP Signalling Info:
       Src 1.1.1.1, Dst 4.4.4.4, Tun_Id 0, Tun_Instance 26
    RSVP Path Info:
      My Address: 1.1.1.1
      Explicit Route: 10.1.12.2 10.1.24.4 4.4.4.4

---text omitted---
```

**The "Tunnel 0" interface uses the 1st path option since the tunnel priority does not come into play in this scenario and the bandwidth is available.**

```
R1#Sh mpls traffic-eng tunnels name-regexp ^R1_t1$
```

**Yes….We can use regular expressions to match on names!!**

```
Name: R1_t1                        (Tunnel1) Destination: 4.4.4.4

  Status:

    Admin: up       Oper: up     Path: valid       Signalling: connected
```

```
   path option 1, type explicit SLOW_LINK (Basis for Setup, path weight 128)
   path option 2, type explicit MEDIUM_LINK

  Config Parameters:
   Bandwidth: 30       kbps (Global)  Priority: 4  4   Affinity: 0x0/0xFFFF
   Metric Type: TE (default)
   AutoRoute:  enabled   LockDown: disabled  Loadshare: 30       bw-based
   auto-bw: disabled

  InLabel  :  -
  OutLabel : Serial0/0.12, 17
  RSVP Signalling Info:
      Src 1.1.1.1, Dst 4.4.4.4, Tun_Id 1, Tun_Instance 1
   RSVP Path Info:
     My Address: 1.1.1.1
     Explicit Route: 10.1.12.2 10.1.24.4 4.4.4.4

---text omitted---
```

**"Tunnel 1" also uses the SLOW_LINK. There is enough bandwidth available and it is also the first option on this tunnel as well.**

## Task 4

Configure Tunnel "1 "to use 40 K of bandwidth and reoptomize the tunnels. Since RSVP bandwidth has been limited to 60K on every interface, view the changes that result in reoptimization of the tunnels.

**On R1**

```
R1(config)#interface tunnel 1
R1(config-if)#tunnel mpls traffic-eng bandwidth 40
```

**We will view the effects of changing the bandwidth requirement of "Tunnel 1"**

```
R1#Sh mpls traffic-eng tunnels brie

Signalling Summary:
    LSP Tunnels Process:           running
    RSVP Process:                  running
    Forwarding:                    enabled
    Periodic reoptimization:       every 3600 seconds, next in 1585 seconds
```

```
  Periodic auto-bw collection:    disabled
TUNNEL NAME          DESTINATION       UP IF       DOWN IF    STATE/PROT
R1_t0                4.4.4.4            -          Se0/0.12   up/up
R1_t1                4.4.4.4            -          Se0/0.13   up/up
Displayed 2 (of 2) heads, 0 (of 0) midpoints, 0 (of 0) tails
```

**The traffic that is currently using the SLOW_LINK may not change until reoptimization takes place (3600 seconds). In ths case, a changing of the resources has taken place, which triggers TE path change due to the constraint. Disabling timer frequency or events will have no effect.**

```
R1#Sh mpls traffic-eng tunnel role head source-id 1.1.1.1 name R1_t1

Name: R1_t1                        (Tunnel1) Destination: 4.4.4.4
  Status:
    Admin: up        Oper: up     Path: valid      Signalling: connected

    path option 2, type explicit MEDIUM_LINK (Basis for Setup, path weight 128)
    path option 1, type explicit SLOW_LINK

  Config Parameters:
    Bandwidth: 40       kbps (Global)  Priority: 4  4   Affinity: 0x0/0xFFFF
    Metric Type: TE (default)
    AutoRoute:  enabled   LockDown: disabled  Loadshare: 40       bw-based
    auto-bw: disabled

  InLabel  :  -
  OutLabel : Serial0/0.13, 16
  RSVP Signalling Info:
       Src 1.1.1.1, Dst 4.4.4.4, Tun_Id 1, Tun_Instance 3
    RSVP Path Info:
      My Address: 1.1.1.1
      Explicit Route: 10.1.13.3 10.1.34.4 4.4.4.4
      Record Route:  NONE
      Tspec: ave rate=40 kbits, burst=1000 bytes, peak rate=40 kbits
    RSVP Resv Info:
      Record Route:  NONE
      Fspec: ave rate=40 kbits, burst=1000 bytes, peak rate=40 kbits
  History:
    Tunnel:
      Time since created: 5 minutes, 30 seconds
      Time since path change: 2 minutes, 6 seconds
    Current LSP:
      Uptime: 2 minutes, 6 seconds

      Selection: reoptimation
```

```
Prior LSP:
  ID: path option 1 [1]
  Removal Trigger: configuration changed
  Last Error: PCALC:: Can't use link 0.0.0.0 on node 1.1.1.1
```

**Notice how although the SLOW_LINK is the 1st option, the MEDIUM_PATH is chosen because the primary link only has a bandwidth of 60 kbps configured on the interface. The "Tunnel 0" interface has a requirement of 30 kbps. The Tunnel 1" interface has a requirement of 40 kbps and RSVP will reject the reservation on the primary path. The second path is unused and is chosen as the best for this tunnel.**
**Also, the Current LSP information displays that the new LSP was due to reoptimization. The reoptimization is now automatic when a change is performed on the tunnel and does not necessarily need to be manually triggered. In Cisco IOS, The lowered number tunnel will win if ALL constraints are tied except for bandwidth.**

## Task 5

Configure the tunnels such that when shutdown manually, Tunnel"1" will use the primary path no matter which order the tunnels come up. Even during reoptimization.

**On R1**

```
R1(config)#int tun 0
R1(config-if)#shut
R1(config-if)#int tun 1
R1(config-if)#shut

R1(config)#int tunnel 1
R1(config-if)#tunnel mpls traffic-eng priority 3 4
% Setup priority (3) may not be higher than hold priority (4)

R1(config-if)#tunnel mpls traffic-eng priority 3 3
```

**The priority is used to accomplish the task correctly. The hold priority cannot be a higher number (inferior) than the setup priority. They are normally set to be identical but the hold priority can be set lower (better) than the setup priority. In some cases the traffic using the tunnel may not need a better priority to be established but once it is established it should not be taken down.**

```
R1(config)#int tun 0
R1(config-if)#NO shut

*Mar  1 00:48:26.939: %LINK-3-UPDOWN: Interface Tunnel0, changed state to up
```

```
*Mar  1 00:48:27.939: %LINEPROTO-5-UPDOWN: Line protocol on Interface Tunnel0, changed state to up

R1#Sh mpls traffic-eng tunnel brie | b TUNNEL

TUNNEL NAME              DESTINATION      UP IF      DOWN IF    STATE/PROT
R1_t0                    4.4.4.4          -          Se0/0.12   up/up
R1_t1                    4.4.4.4          -          unknown    admin-down
Displayed 2 (of 2) heads, 0 (of 0) midpoints, 0 (of 0) tails
```

**We bring up tunnel "0" first and make sure it is established. It uses the primary path through R2.**

```
R1(config)#int tun 1
R1(config-if)#NO shut
```

## Verifying the Priority constraint:

### On R1

```
R1#Sh mpls traffic-eng tunnels name R1_t1 | include path option

  path option 1, type explicit SLOW_LINK (Basis for Setup, path weight 128)
  path option 2, type explicit MEDIUM_LINK
    ID: path option 2 [16]
```

**As we can see. Tunnel "1" gets established along the primary path.  It bumps off Tunnel 0" in due to the better setup priority.**

```
R1#Sh mpls traffic-eng tunnels name R1_t0 | include path option

  path option 2, type explicit MEDIUM_LINK (Basis for Setup, path weight 128)
  path option 1, type explicit SLOW_LINK
    ID: path option 1 [33]
```

**Tunnel "0" is setup across the MEDIUM_LINK path instead.**

## Task 6

**Stop all routers in the console window and exit. Stop the Server and press the "Erase Start" launcher before proceeding.**

# Advanced
# CCIE SERVICE PROVIDER
# v3.0

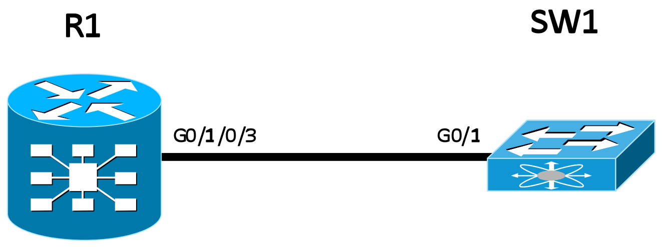
www.MicronicsTraining.com

**Narbik Kocharians**
**CCIE #12410**
**R&S, Security, SP**

**Paul Negron**
**CCIE #14856**
**SP**

## IOS-XR

# LAB-1
# IOS-XR Quick Reference



**Lab Setup:**

| Router | Interface | IP address |
|---|---|---|
| R1 | Lo0<br>G0/1/0/3 | 1.1.1.1/32<br>10.1.12.1 /24 |
| SW1 | Lo0<br>F0/0 | 2.2.2.2 /32<br>10.1.12.2 /24 |

**Task 1**

Configure a hostname on the IOS-XR platform.

**On R1**

```
12A con0/3/CPU0 is now available

Press RETURN to get started.

