CEF table is built by grabbing the prefixes from IGPs

```
show ip cef
show ip cef summary
debug ip cef table
```

Initial setup/ framework

```
ip cef/ no ip cef
mpls label protocol [ ldp I tdp I both ]
            -- set globally and run the same command on the interface configs
mpls ldp router-id loopback 0 force
            -- finally, turn on in each interface config:
mpls ip
```

```
show mpls ldp neighbor
show mpls ldp bindings [ optionally specify IP address/prefix in form x.x.x.x xx ]
debug mpls ldp peer state-machine  -- info about state transitions for LDP sessions
debug mpls ldp messages [ sent I recieved ]
show mpls interfaces
```

Specify/ limit what label bindings will be imported

```
ip access-list standard ACCEPT
  permit 56.56.56.0 0.0.0.255
----------- From these neighbors, only accept labels referred to in this access list
mpls ldp neighbor 3.3.3.3 labels accept ACCEPT
mpls ldp neighbor 4.4.4.4 labels accept ACCEPT
----- imports 5.5.5.5 and 6.6.6.6 bindings (residing in 56.56.56.0) from 3.3.3.3 and 4.4.4.4
```

MPLS Traffic Engineering for the backbone/ core routers (instead of LDP/TDP)

```
--- hardcode the RID for TE to loopback (TE RID is separate from LDP/TPD RID)
router ospf 2
 mpls traffic-eng area 0
 mpls traffic-eng router-id lo0
            --- sets it up, now turn it on:
mpls traffic-eng tunnels    --- Enable globally, then enable on ints
int e0/1
 mpls traffic-eng tunnels
 ip rsvp bandwidth 5000    --- 5Mbps
```

```
show mpls traffic-eng topology brief
show mpls traffic-eng link-management summary
debug mpls traffic-eng areas
debug mpls traffic-eng link-management events
```

TE Tunnels - unidirectional label switching paths

Tunneling traffic from R2 to R3 *through* R4

R2

```
ip explicit-path name <PATHNAME>
```

```
 next-address 24.24.24.4   --- R4 int that connects to R2
 next-address 34.34.34.3   --- R3 int that connects to R4
 next-address 3.3.3.3      --- R3's loopback int

interface tunnel 23         --- this is a UNIDIRECTIONAL tunnel!
 ip unnumber lo0
 tunnel destination 3.3.3.3
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
          --- tell IGP to use the tunnel in its SPF calculation "it isn't in OSPF but treated like it is"
 tunnel mpls traffic-eng bandwidth 2000
          --- last config RSVPd 5Mbit BW, this will take up 2Mbit of that
 tunnel mpls traffic-eng path-option 1 explicit name <PATHNAME>
!
show mpls traffic-eng tunnels
show ip ospf neighbor      --- verify OSPF is seeing tunnel companion as connected neighbor
show ip route 3.3.3.3    ---display int that connects to R3
debug mpls traffic-eng tunnels events

 ip unnumber lo0
--- enable IP processing on an inte without assigning it an explicit IP address.
--- An ip unnumbered int can "borrow" the IP of another int already configured
--- Can conserve network and address space.
```

MPLS VPN [IP VPN for MPLS allows SP to deploy scalable IPv4 L3 VPN backbone services]
BGP for path of MPLS packet forwarding/AS traversal; iBGP between PE and PE
PE router receives IP prefix from CE and add 8 bytes RD (Route Distinguisher) to the IP prefix
VRF defined VPN consists of RD and route target (RT)

Configure VRFs on R2 and R6

```
R2
show run interface e0/3
          --- note the IP address/mask first; configuring VRF will remove it from the global routing table to
put in (one of) the VRF tables.  IP is removed from interface and must be re-assigned!

ip vrf <VPN-NAME>
 rd 2.2.2.2:2
          --- route descriptor/ route-distinguisher - it is coming from R2 so it gets it's loopback
          --- added to beginning of customer's IPv4 prefixes to make globally unique VPN-IPv4 prefixes
 route-target export 2.2.2.2:2
          --- exporting the VPNv4 target address advertised out
          --- specifies VPN route-target export communities
 route-target import 6.6.6.6:6
          --- similarly imported to the VRF from BGP VPNv4 routes
          --- specifies VPN route-target import communities

          --- now, associate the interface to the VRF
interface e0/3
 ip vrf forwarding <VPN-NAME>
          --- As previously mentioned we have to re-add the IP since it will zap it
 ip add 12.12.12.2 255.255.255.0
```

R6 - generally the same as above, reversed appropriately:

```
show run interface e0/3
ip vrf  <VPN-NAME>
 rd 6.6.6.6:6
 route-target export 6.6.6.6:6
 route-target import 2.2.2.2:2
interface e0/3
 ip vrf forwarding  <VPN-NAME>
 ip address 67.67.67.6 255.255.255.0

show ip vrf <VPN-NAME>
show ip vrf interfaces
```

Getting rid of IPv4 Unicast (BGP)

```
show ip bgp sum
show run I be router bgp
```

---- check out your stuff first.   BGP adjacency went down from when we switched interface to VRF before.

```
router bgp 1
 no bgp default ipv4-unicast
```

--- this is on by default and we are turning it off.
--- doing so specific address-family IPv4 entries for BGP adjacent neighbors

PE-to-PE Routing Sessions

Set up R2-to-R6 for this through R4 (in the middle)