User Access Verification

Username:
```

**When looging into an IOS-XR platform, The prompt does not enter "user mode" as it does with IOS. A password is required.**

```
Username: cisco
Password: cisco
RP/0/3/CPU0:Mar 10 16:13:15.562 : exec[65696]: %SECURITY-login-6-
AUTHEN_SUCCESS : Successfully authenticated user 'cisco' from 'console'
on 'con0_3_CPU0'
```

```
RP/0/3/CPU0:ios#
```

**The default username is "ios" but the device will always show you which RP and slot you are logged into. In this case, it is the RP located in slot 3. The 12000 series router displays the location in te following format: "*module/slot/CPU*:name"**

```
RP/0/3/CPU0:ios#Conf
RP/0/3/CPU0:ios(config)#hostname R1
```

**Notice how the router did not apply the configuration right away.**

```
RP/0/3/CPU0:ios(config)#Show configuration

Building configuration...
!! IOS XR Configuration 0.0.0
hostname R1
end
```

**Here we see the configuration is stored in a parser that has only made sure that the sytax check has passed but the symantex check. This location is called the "target" configuration. We can make changes to the configuration or clear its contents without affecting the running configuration. This is also called the First stage of configuration. Also, notice how the "show configuration" command was executed in global configuration mode.**

```
RP/0/3/CPU0:ios(config)#Commit
RP/0/3/CPU0:Mar 10 16:20:27.844 : config[65731]: %MGBL-CONFIG-6-DB_COMMIT :
Configuration committed by user 'cisco'. Use 'show configuration commit changes
1000000020' to view the changes.
RP/0/3/CPU0:R1(config)#
```

**In order for the configuration to be a part of the running configuration, a "commit" must be issued. When a commit is issued, several actions take place. 1) A copy of the config is stored as a binary on the disk. 2) The target = running configuration. 3) The config is stored in a commit list to be viewed or to allow a rollback action just in case it caused any probems. The filed has a default label of a long indexed number, in this case, "1000000020".**

```
RP/0/3/CPU0:R1(config)#Sh config commit list

SNo. Label/ID    User     Line       Client     Time Stamp
~~~~ ~~~~~~~~    ~~~~     ~~~~       ~~~~~~     ~~~~~~~~~~
1    1000000020  cisco    con0_3_CPU  CLI       Sat Mar 10 16:20:27 2012
2    1000000019  cisco    con0_3_CPU  CLI       Sat Mar 10 16:19:01 2012
3    1000000018  cisco    con0_3_CPU  CLI       Thu Mar  8 07:00:27 2012
*** text omitted ***
```

**When we commited the configuration it caused the "1000000020" change to be created. Since it was**

**just recently commited, it appears at the top of the list ahead of the previous change.**

```
RP/0/3/CPU0:12A(config)#exit

RP/0/3/CPU0:R1#Sh config commit changes 1000000020

Building configuration...
!! IOS XR Configuration 0.0.0
hostname 12A
end
```

**Because we are viewing the changes with the "comitt" option, the individual commands that were stored in the target config just before you commited them are seen.**

```
RP/0/3/CPU0:R1#Sh configuration rollback changes 1000000020
Sat Mar 10 22:25:09.652 UTC
Building configuration...
!! IOS XR Configuration 0.0.0
hostname ios
end
```

**By using the "rollback" option, the output shows what would happen if you rolled back that change. In most cases, the "no" keyword would be inserted to negate the change. Sometimes a default may show up as the action.**

```
RP/0/3/CPU0:R1(config)#exit

RP/0/3/CPU0:12A#rollback  configuration last 1

Loading Rollback Changes.
Loaded Rollback Changes in 1 sec
Committing.
1 items committed in 1 sec (0)items/sec
Updating.RP/0/3/CPU0:Mar 10 16:23:03.572 : config_rollback[65731]: %MGBL-CONFIG-6-
DB_COMMIT : Configuration committed by user 'cisco'. Use 'show configuration commit
changes 1000000021' to view the changes.
```

**Rolling back a configuration to the last change will cause another rollback point to be created with a unique label.**

Updated Commit database in 1 sec
Configuration successfully rolled back 1 commits.

```
RP/0/3/CPU0:ios#
RP/0/3/CPU0:ios(config)#load commit changes 1000000020
Building configuration...
Loading.
```

```
48 bytes parsed in 1 sec (47)bytes/sec
```

**We can commit the individual change if needed as well.**

```
RP/0/3/CPU0:ios(config)#Sh config

Building configuration...
!! IOS XR Configuration 0.0.0
hostname R1
end
```

**When loading a configuration, it is stored in the target but still needs to be commited. It is almost like copying from a notepad and pasting to the console. The "clear" command in this mode would clear the target configuration as if you did not type anything.**

```
RP/0/3/CPU0:ios(config)#commit
RP/0/3/CPU0:Mar 10 16:25:49.783 : config[65731]: %MGBL-CONFIG-6-DB_COMMIT :
Configuration committed by user 'cisco'. Use 'show configuration commit changes
1000000022' to view the changes.
RP/0/3/CPU0:R1(config)#
```

**After commiting again, another unique label is created. A lot of time can be spent just playing with the "commit" and "rollback" feature but it is important in a real production environment to make safe changes.**

## Task 2

Configure IP connectivity between the 3400 and the 12000 Series Router using the table presented in the lab setup.

### On SW1

```
SW1(config)#int lo0
SW1(config-if)#ip add 2.2.2.2 255.255.255.255

SW1(config-if)#int g0/1
SW1(config-if)#No switchport
SW1(config-if)#ip address 10.1.12.2 255.255.255.0
SW1(config-if)#No shut
```

**The standard IOS way of doing things.**

### On R1

```
RP/0/3/CPU0:R1(config)#int lo0
RP/0/3/CPU0:R1(config-if)#ipv4 address 1.1.1.1/32

RP/0/3/CPU0:R1(config-if)#int g0/1/0/3
RP/0/3/CPU0:R1(config-if)#ipv4 address 10.1.12.1/24
RP/0/3/CPU0:R1(config-if)#No shut
RP/0/3/CPU0:R1(config-if)#end
Uncommitted changes found, commit them before exiting(yes/no/cancel)?
[cancel]:yes
```

**Here we see a few changes woth IOS-XR. The protocol must always be specified (ipv4). The mask can be configured with a decimal or bit mask format. Also, commands can often be built in the same line. AN example of thi is when applying an access-list. "int g0/1/0/3 ipv4 address 10.1.12.2/24" is acceptable but adding "no shut" on the same line would not.**

**Whe tried to end the configuration like in IOS but we did not commit first.**
**Yes = commit the changes and continue back to "EXEC" mode.**
**No = do not commit the changes and continue back to "EXEC" mode**
**Cancel = "I changed my mind so do nothing. You also stay in that mode.**

## Verify Connectivity:

### On SW1

```
SW1#Sh run int g0/1
Building configuration...

Current configuration : 102 bytes
!
interface GigabitEthernet0/1
 port-type nni   "This command is covered in the switching section"
 no switchport
 ip address 10.1.12.2 255.255.255.0
end
```

### On R1

```
RP/0/3/CPU0:R1#Sh run int g0/1/0/3

interface GigabitEthernet0/1/0/3
 ipv4 address 10.1.12.1 255.255.255.0

RP/0/3/CPU0:R1#Ping 10.1.12.2

Type escape sequence to abort.
```

```
Sending 5, 100-byte ICMP Echos to 10.1.12.2, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 2/2/3 ms
```

## Task 3

Configure IP connectivity between the 3400 and the 12000 Series Router using the table presented in the lab setup.

### On SW1

```
SW1(config)#ip routing
SW1(config)#ip route 1.0.0.0 255.0.0.0 10.1.12.1
```

### On R1

```
RP/0/3/CPU0:R1(config)#router static
RP/0/3/CPU0:R1(config-static)#address ipv4 unicast
RP/0/3/CPU0:R1(config-static-afi)#2.0.0.0/8 10.1.12.2
```

**This is very different than IOS. The IOS-XR code uses protocol based CLI. If the protocol is "address-family aware, you have an option to use it. Static routes require the address-family option. In this case, it is for ipv4.**

### Verify Connectivity:

### On SW1

```
SW1#Sh run | i ip route

 no ip route-cache cef
 no ip route-cache
ip route 1.0.0.0 255.0.0.0 10.1.12.1
```

**The "|" option can be very useful to parse portions of configuration and outout. Sometimes more filtering must be performed to take out unwanted text.**

```
SW1#Sh ip route static
S    1.0.0.0/8 [1/0] via 10.1.12.1
```

### On R1

```
RP/0/3/CPU0:R1#Sh run router static

router static
 address-family ipv4 unicast
  2.0.0.0/8 10.1.12.2
```

**The “show run” command ha smore options for fitering than IOS. We can get away without using the “|” command unless necessary.**

```
RP/0/3/CPU0:R1#Sh route static

S    2.0.0.0/8 [1/0] via 10.1.12.2, 00:08:11
```

**The “ip” keyword has been removed as an option for verifying routing information. The default is “ipv4”. We could have typed “show route ipv4”. This is fairly consistent in XR code. Other examples would be  “show bgp” or “show ospf”**

```
RP/0/3/CPU0:R1#Ping 2.2.2.2 source lo0

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2.2.2.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 22/25/34 ms
```

## Task 4

Remove the static routig and configure OSPF as the dynamic routing protocol.

### On SW1

```
SW1(config)#No ip route 1.0.0.0 255.0.0.0 10.1.12.1

SW1(config)#router ospf 1
SW1(config-router)#netw 10.1.12.0 0.0.0.255 area 0

SW1(config-router)#int lo0
SW1(config-if)#ip ospf 1 area 0

SW1(config)#router ospf 1
SW1(config-router)#passive-interface lo0
```

**Here we see the classic process based and interface based OSPF configurations. The “interface” based**

**command eliminated the need for "network" comand but was still a challenge to verify the OSPF configuration easily.**

## On R1

```
RP/0/3/CPU0:R1(config)#router ospf TST
RP/0/3/CPU0:R1(config-ospf)#router-id 0.0.0.1
RP/0/3/CPU0:R1(config-ospf)#area 0
RP/0/3/CPU0:R1(config-ospf)#mtu enable
RP/0/3/CPU0:R1(config-ospf-ar)#interface g0/1/0/3
RP/0/3/CPU0:R1(config-ospf-ar-if)#exit

RP/0/3/CPU0:R1(config-ospf-ar)#interface lo0
RP/0/3/CPU0:R1(config-ospf-ar-if)#passive
```

**The IOS-XR configuration is quite scalable and allows for inheritance of commands that is more consistence. For example, hello timers and authentication can be performed at the process, area or interface level withing the OSPF context. If cost is configured under the area level, ALL interfaces inherit that cost. If a cost is specified under the interface within the ospf configuration, it overrides the cost at all levels above it.  Since the iterface on a 3400 NNI port defaults to 9000 as the MTU and the 12000 defaulted to using an MTU if 1514, the mtu is set to enable under the area which will igmore the MTU mismatch under ospf to allow the peering to go to the full state. All of the interface inherit this and can be overridden at the interface level.**

## Verifying the Configuration:

## SW1

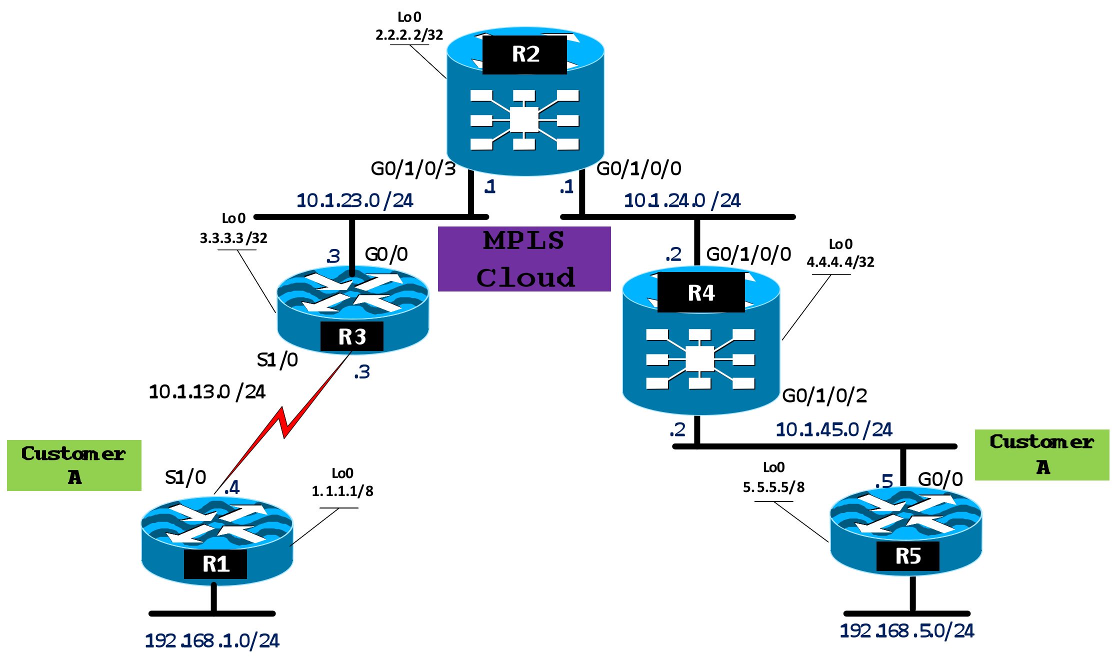
```
SW1#Sh ip route ospf | i O

O       1.1.1.1 [110/2] via 10.1.12.1, 00:05:36, GigabitEthernet0/1
```

## On R1

```
RP/0/3/CPU0:R1#Sh run router ospf

router ospf TST
 router-id 0.0.0.1
 area 0
  mtu-ignore enable
  interface Loopback0
   passive enable
  !
 interface GigabitEthernet0/1/0/3
```

```
RP/0/3/CPU0:R1#Sh ospf neighbor

* Indicates MADJ interface

Neighbors for OSPF TST

Neighbor ID     Pri   State         Dead Time   Address         Interface
0.0.0.2         1     FULL/DR       00:00:38    10.1.12.2       GigabitEthernet0/1/0/3
    Neighbor is up for 00:04:06

Total neighbor count: 1

RP/0/3/CPU0:R1#Sh route ospf

2.2.2.2/32 [110/2] via 10.1.12.2, 00:06:14, GigabitEthernet0/1/0/3

RP/0/3/CPU0:R1#Ping 2.2.2.2 so lo0

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2.2.2.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 2/2/4 ms
```

# Lab 2
# Static & RIPv2 routing in a VPN



## Lab Setup:

- The connections between R1 and R3, R4 and R5 should be configured with HDLC encapsulation. The clock rate should be set to 64000.
- Configure F0/0 of R1 in VLAN 100 and R5 in VLAN 500.
- Configure the F0/0 interface of R2 and R3 in VLAN 200.
- Configure the F0/1 interface of R2 and R4 in VLAN 300.
- Configure the rest of the routers according to the above diagram.

## Task 1

Configure OSPF on Core MPLS routers (R2, R3 and R4), you should run OSPF area 0 on the F0/0 interface of R2 and R3, F0/1 interface of R2 and R4 and the loopback interfaces of these routers.

## On R2

```
RP/0/5/CPU0:R2(config)#router ospf TST
RP/0/5/CPU0:R2(config-ospf)#area 0
RP/0/5/CPU0:R2(config-ospf-ar)#int g0/1/0/0
RP/0/5/CPU0:R2(config-ospf-ar-if)#int g0/1/0/3
RP/0/5/CPU0:R2(config-ospf-ar-if)#int lo0
RP/0/5/CPU0:R2(config-ospf-ar-if)#passive
RP/0/5/CPU0:R2(config-ospf-ar-if)#commit
RP/0/5/CPU0:R2(config-ospf-ar-if)#end
```

## On R3

```
R3(config)#router ospf 1
R3(config-router)#netw 3.3.3.3 0.0.0.0 area 0
R3(config-router)#netw 10.1.23.3 0.0.0.0 are 0
```

## On R4

```
RP/0/5/CPU0:R2(config)#router ospf TST
RP/0/5/CPU0:R2(config-ospf)#area 0
RP/0/5/CPU0:R2(config-ospf-ar)#int g0/1/0/0
RP/0/5/CPU0:R2(config-ospf-ar-if)#int lo0
RP/0/5/CPU0:R2(config-ospf-ar-if)#passive
RP/0/5/CPU0:R2(config-ospf-ar-if)#commit
RP/0/5/CPU0:R2(config-ospf-ar-if)#end
```

## To verify the configuration:

## On R3

```
R3#Sh ip route ospf | i O

       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       o - ODR, P - periodic downloaded static route, H - NHRP
O         2.2.2.2 [110/2] via 10.1.23.2, 00:07:28, GigabitEthernet0/0
O         4.4.4.4 [110/3] via 10.1.23.2, 00:01:59, GigabitEthernet0/0

O         10.1.24.0/24 [110/2] via 10.1.23.2, 00:07:28, GigabitEthernet0/0
```

## On R2

```
RP/0/5/CPU0:R2#Sh ip route ospf | i O
```

```
O    3.3.3.3/32 [110/2] via 10.1.23.3, 00:05:44, GigabitEthernet0/1/0/3
O    4.4.4.4/32 [110/2] via 10.1.24.4, 00:00:18, GigabitEthernet0/1/0/0
```

**On R4**

```
RP/0/5/CPU0:R4#Sh ip route ospf | i O

O    2.2.2.2/32 [110/2] via 10.1.24.2, 00:35:03, GigabitEthernet0/1/0/0
O    3.3.3.3/32 [110/3] via 10.1.24.2, 00:35:03, GigabitEthernet0/1/0/0
O    10.1.23.0/24 [110/2] via 10.1.24.2, 00:35:03, GigabitEthernet0/1/0/0
```

## Task 2

Configure LDP between the core routers. These routers should use their loopback 0 interfaces as their LDP router ID; the core MPLS routers (R2, R3 and R4) should use the following label range:

R2 – 65200 – 65299
R3 – 65300 – 65399
R4 – 65400 – 65499

**On R3**

```
R3(config)#MPLS label protocol LDP

R3(config)#MPLS ldp router-id lo0

R3(config)#MPLS label range 65300 65399

R3(config)#int g0/0
R3(config-if)#MPLS IP
```

**On R2**

```
RP/0/5/CPU0:R2(config)#mpls label range 65200 65299

RP/0/5/CPU0:R2(config)#mpls ldp
RP/0/5/CPU0:R2(config-ldp)#router-id 2.2.2.2
RP/0/5/CPU0:R2(config-ldp)#int g0/1/0/0
RP/0/5/CPU0:R2(config-ldp-if)#int g0/1/0/3
RP/0/5/CPU0:R2(config-ldp-if)#commit
```

```
RP/0/5/CPU0:R2(config-ldp-if)#end
```

**On R4**

```
RP/0/5/CPU0:R4(config)#mpls label range 65400 65499

RP/0/5/CPU0:R4(config)#mpls ldp
RP/0/5/CPU0:R4(config-ldp)#router-id 4.4.4.4
RP/0/5/CPU0:R4(config-ldp)#int g0/1/0/0
RP/0/5/CPU0:R4(config-ldp-if)#commit
RP/0/5/CPU0:R4(config-ldp-if)#end
```