On both R2 and R6 with 4.4.4.4 as R4 loopback

```
router bgp 1
 address-family vpnv4
 neighbor 4.4.4.4 activate
```

--- The address-family vpnv4 command replaces the match nlri and set nlri commands
--- For all other address families, address exchange is disabled by default.
--- Neighbor goes down and comes back up when we do this since it is adding a feature to the adjacency (vpnv4)

```
show ip bgp sum
debug ip bgp vpnv4 unicast
```

--- display info related to importing BGP paths into a VRF table

```
sh run I be router bgp
```

PE to CE Routing Sessions - BGP only (no routing protocol redistribution needed)

---- not in global BGP table - is all under address-family (in a specific VRF table)
---- R1 is in AS 2 and R7 is in AS3

```
router bgp 1
 address-family ipv4 vrf <VPN-NAME>
  neighbor 12.12.12.1 remote 2
  neighbor 12.12.12.1 activate

show ip vrf <VPN-NAME>
show ip route vrf <VPN-NAME>
show ip bgp vpnv4 all
```

--- BGP route is now coming from over the VRF, and the we can see R1 over VRF
--- On R6 should have populated and show up there as well. Tie it together:

```
router bgp 1
```

```
 address-family ipv4 vrf <VPN-NAME>
  neighbor 67.67.67.7 remote 3
  neighbor 67.67.67.7 activate
            --- Same show commands should show R7 on CE
            --- show ip route and ping from each CE router.  Done.
```

PE to CE when CE is running Static, RIP or OSPF (other side of CE is BGP)

R1 of CE has this:

```
router rip
 no auto-summary
 version 2
 network 12.0.0.0
 network 11.0.0.0
router bgp 2
 no neighbor 12.12.12.2 remote-as 1   ---removed BGP neighbor to functionally disable
```

R2 at PE has this:

```
router rip
 address-family ipv4 vrf <VPN-NAME>     ----- here is how you put the VRF in
  no auto-summary
  version 2
  network 12.12.12.0
   redistribute bgp 1 metric 2         ----- even though this is inside the VRF table, it is said this lets VRF know about BGP
router bgp 1
 address-family ipv4 vrf <VPN-NAME>
  no neighbor 12.12.12.1 remote 2     ---- like before (R1) get rid of BGP neighbor on this side of VRF
  redistribute rip                    ---- tells it to refer to RIP for the routing info inside the VRF
```

<u>Same for OSPF</u>

R1

```
no router rip
router ospf 1
 router-id 1.1.1.1
 network 1.1.1.1 0.0.0.0 area 0
 network 11.11.11.11 0.0.0.0 area 0
 network 11.11.11.131 0.0.0.0 area 0
 network 12.12.12.1 0.0.0.0 area 0
```

R2

```
no router rip
router ospf 12 vrf <VPN-NAME>
            --- specify VRF table after OSPF process ID
 network 12.12.12.2 0.0.0.0 area 0
 redistribute bgp 1 subnets
            --- same as with RIP.  Note the use of "subnets" here

router bgp 1
 address-family ipv4 vrf <VPN-NAME>
  no redistribute rip
  redistribute ospf 12 vrf <VPN-NAME>
            --- with RIP we didn't have to specify process ID, but also didn't have to specify VRF
```

<u>Same for Static</u>

```
R1
 no router ospf 1
ip route 0.0.0.0 0.0.0.0 12.12.12.2

R2
no router ospf 12 vrf <VPN-NAME>
ip route vrf VPN-A 11.0.0.0 255.0.0.0 12.12.12.1    --- how to add a static route in a VRF
router bgp 1
 address-family ipv4 vrf <VPN-NAME>
  no redistribute ospf 12 vrf <VPN-NAME>
  redistribute static      <------------------
```

InterAS VPN - Setting up the tunnels on the CEs

First clean up old clutter

```
R1
 no ip route 0.0.0.0 0.0.0.0 12.12.12.2
router bgp 2
 no network 11.11.11.0 mask 255.255.255.128
 no network 11.11.11.128 mask 255.255.255.128

R2
no ip route vrf VPN-A 11.0.0.0 255.0.0.0 12.12.12.1
interface Ethernet0/3
 no ip vrf forwarding VPN-A
 ip address 12.12.12.2 255.255.255.0
router bgp 1
 address-family ipv4 vrf VPN-A
  no redistribute static
```

--- Time for setup!

```
R1 in AS2
router bgp 2
 neighbor 12.12.12.2 remote-as 1 <------------R2 is in AS 1
 no bgp default route-target filter
            --- overrides default behavior for whether or not you have the prefix
            --- "because you may or may not have the VRF on your router"
 address-family vpnv4
 neighbor 12.12.12.2 activate    <---- activating peer (R2) to exchange routes

R2 in AS1
router bgp 1
 no bgp default route-target filter     <---- just like before
 neighbor 12.12.12.1 remote-as 2   <------------R1 in AS2
 !
 address-family vpnv4
 neighbor 12.12.12.1 activate
 neighbor 4.4.4.4 next-hop-self
```

----- If you need to pass on those VPNv4 routes that you have exchanged over eBGP, you need to set next-hop-self for VPNv4 addresses

```
show ip bgp vpnv4 all
```

------------In this one, the video ran the above command on R2 and it did not reveal anything the narration said it did.