**To verify the configuration:**

**On R4**

```
RP/0/5/CPU0:R4#Show mpls interface

Interface                  LDP      Tunnel   Enabled
-------------------------- -------- -------- --------
GigabitEthernet0/1/0/0     Yes      No       Yes

RP/0/5/CPU0:R4#Show mpls ldp neighbor

Peer LDP Identifier: 2.2.2.2:0
  TCP connection: 2.2.2.2:646 - 4.4.4.4:14775
  Graceful Restart: No
  Session Holdtime: 180 sec
  State: Oper; Msgs sent/rcvd: 11/9; Downstream-Unsolicited
  Up time: 00:01:34
  LDP Discovery Sources:
    GigabitEthernet0/1/0/0
  Addresses bound to this peer:
    2.2.2.2          10.1.23.2        10.1.24.2

RP/0/5/CPU0:R4#Sh mpls ldp discovery detail

Local LDP Identifier: 4.4.4.4:0
Discovery Sources:
  Interfaces:

    GigabitEthernet0/1/0/0 (0x2000300) : xmit/recv
      Source address: 10.1.24.4; Transport address: 4.4.4.4
      Hello interval: 5 sec (due in 917 msec)
      Quick-start: Enabled
      LDP Id: 2.2.2.2:0

          Source address: 10.1.24.2; Transport address: 2.2.2.2
          Hold time: 15 sec (local:15 sec, peer:15 sec)
                     (expiring in 12 sec)

RP/0/5/CPU0:R4#Show mpls label range

Range for dynamic labels: Min/Max: 65400/65499
```

**The default range for the labels is 16000 to 1048575 on this platform. These number ranges are software and platform specific. Over 353,000 labels are supported on 6500's as of the writing of this work book.**

## On R2

```
RP/0/5/CPU0:R2#Sh mpls interface

Interface                  LDP      Tunnel   Enabled
-------------------------- -------- -------- --------
GigabitEthernet0/1/0/0     Yes      No       Yes
GigabitEthernet0/1/0/3     Yes      No       Yes

RP/0/5/CPU0:R2#Sh mpls ldp neighbor

Peer LDP Identifier: 3.3.3.3:0
  TCP connection: 3.3.3.3:60068 - 2.2.2.2:646
  Graceful Restart: No
  Session Holdtime: 180 sec
  State: Oper; Msgs sent/rcvd: 17/18; Downstream-Unsolicited
  Up time: 00:08:14
  LDP Discovery Sources:
    GigabitEthernet0/1/0/3
  Addresses bound to this peer:
    3.3.3.3          10.1.13.3        10.1.23.3

Peer LDP Identifier: 4.4.4.4:0
  TCP connection: 4.4.4.4:14775 - 2.2.2.2:646
  Graceful Restart: No
  Session Holdtime: 180 sec
  State: Oper; Msgs sent/rcvd: 15/17; Downstream-Unsolicited
  Up time: 00:06:42
  LDP Discovery Sources:

    GigabitEthernet0/1/0/0
  Addresses bound to this peer:
    4.4.4.4          10.1.24.4        10.1.45.4        10.1.46.4

RP/0/5/CPU0:R2#Sh mpls ldp discovery

Local LDP Identifier: 2.2.2.2:0
Discovery Sources:
  Interfaces:
    GigabitEthernet0/1/0/0 : xmit/recv
      LDP Id: 4.4.4.4:0, Transport address: 4.4.4.4
          Hold time: 15 sec (local:15 sec, peer:15 sec)

    GigabitEthernet0/1/0/3 : xmit/recv
      LDP Id: 3.3.3.3:0, Transport address: 3.3.3.3
          Hold time: 15 sec (local:15 sec, peer:15 sec)

RP/0/5/CPU0:R2#Sh mpls label range

Range for dynamic labels: Min/Max: 65200/65299
```

**<u>On R3</u>**

```
R3#Sh mpls interfaces

Interface              IP            Tunnel   BGP Static Operational
GigabitEthernet0/0     Yes (ldp)     No       No  No     Yes

R3#Show mpls ldp neighbor

    Peer LDP Ident: 2.2.2.2:0; Local LDP Ident 3.3.3.3:0
        TCP connection: 2.2.2.2.646 - 3.3.3.3.23184
        State: Oper; Msgs sent/rcvd: 13/12; Downstream
        Up time: 00:04:01
        LDP discovery sources:
          GigabitEthernet0/0, Src IP addr: 10.1.23.2
        Addresses bound to peer LDP Ident:
          10.1.23.2       10.1.24.2       2.2.2.2

R3#Show mpls ldp discovery all

Local LDP Identifier:
    3.3.3.3:0
    Discovery Sources:
    Interfaces:
        GigabitEthernet0/0 (ldp): xmit/recv

            LDP Id: 2.2.2.2:0
```

```
R3#Show mpls label range

Downstream Generic label region: Min/Max label:  65300/65399
```

**<u>To verify the LFIB of these routers:</u>**

---

**<u>NOTE:</u>**

**The labels displayed in the output of this lab may differ from your lab but the principle remains the same.**

---

**<u>On R3</u>**

```
R3#Sh mpls forwarding-table

Local      Outgoing    Prefix            Bytes Label   Outgoing   Next Hop
Label      Label       or Tunnel Id      Switched      interface
65300      Pop Label   2.2.2.2/32        0             Gi0/0      10.1.23.2
65301      65201       4.4.4.4/32        0             Gi0/0      10.1.23.2
65302      Pop Label   10.1.24.0/24      0             Gi0/0      10.1.23.2
```

**<u>On R2</u>**

```
RP/0/5/CPU0:R2#Sh mpls forwarding

Local  Outgoing    Prefix             Outgoing     Next Hop        Bytes
Label  Label       or ID              Interface
Switched
------ ----------- ------------------ ------------ --------------- ------
65200  Pop         3.3.3.3/32         Gi0/1/0/3    10.1.23.3       2798
65201  Pop         4.4.4.4/32         Gi0/1/0/0    10.1.24.4       2438
```

**<u>On R4</u>**

```
RP/0/5/CPU0:R4#Sh mpls forwarding

Local  Outgoing    Prefix             Outgoing     Next Hop        Bytes
Label  Label       or ID              Interface                    Switched
------ ----------- ------------------ ------------ --------------- ------
65400  Pop         2.2.2.2/32         Gi0/1/0/0    10.1.24.2       2684
65401  65200       3.3.3.3/32         Gi0/1/0/0    10.1.24.2       0
65402  Pop         10.1.23.0/24       Gi0/1/0/0    10.1.24.2       0
```

## Task 3

Configure MP-BGP between R3 and R4 as they represent the Provider Edge routers in this topology in AS 65001. Do not allow the BGP peers to share IPV4 routing information by default. The only bgp peering relationship should be VPNv4.

### On R3

```
R3(config)#Router bgp 65001
R3(config-router)#NO BGP default ipv4-unicast
R3(config-router)#Neighbor 4.4.4.4 remote-as 65001
R3(config-router)#Neighbor 4.4.4.4 Update-source Lo0
```

**Note the exchange of IPv4 routes between BGP neighbors are enabled by default on a this 7200 platform, which means the configured neighbors will NOT only establish a BGP session but they will also receive the advertised prefixes.**

**Since the "NO bgp default ipv4-unicast" is configured first (Before the neighbor commands), the local router will NOT establish a BGP peer session with any neighbor unless that neighbor is activated.**

### On R4

```
RP/0/5/CPU0:R4(config)#router bgp 65001
RP/0/5/CPU0:R4(config-bgp)#neighbor 3.3.3.3
RP/0/5/CPU0:R4(config-bgp-nbr)#remote-as 65001
RP/0/5/CPU0:R4(config-bgp-nbr)#update-source lo0
RP/0/5/CPU0:R4(config-bgp-nbr)#commit
```

**Note the exchange of IPv4 routes between BGP neighbors is NOT enabled by default on an IOS-XR platform. The BGP address-family must be declared directly under the process to be activated. The neighbor must also be manually configured to participate in any address-family that has been declared.**

### To verify the configuration:

### On R3

```
R3#Show ip bgp
R3#

R3#Show ip bgp summary
R3#
```

**Note there is no IPv4 neighbor adjacency established between the two routers. Since the task states that these two BGP speakers should ONLY establish a VPNv4 peer session, the neighbors MUST be activated in the address-family VPNv4.**

**On R3**

```
R3(config)#Router bgp 65001
R3(config-router)#Address-family vpnv4 unicast
R3(config-router-af)#Neighbor 4.4.4.4 Activate
```

**On R4**

```
RP/0/5/CPU0:R4(config)#router bgp 65001
RP/0/5/CPU0:R4(config-bgp)#address-family vpnv4 unicast
RP/0/5/CPU0:R4(config-bgp-af)#neighbor 3.3.3.3
RP/0/5/CPU0:R4(config-bgp-nbr)#address-family vpnv4 unicast
RP/0/5/CPU0:R4(config-bgp-nbr-af)#commit
```

**You should see the following console message on R3 :**

```
%BGP-5-ADJCHANGE: neighbor 4.4.4.4 Up
```

**To verify the configuration:**

**On R3**

```
R3#Show ip bgp VPNv4 all summary | B Neighbor

Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
4.4.4.4         4 65001      12      12        1    0    0 00:08:44        0
```

**On R4**

```
RP/0/5/CPU0:R4#sh bgp vpnv4 unicast summary | b Neighbor

Neighbor        Spk    AS MsgRcvd MsgSent   TblVer  InQ OutQ  Up/Down  St/PfxRcd
3.3.3.3           0 65001       4       3        1    0    0 00:01:06          0
```

**NOTE: The BGP speakers have ONLY established a VPNv4 peering.**

**On R3**

```
RP/0/5/CPU0:R4#Sh bgp summ | b Neigbor
```

**On R4**

```
RP/0/5/CPU0:R4#Sh bgp summ

% None of the requested address families are configured for instance
```

```
'default'
```

**On R2**

```
RP/0/5/CPU0:R2#Show bgp summ

% BGP instance 'default' not active
```

**Note R2 is NOT running BGP at all, and ONLY the edge routers are running BGP. Running BGP on the core routers (P) is NOT necessary; these routers ONLY perform label switching.**

## Task 4

Configure a **Virtual Routing Forwarding (VRF) Instance with a name of CA** (For Customer A), a **route-distinguisher (RD) of "1:10"** and a **route-target (RT) of "1:100"** on R3. On R4, the same route targets will be used for the vrf, but the **RD should be configured to be "1:20"** and the name of the VRF should be configured as **CB**.

**The "IP vrf" command creates a new VRF and enters the global configuration mode for that specific VRF. The name of the VRF is locally significant and it's case sensitive. VRF is NOT operational unless the RD is defined.**

**The "RD" VRF configuration mode command is used to define and assign an RD to a VRF, remember that the VRF is NOT operational without an RD. RD is a 64 bit value used to transform 32 bit customer IPv4 address, which is NON-Unique into a Unique 96 bit addresses called VPNv4. These addresses are ONLY exchanged between the PE routers and NEVER between the CE routers.**

**When the CE router sends an update to a PE router, the PE router prepends a 64 bit RD to the IPv4 address (32 bit address) resulting in a globally unique 96 bit address called VPNv4.**

**The PE router will then send the VPNv4 address/es via MP-BGP session to the other PE router/s. The receiving PE router strips the RD from the VPNv4 prefix, resulting in an IPv4 address.**

**Remember that the RD does NOT indicate which VRF a given prefix belongs to, it's used to make the**

**VRF prefix/es unique within the cloud. RDs DO NOT identify the VPN.**

**Let's configure the VRFs and the RDs:**

**On R3**

```
R3(config)#ip vrf CA
R3(config-vrf)#rd 1:10
```

**Since VRFs are locally significant and the RDs will keep the CE routes Unique within the cloud, and the receiving PE will strip off the 64 bit RD value, how does the receiving PE know which VRF does the IP address belong to? The answer is "Route-Target".**

**The "Route-target import|export" command defines the "RT"; An RT is a BGP extended community that indicates which routes should be exported or imported from MP-BGP into the VRF.**

**Basically RTs were introduced to support identifying a site that participates in more than one VPN.**

**The "route-target export" command specifies an RT is to be attached to every route exported from the local VRF to MP-BGP. Whereas, the "route-target import" command specifies an RT to be used as an import filter, ONLY routes matching the RT are imported into the VRF.**
**This implementation allows a route to have many imported or exported RTs, all to be attached to every imported or exported route.**

**Let's configure RTs:**

## On R3

```
R3(config)#ip vrf CA
R3(config-vrf)#route-target import 1:100
R3(config-vrf)#route-target export 1:100
```

**On an IOS platform, the "both" keyword can be used to replace both "import" and "export" keywords with one statement.**

## On R4

```
RP/0/5/CPU0:R4(config)#vrf CB
RP/0/5/CPU0:R4(config-vrf)#address-family ipv4 unicast
RP/0/5/CPU0:R4(config-vrf-af)#import route-target 1:100
RP/0/5/CPU0:R4(config-vrf-af)#export route-target 1:100
RP/0/5/CPU0:R4(config-vrf-af)#commit
```

**The VRF's are configured slightly different in IOS-XR. The Address-Family MUST be noted and the RD is not configured in this context as it is a strictly used as a  BGP control Plane function.**

**Route-Targets are BGP extended communities that are attached to the VPNv4 addresses, and communities are NOT sent unless they are configured to be sent in IOS. In IOS-XR all communities are propagated to IBGP neighbors by default.**

**Let's configure R3 to send the extended communities:**

__On R3__

```
R3(config)#Router bgp 65001
R3(config-router)#Address-family vpnv4 unicast
R3(config-router-af)#Neighbor 4.4.4.4 Send-community extended
```

__On R4__

```
RP/0/5/CPU0:R4(config)#router bgp 65001
RP/0/5/CPU0:R4(config-bgp)#address-family vpnv4 unicast
RP/0/5/CPU0:R4(config-bgp-af)#vrf CB
RP/0/5/CPU0:R4(config-bgp-vrf)#rd 1:20
RP/0/5/CPU0:R4(config-bgp-vrf)#commit
```

**Here we see where the RD is configured. It is under the VRF but the vrf is applied to the Router BGP process first.**

__To verify the VRF Configuration:__

__On R3__

```
R3#Sh ip vrf detail

VRF CA (VRF Id = 1); default RD 1:10; default VPNID <not set>
  No interfaces
VRF Table ID = 1
  Export VPN route-target communities
    RT:1:100
  Import VPN route-target communities
    RT:1:100
  No import route-map
  No export route-map
  VRF label distribution protocol: not configured
  VRF label allocation mode: per-prefix
```

**Think of VRFs as VLANs; once a VLAN is configured, the appropriate interface/s are assigned to that VLAN. The same philosophy applies to VRFs, you can think of VRFs as layer three VLANs, in this case, the VRFs are configured, and now the appropriate interface/s needs to be assigned to that VRF:**

**To associate an interface/s to a given VRF:**

__On R3__

```
R3(config)#int s1/0
R3(config-if)#ip vrf forwarding CA
```

**The above command associates an interface with the specified VRF; remember when this command is applied to a given interface, the IP address of that interface is removed and it should be reconfigured.**

**You should get the following console message:**

```
% Interface Serial0/1 IP address 10.1.13.3 removed due to enabling VRF CA

R3(config-if)#ip address 10.1.13.3 255.255.255.0
```

**Let's configure R4:**

## On R4

```
RP/0/5/CPU0:R4(config)#int g0/1/0/2
RP/0/5/CPU0:R4(config-if)#vrf CB
RP/0/5/CPU0:R4(config-if)#commit

% Failed to commit one or more configuration items during a pseudo-atomic
operation. All changes made have been reverted. Please issue 'show
configuration failed' from this session to view the errors

RP/0/5/CPU0:R4(config-if)#root
RP/0/5/CPU0:R4(config)#Show config failed

!! SEMANTIC ERRORS: This configuration was rejected by
!! the system due to semantic errors. The individual
!! errors with each failed configuration command can be
!! found below.

interface GigabitEthernet0/1/0/2
 vrf CB
!!% 'RSI' detected the 'fatal' condition 'The interface's numbered and
unnumbered IPv4/IPv6 addresses must be removed prior to changing or
deleting the VRF'
!
end

RP/0/5/CPU0:R4(config)#clear
RP/0/5/CPU0:R4(config)#int g0/1/0/2
RP/0/5/CPU0:R4(config-if)#No ipv4 address
RP/0/5/CPU0:R4(config-if)#vrf CB
RP/0/5/CPU0:R4(config-if)#ipv4 address 10.1.45.4/24
RP/0/5/CPU0:R4(config-if)#commit
```

## To verify the configuration:

## On R4

```
RP/0/5/CPU0:R4#Show vrf CB detail

VRF CB; RD 1:20; VPN ID not set
VRF mode: Regular
Description not set
Interfaces:
  GigabitEthernet0/1/0/2
Address family IPV4 Unicast
  Import VPN route-target communities:
    RT:1:100
  Export VPN route-target communities:
    RT:1:100
  No import route policy
  No export route policy
Address family IPV6 Unicast
  No import VPN route-target communities
  No export VPN route-target communities
  No import route policy
  No export route policy
```

**The above output begins with the name of the VRF, the default Route Distinguisher value and the VPNID which is normally not set by default. The VPNID is an extension used to further identify the VPN by using the customer "OUI" and "vpn index" that can be keyed in as a Hexadecimal or Decimal formatted number.**

**The interfaces that have the "VRF CB" applied will be listed under the Interfaces: list along with the warning that Connected addresses are no longer in the global routing table. It is a good practice to have at least 1 route target that the router will be both EXPORTING and IMPORTING. The 1:100 Route Target follows this basic design rule.**

**Currently there is no Export or Import route-maps applied. VRF-SELECT will be discussed later.**

## On R3

```
R3#Show ip vrf CA

Name                             Default RD          Interfaces
CA                               1:10                Se0/1

R3#Sh ip vrf interfaces

Interface              IP-Address      VRF                      Protocol
Se1/0                  10.1.13.3       CA                       up
```

## To verify the connectivity between PEs and the CEs:

### On R4:

```
R4#Ping 10.1.45.5

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.45.5, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```

**A regular ping will no longer work. A ping with no other key words will default to using the global routing table. The 10.1.45.0 /24 prefix is not accessible in the global routing table anymore. The ping must be added with the proper VRF keyword. Remember the IP address is in the VRF and NOT the global routing table.**

```
R4#Ping VRF CB 10.1.45.5

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.45.5, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/32 ms
```

### On R3:

```
R3#Ping VRF CA 10.1.13.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.13.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/29 ms
```

## Task 5

Configure a static default route on each Customer router located in CA and CB; these static routes should be configured to point to their respective PE router (**R3** for R1 and **R4** for R5). The PE Routers (R3 and R4), should both be configured with a static route that reaches the loopback and F0/0 interface of the Customer router, R3 and R4 should be able to see both static routes in their BGP table.

## To configure a default route on the CEs:

**On R1**

```
R1(config)#ip route 0.0.0.0 0.0.0.0 10.1.13.3
```

**On R5**

```
R5(config)#ip route 0.0.0.0 0.0.0.0 10.1.45.4
```

**Just a regular default route from the customer's perspective. This method is one of the recommended methods that most Service Providers prefer when offering MPLS VPN as a service to a very small company.**

## To configure a static route on the PEs:

**Note the output of the following show command reveals that when the VRF forwarding was enabled on S1/0 interface, the "Address-family IPv4 VRF CA" was added to the BGP configuration.**

**On R4**

```
RP/0/5/CPU0:R4(config)#router static
RP/0/5/CPU0:R4(config-static)#vrf CB
RP/0/5/CPU0:R4(config-static-vrf)#address-family ipv4 uni
RP/0/5/CPU0:R4(config-static-vrf-afi)#5.0.0.0/8 10.1.45.5
RP/0/5/CPU0:R4(config-static-vrf-afi)#192.168.5.0/24 10.1.45.5
RP/0/5/CPU0:R4(config-static-vrf-afi)#commit
```

**On R3**

```
R3(config)#ip route vrf CA 1.0.0.0 255.0.0.0 10.1.13.1
R3(config)#ip route vrf CA 192.168.1.0 255.255.255.0 10.1.13.1
```

## To verify and test the configuration:

**On R3**

```
R3#Show ip route vrf CA | b Gateway
Gateway of last resort is not set

S     1.0.0.0/8 [1/0] via 10.1.13.1
      10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C        10.1.13.0/24 is directly connected, Serial1/0
L        10.1.13.3/32 is directly connected, Serial1/0
S     192.168.1.0/24 [1/0] via 10.1.13.1
```

```
R3#Ping vrf CA 1.1.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 1.1.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/28 ms

R3#Ping vrf CA 192.168.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/28 ms
```

**On R4**

```
RP/0/5/CPU0:R4#Show route vrf CB | b Gateway
Gateway of last resort is not set

S    5.0.0.0/8 [1/0] via 10.1.45.5, 00:07:41
C    10.1.45.0/24 is directly connected, 00:27:56, GigabitEthernet0/1/0/2
L    10.1.45.4/32 is directly connected, 00:27:56, GigabitEthernet0/1/0/2
S    192.168.5.0/24 [1/0] via 10.1.45.5, 00:07:41

R4#Ping vrf CB 5.5.5.5

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 5.5.5.5, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/29/32 ms

R4#Ping vrf CB 192.168.5.5

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.5.5, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/56/60 ms
```

**In the last step of this task, the static routes that were configured on the PE routers should be redistributed into BGP so the neighboring PE router can see the route in its BGP table. This redistribution should be done under another address-family called "IPv4 Unicast"; this address-family is created automatically when the VRF, RD and the RTs are configured; the following show command verifies this fact:**

**On R4**

```
R4#Sh run | B router bgp

RP/0/5/CPU0:R4#sh run router bgp
Sat Apr 14 22:59:03.553 UTC
router bgp 65001
 address-family vpnv4 unicast
 !
 neighbor 3.3.3.3
  remote-as 65001
  update-source Loopback0
  address-family vpnv4 unicast
  !
 !
 vrf CB
  rd 1:20
```

**On R3**

```
R3#Sh run | B router bgp

router bgp 65001
 no bgp default ipv4-unicast
 bgp log-neighbor-changes
 neighbor 4.4.4.4 remote-as 65001
 neighbor 4.4.4.4 update-source Loopback0
 !
 address-family vpnv4
  neighbor 4.4.4.4 activate
  neighbor 4.4.4.4 send-community extended
 exit-address-family
 !
 address-family ipv4 vrf CA
  no synchronization
 exit-address-family
```

**The address-family shows up automatically in most IOS platforms but may not show up on others. If it does NOT show up, the next step will configure the address-family to redistribute the static routes. To configure the redistribution of the static routes:**

**On R4**

```
RP/0/5/CPU0:R4(config)#router bgp 65001
RP/0/5/CPU0:R4(config-bgp)#address-family vpnv4 unicast

RP/0/5/CPU0:R4(config-bgp-af)#vrf CB
RP/0/5/CPU0:R4(config-bgp-vrf)#address-family ipv4 unicast
```

```
RP/0/5/CPU0:R4(config-bgp-vrf-af)#redistribute static
RP/0/5/CPU0:R4(config-bgp-vrf-af)#commit
RP/0/5/CPU0:R4(config-bgp-vrf-af)#end
```

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

**<u>On R3</u>**

```
R3#Show ip route vrf CA | b Gateway
Gateway of last resort is not set

S    1.0.0.0/8 [1/0] via 10.1.13.1
B    5.0.0.0/8 [200/0] via 4.4.4.4, 00:00:27
B    192.168.5.0/24 [200/0] via 4.4.4.4, 00:00:27
     10.0.0.0/24 is subnetted, 1 subnets
C       10.1.13.0 is directly connected, Serial0/1
S    192.168.1.0/24 [1/0] via 10.1.13.1
```

**NOTE: The redistributed routes show up on R3 as BGP routes. Let's configure R3 to redistribute its static routes:**

**<u>On R3</u>**

```
R3(config)#Router bgp 65001
R3(config-router)#Address-family IPv4 VRF CA
R3(config-router-af)#Redistribute static
```

**<u>On R4</u>**

```
R4#Show ip route vrf CB | B Gateway
Gateway of last resort is not set

B    1.0.0.0/8 [200/0] via 3.3.3.3, 00:02:51
S    5.0.0.0/8 [1/0] via 10.1.45.5
S    192.168.5.0/24 [1/0] via 10.1.45.5
     10.0.0.0/24 is subnetted, 1 subnets
C       10.1.45.0 is directly connected, Serial0/1
B    192.168.1.0/24 [200/0] via 3.3.3.3, 00:02:51
```

**Note the redistributed routes should be verified on R3, if the configuration is performed successfully, R3 should be able to see networks 5.0.0.0 /8 and 192.168.5.0 /24 in its vrf CA routing table.**

**To verify the routes in VPNv4:**

**<u>On R3</u>**

```
R3#Show ip bgp vpnv4 all | B Network

   Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 1:10 (default for vrf CA)
*> 1.0.0.0          10.1.13.1                0         32768 ?
*>i5.0.0.0          4.4.4.4                  0    100      0 ?
*> 192.168.1.0      10.1.13.1                0         32768 ?
*>i192.168.5.0      4.4.4.4                  0    100      0 ?
Route Distinguisher: 1:20
*>i5.0.0.0          4.4.4.4                  0    100      0 ?
*>i192.168.5.0      4.4.4.4                  0    100      0 ?
```

## On R4

```
R4#Show ip bgp vpnv4 unicast | B Network

   Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 1:10
*>i1.0.0.0          3.3.3.3                  0    100      0 ?
*>i192.168.1.0      3.3.3.3                  0    100      0 ?
Route Distinguisher: 1:20 (default for vrf CB)
*>i1.0.0.0          3.3.3.3                  0    100      0 ?
*> 5.0.0.0          10.1.45.5                0         32768 ?
*>i192.168.1.0      3.3.3.3                  0    100      0 ?
*> 192.168.5.0      10.1.45.5                0         32768 ?
```

**One of the reasons to use different Route Distinguishers for ALL VRFs in the network is for clarity, especially when trying to figure out which VPN a route belongs to and where it's originated. As you can see, the "1:20" RD is local, but the 1.0.0.0 & 192.168.1.0 networks are present in both RDs, which means that 1.0.0.0 & 192.168.1.0 networks are participating in the same VPN. Even though the VPN is defined by the Route Target, it is still apparent that the prefixes are being imported and exported into both VRFs.**

## To test the configuration:

## On R1

```
R1#Ping 5.5.5.5 source 1.1.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 5.5.5.5, timeout is 2 seconds:
Packet sent with a source address of 1.1.1.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/56/60 ms
```

```
R1#Ping 5.5.5.5 source 192.168.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 5.5.5.5, timeout is 2 seconds:
Packet sent with a source address of 192.168.1.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 84/84/88 ms
```

**On R5**

```
R5#Ping 1.1.1.1 source 5.5.5.5

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 = 56/56/57 ms

R5#Ping 1.1.1.1 source 192.168.5.5

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 192.168.5.5
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 84/84/84 ms
```

**Note the source of the two ping commands are specified to be the loopback 0 & F0/0's IP address, if the source is NOT specified, the Ping will NOT be successful:**

**On R5**

```
R5#Ping 1.1.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 1.1.1.1, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```

**On R1**

```
R1#Ping 5.5.5.5

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 5.5.5.5, timeout is 2 seconds:
.....
```

```
Success rate is 0 percent (0/5)
```

**The ping failed because the source was NOT set based on the loopback0 or the F0/0 interface, if the source is not specified, the source IP address will be the closest interface, in this case the IP address of the S0/1 interface. In order to provide the capability of Pinging without specifying the source IP address, the PE routers should also redistribute the IP address of their S0/1 interface, this can be done using the "Redistribute Connected":**

## On R3

```
R3(config)#router bgp 65001
R3(config-router)#Address-family IPv4 vrf CA
R3(config-router-af)#Redistribute connected
```

## To verify the configuration:

## On R4

```
RP/0/5/CPU0:R4#Sh route vrf CB | b Gateway
Gateway of last resort is not set

B    1.0.0.0/8 [200/0] via 3.3.3.3 (nexthop in vrf default), 00:13:58
S    5.0.0.0/8 [1/0] via 10.1.45.5, 00:22:38
B    10.1.13.0/24 [200/0] via 3.3.3.3 (nexthop in vrf default), 00:00:15
C    10.1.45.0/24 is directly connected, 00:22:38, GigabitEthernet0/1/0/2
L    10.1.45.4/32 is directly connected, 00:22:38, GigabitEthernet0/1/0/2
B    192.168.1.0/24 [200/0] via 3.3.3.3 (nexthop in vrf default),
00:13:58
S    192.168.5.0/24 [1/0] via 10.1.45.5, 00:22:38
```

## On R4

```
RP/0/5/CPU0:R4(config)#router bgp 65001
RP/0/5/CPU0:R4(config-bgp)#address vpnv4 uni
RP/0/5/CPU0:R4(config-bgp-af)#vrf CB
RP/0/5/CPU0:R4(config-bgp-vrf)#address ipv4 uni
RP/0/5/CPU0:R4(config-bgp-vrf-af)#redistribute connected
RP/0/5/CPU0:R4(config-bgp-vrf-af)#commit
RP/0/5/CPU0:R4(config-bgp-vrf-af)#end
```

## To verify the configuration:

## On R3

```
R3#Sh ip route vrf CA | b Gateway
Gateway of last resort is not set

S     1.0.0.0/8 [1/0] via 10.1.13.1
B     5.0.0.0/8 [200/0] via 4.4.4.4, 00:17:08
      10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
C        10.1.13.0/24 is directly connected, Serial1/0
L        10.1.13.3/32 is directly connected, Serial1/0
B        10.1.45.0/24 [200/0] via 4.4.4.4, 00:01:12
S     192.168.1.0/24 [1/0] via 10.1.13.1
B     192.168.5.0/24 [200/0] via 4.4.4.4, 00:17:08
```

**To test the connectivity:**

**On R1**

```
R1#Ping 5.5.5.5

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 5.5.5.5, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/57/60 ms
```

**On R5**

```
R5#Ping 1.1.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 1.1.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/56/57 ms
```

**To verify the extended community attached to a given prefix:**

**On R4**

```
RP/0/5/CPU0:R4#Sh bgp vpnv4 uni vrf CB 5.0.0.0

BGP routing table entry for 5.0.0.0/8, Route Distinguisher: 1:20
Versions:

  Process           bRIB/RIB  SendTblVer
  Speaker                  4           4
    Local Label: 65403
```

```
Last Modified: Apr 14 23:16:20.621 for 00:32:04
Paths: (1 available, best #1)
  Advertised to peers (in unique update groups):
    3.3.3.3
  Path #1: Received by speaker 0
  Advertised to peers (in unique update groups):
    3.3.3.3
  Local
    10.1.45.5 from 0.0.0.0 (4.4.4.4)
      Origin incomplete, metric 0, localpref 100, weight 32768, valid,
redistributed, best, group-best, import-candidate
      Received Path ID 0, Local Path ID 1, version 4
      Extended community: RT:1:100
```

**We have successfully verified that the route-target has been added to the routes as they are exported from the local VRF. This is important as NO vrf will be able to import this route if the extended community is not added to the prefix. The VPNv4 label has also been added to the prefix. The "65403(Local Label) and the nolabel (Rcvd)" shows that this router expects to see this route with label 65403 when it recieves the data traffic and will remove the label before forwarding it to the Customer (R5) router.**

```
RP/0/5/CPU0:R4#Sh bgp vpnv4 uni labels | b Netw

Sat Apr 14 23:50:11.534 UTC
   Network            Next Hop        Rcvd Label      Local Label
Route Distinguisher: 1:10
*>i1.0.0.0/8          3.3.3.3         65303           nolabel
*>i10.1.13.0/24       3.3.3.3         65305           nolabel
*>i192.168.1.0/24     3.3.3.3         65304           nolabel
Route Distinguisher: 1:20 (default for vrf CB)
*>i1.0.0.0/8          3.3.3.3         65303           nolabel
*> 5.0.0.0/8          10.1.45.5       nolabel         65403
*>i10.1.13.0/24       3.3.3.3         65305           nolabel
*> 10.1.45.0/24       0.0.0.0         nolabel         65405
*>i192.168.1.0/24     3.3.3.3         65304           nolabel
*> 192.168.5.0/24     10.1.45.5       nolabel         65404
```

**This command displays the VPNv4 labels that are added to the routes. A common question that is often asked is "What does the aggregate keyword mean?" when it shows up in the table. The aggregate keyword means that the route requires an IP lookup to determine the next hop of this packet. When we redistributed the connected routes, the far end does not know about this next hop due to the fact that this route is not in the VPN.**

**By redistributing the connected route, a normal ping from the interface closest to the destination can be performed by R1 due to the fact that R5 now knows about the source of the packet.**

```
RP/0/5/CPU0:R4#Sh mpls forwarding vrf CB

Local  Outgoing    Prefix             Outgoing     Next Hop        Bytes
Label  Label       or ID              Interface
Switched
------ ----------- ------------------ ------------ --------------- -----
65403  Unlabelled  5.0.0.0/8[V]       Gi0/1/0/2    10.1.45.5       0
65404  Unlabelled  192.168.5.0/24[V]  Gi0/1/0/2    10.1.45.5       0
65405  Aggregate   CB: Per-VRF Aggr[V]   \
                                      CB                           0
```

**On R3**

```
R3#Show mpls forwarding-table vrf CA

Local   Outgoing    Prefix            Bytes tag  Outgoing   Next Hop
tag     tag or VC   or Tunnel Id      switched   interface
303     Untagged    1.0.0.0/8[V]      2600       Se0/1      point2point
304     Untagged    192.168.1.0/24[V] 520        Se0/1      point2point
305     Aggregate   10.1.13.0/24[V]   520
```

## Task 6

Remove the Static routes and replace the current method of routing between the PE and Customers with RIPv2 Routing.

**Removing the Static Configuration on the CE routers:**

**On R1**

```
R1(config)#NO ip route 0.0.0.0 0.0.0.0
```

**On R5**

```
R5(config)#NO ip route 0.0.0.0 0.0.0.0
```

**Removing the Static Configuration on the PE routers:**

**On R3**

```
R3(config)#NO ip route vrf CA 1.0.0.0 255.0.0.0
R3(config)#NO ip route vrf CA 192.168.1.0 255.255.255.0
```

```
R3(config)#router bgp 65001
R3(config-router)# address ipv4 vrf CA
R3(config-router-af)#NO redistribute static
R3(config-router-af)#NO redistribute connected
```

**On R4**

```
RP/0/5/CPU0:R4(config)#NO router static
RP/0/5/CPU0:R4(config)#commit

RP/0/5/CPU0:R4(config)#router bgp 65001
RP/0/5/CPU0:R4(config-bgp)#vrf CB
RP/0/5/CPU0:R4(config-bgp-vrf)#address-family ipv4 uni
RP/0/5/CPU0:R4(config-bgp-vrf-af)#NO redistribute static
RP/0/5/CPU0:R4(config-bgp-vrf-af)#NO redistribute connected
RP/0/5/CPU0:R4(config-bgp-vrf-af)#commit
```

## Configuring RIPv2 routing on CE routers:

**On R1**

```
R1(config)#router rip
R1(config-router)#No au
R1(config-router)#ver 2
R1(config-router)#netw 10.0.0.0
R1(config-router)#netw 192.168.1.0
R1(config-router)#netw 1.0.0.0
```

**On R5**

```
R5(config)#router rip
R5(config-router)#No au
R5(config-router)#ver 2
R5(config-router)#network 10.0.0.0

R5(config-router)#netw 5.0.0.0
R5(config-router)#netw 192.168.5.0
```

## Configuring RIPv2 routing on PE routers:

**On R3**

```
R3(config)#router rip
R3(config-router)#ver 2
R3(config-router)#Address-family ipv4 vrf CA
```

```
R3(config-router-af)#Network 10.0.0.0
R3(config-router-af)#version 2
R3(config-router-af)#No au
```

## On R4

```
RP/0/5/CPU0:R4(config)#router rip
RP/0/5/CPU0:R4(config-rip)#vrf CB
RP/0/5/CPU0:R4(config-rip-vrf)#No auto
RP/0/5/CPU0:R4(config-rip-vrf)#int g0/1/0/2
```

## To verify the configuration:

## On R3

```
R3#Sh ip route vrf CA | b Gateway
Gateway of last resort is not set

      1.0.0.0/32 is subnetted, 1 subnets
R        1.1.1.1 [120/1] via 10.1.13.1, 00:00:06, Serial1/0
      10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C        10.1.13.0/24 is directly connected, Serial1/0
L        10.1.13.3/32 is directly connected, Serial1/0
R     192.168.1.0/24 [120/1] via 10.1.13.1, 00:00:06, Serial1/0
```

## On R4

```
RP/0/5/CPU0:R4#Sh route vrf CB | b Gateway
Gateway of last resort is not set

R    5.5.5.5/32 [120/1] via 10.1.45.5, 00:01:46, GigabitEthernet0/1/0/2
C    10.1.45.0/24 is directly connected, 17:53:07, GigabitEthernet0/1/0/2
L    10.1.45.4/32 is directly connected, 17:53:07, GigabitEthernet0/1/0/2
R    192.168.5.0/24 [120/1] via 10.1.45.5, 00:01:46,
GigabitEthernet0/1/0/2
```

**Note the routes are in the appropriate VRFs, the next step is to redistribute the routes.**

**The redistribution of RIPv2 into MP-BGP is necessary for the routes to show up on the other PE Router.**

**The redistribution of MP-BGP into RIP is necessary for the local router to translate the routes that have been received by the remote PE router so the CE router/s can see the routes.**

## Step One

**RIP routes are redistributed into the BGP for the specified VRF:**

## On R4

```
RP/0/5/CPU0:R4(config)#router bgp 65001
RP/0/5/CPU0:R4(config-bgp)#vrf CB
RP/0/5/CPU0:R4(config-bgp-vrf)#address-family ipv4 unicast
RP/0/5/CPU0:R4(config-bgp-vrf-af)#redistribute rip
RP/0/5/CPU0:R4(config-bgp-vrf-af)#commit
```

## To verify the configuration:

**Because of the redistribution, the PE router (R3) on the other side can now see the routes in its BGP table:**

## On R3

```
R3#Sh ip bgp vpnv4 vrf CA | b Netw

   Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 1:10 (default for vrf CA)
*>i5.5.5.5/32       4.4.4.4                  1    100      0 ?
*>i10.1.45.0/24     4.4.4.4                  0    100      0 ?
*>i192.168.5.0      4.4.4.4                  1    100      0 ?
```

## Step Two

**In this step the routes are redistributed into RIP, so the CE router will have them in its routing table:**

## On R3

**When the routes are redistributed into RIP, a metric of 5 is assigned; this is done on purpose so they can easily be identified in the routing table of the Customer router (R1):**

```
R3(config)#router rip
R3(config-router)#Address-family ipv4 vrf CA
R3(config-router-af)#redistribute BGP 65001 metric 5
```

## To verify the configuration:

## On R1

**Note R1 (The CE router) has the routes in its routing table:**

```
R1#Show ip route rip | Inc R

R    5.0.0.0/8 [120/5] via 10.1.13.3, 00:00:08, Serial0/1
R    192.168.5.0/24 [120/5] via 10.1.13.3, 00:00:08, Serial0/1
R       10.1.45.0 [120/5] via 10.1.13.3, 00:00:08, Serial0/1
```

**The same needs to be done in reverse order from R3 to R4 and verified on R5:**

## On R3

```
R3(config)#router bgp 65001
R3(config-router)#Address-family ipv4 vrf CA
R3(config-router-af)#Redistribute RIP
```

## On R4

```
RP/0/5/CPU0:R4(config)#router rip
RP/0/5/CPU0:R4(config-rip)#vrf CB
RP/0/5/CPU0:R4(config-rip-vrf)#redistribute bgp 65001 internal
RP/0/5/CPU0:R4(config-rip-vrf)#commit
```

## To verify the configuration:

## On R5

```
R5#Show ip route rip | Inc R

R5#sh ip route rip | i R
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       o - ODR, P - periodic downloaded static route, H - NHRP
R        1.1.1.1 [120/2] via 10.1.45.4, 00:00:01, GigabitEthernet0/0
R     192.168.1.0/24 [120/2] via 10.1.45.4, 00:00:01, GigabitEthernet0/0
```

**To see the full picture, the following show commands verifies network 1.0.0.0 /8 from R1 all the way to the upstream router R5; this is called the control plane.**

## On R1

```
R1#Sh ip route 1.0.0.0

Routing entry for 1.0.0.0/8, 2 known subnets
  Attached (2 connections)
  Variably subnetted with 2 masks
  Redistributing via rip
```

```
C        1.0.0.0/8 is directly connected, Loopback0
L        1.1.1.1/32 is directly connected, Loopback0

R1#Sh ip route 1.1.1.1

Routing entry for 1.1.1.1/32
  Known via "connected", distance 0, metric 0 (connected)
  Routing Descriptor Blocks:
  * directly connected, via Loopback0
      Route metric is 0, traffic share count is 1

R1#Show ip rip database 1.0.0.0 255.0.0.0

1.0.0.0/8    directly connected, Loopback0
```

**On R3**

```
R3#Sh ip route vrf CA | b Gateway
Gateway of last resort is not set

      1.0.0.0/8 is subnetted, 1 subnets
R        1.0.0.0 [120/1] via 10.1.13.1, 00:00:30, Serial1/0
      5.0.0.0/8 is subnetted, 1 subnets
B        5.0.0.0 [200/1] via 4.4.4.4, 00:03:37
      10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
C        10.1.13.0/24 is directly connected, Serial1/0
L        10.1.13.3/32 is directly connected, Serial1/0
B        10.1.45.0/24 [200/0] via 4.4.4.4, 00:18:50
R     192.168.1.0/24 [120/1] via 10.1.13.1, 00:00:30, Serial1/0
B     192.168.5.0/24 [200/1] via 4.4.4.4, 00:18:50

R3#Sh ip route vrf CA 1.1.1.1

Routing Table: CA
Routing entry for 1.0.0.0/8
  Known via "rip", distance 120, metric 1
  Redistributing via bgp 65001, rip
  Advertised by bgp 65001
  Last update from 10.1.13.1 on Serial1/0, 00:00:11 ago
  Routing Descriptor Blocks:
  * 10.1.13.1, from 10.1.13.1, 00:00:11 ago, via Serial1/0
      Route metric is 1, traffic share count is 1

R3#Sh ip bgp vpnv4 vrf CA | b Network

   Network          Next Hop            Metric LocPrf Weight Path
```

```
Route Distinguisher: 1:10 (default for vrf CA)
*> 1.0.0.0          10.1.13.1                1         32768 ?
*>i5.0.0.0          4.4.4.4                  1    100      0 ?
*> 10.1.13.0/24     0.0.0.0                  0         32768 ?
*>i10.1.45.0/24     4.4.4.4                  0    100      0 ?
*> 192.168.1.0      10.1.13.1                1         32768 ?
*>i192.168.5.0      4.4.4.4                  1    100      0 ?

R3#Sh ip bgp vpnv4 vrf CA 1.0.0.0/8

BGP routing table entry for 1:10:1.0.0.0/8, version 29
Paths: (1 available, best #1, table CA)
  Advertised to update-groups:
     1
  Local
    10.1.13.1 from 0.0.0.0 (3.3.3.3)
      Origin incomplete, metric 1, localpref 100, weight 32768, valid,
sourced, best
      Extended Community: RT:1:100
      mpls labels in/out 65306/nolabel
```

**Note the Origin code is (incomplete) because the route was redistributed and therefore, the origin of the route is unknown. The Weight attribute is 32768, because the local router is advertising the route in BGP. The extended community is identified, this is the configured route-target, and the last line states that if the local router receives a packet with a label of 65306 it will remove the label and forward it.**

## On R4

```
RP/0/5/CPU0:R4#Sh bgp vpnv4 unicast vrf CB 1.0.0.0

BGP routing table entry for 1.0.0.0/8, Route Distinguisher: 1:20
Versions:
  Process           bRIB/RIB  SendTblVer
  Speaker                 33          33
Last Modified: Apr 15 17:20:03.621 for 00:09:38
Paths: (1 available, best #1)
  Not advertised to any peer
  Path #1: Received by speaker 0

  Not advertised to any peer
  Local
    3.3.3.3 (metric 3) from 3.3.3.3 (3.3.3.3)
      Received Label 65306
      Origin incomplete, metric 1, localpref 100, valid, internal, best,
group-best, import-candidate, imported
      Received Path ID 0, Local Path ID 1, version 33
      Extended community: RT:1:100

RP/0/5/CPU0:R4#Sh bgp vpnv4 unicast vrf CB | b Network

   Network            Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 1:20 (default for vrf CB)
*>i1.0.0.0/8          3.3.3.3                  1    100      0 ?
*> 5.0.0.0/8          10.1.45.5                1         32768 ?
*>i10.1.13.0/24       3.3.3.3                  0    100      0 ?
*> 10.1.45.0/24       0.0.0.0                  0         32768 ?
*>i192.168.1.0/24     3.3.3.3                  1    100      0 ?
*> 192.168.5.0/24     10.1.45.5                1         32768 ?

RP/0/5/CPU0:R4#Sh route vrf CB | b Gateway
Gateway of last resort is not set

B    1.0.0.0/8 [200/1] via 3.3.3.3 (nexthop in vrf default), 00:12:19
R    5.0.0.0/8 [120/1] via 10.1.45.5, 00:11:59, GigabitEthernet0/1/0/2
B    10.1.13.0/24 [200/0] via 3.3.3.3 (nexthop in vrf default), 00:21:58
C    10.1.45.0/24 is directly connected, 18:24:37, GigabitEthernet0/1/0/2
L    10.1.45.4/32 is directly connected, 18:24:37, GigabitEthernet0/1/0/2
B    192.168.1.0/24 [200/1] via 3.3.3.3 (nexthop in vrf default), 00:21:58
R    192.168.5.0/24 [120/1] via 10.1.45.5, 00:33:16, GigabitEthernet0/1/0/2

RP/0/5/CPU0:R4#Sh route vrf CB 1.0.0.0

Routing entry for 1.0.0.0/8
  Known via "bgp 65001", distance 200, metric 1, type internal
  Installed Apr 15 17:20:03.584 for 00:13:04
  Routing Descriptor Blocks
    3.3.3.3, from 3.3.3.3
      Nexthop in Vrf: "default", Table: "default", IPv4 Unicast, Table
Id: 0xe0000000
      Route metric is 1
  No advertising protos.
```

**<u>On R5</u>**

```
R5#Sh ip route rip | i R

Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       o - ODR, P - periodic downloaded static route, H - NHRP
R        1.0.0.0 [120/2] via 10.1.45.4, 00:00:22, GigabitEthernet0/0
R      192.168.1.0/24 [120/2] via 10.1.45.4, 00:00:22, GigabitEthernet0/0
```

## To test the configuration:

### On R5

```
R5#Ping 1.1.1.1 so 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 = 1/2/4 ms
```

### On R1

```
R1#Ping 5.5.5.5 so lo0

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 5.5.5.5, timeout is 2 seconds:
Packet sent with a source address of 1.1.1.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
```

## Task 7

**Stop all routers in the console window and exit. Stop the Server and press the "Erase Start" launcher before proceeding.**