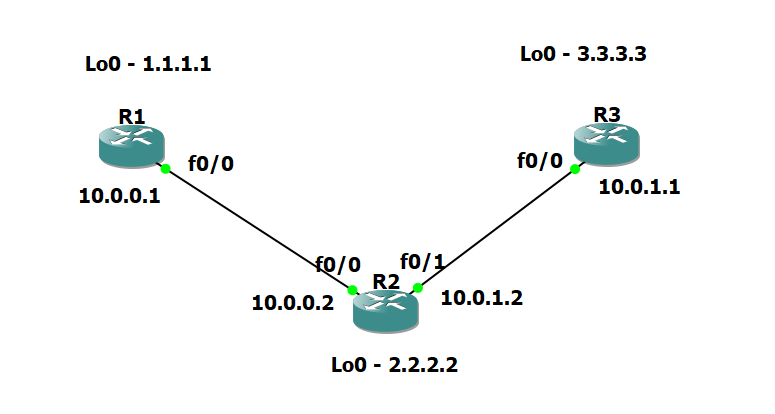
**Practical 8**

**Aim:- Simulating MPLS environment**

**Topology:-**



**Step 1 – IP addressing of MPLS Core and OSPF**

**R1 Console**

conf t

hostname R1

int lo0

ip add 1.1.1.1 255.255.255.255

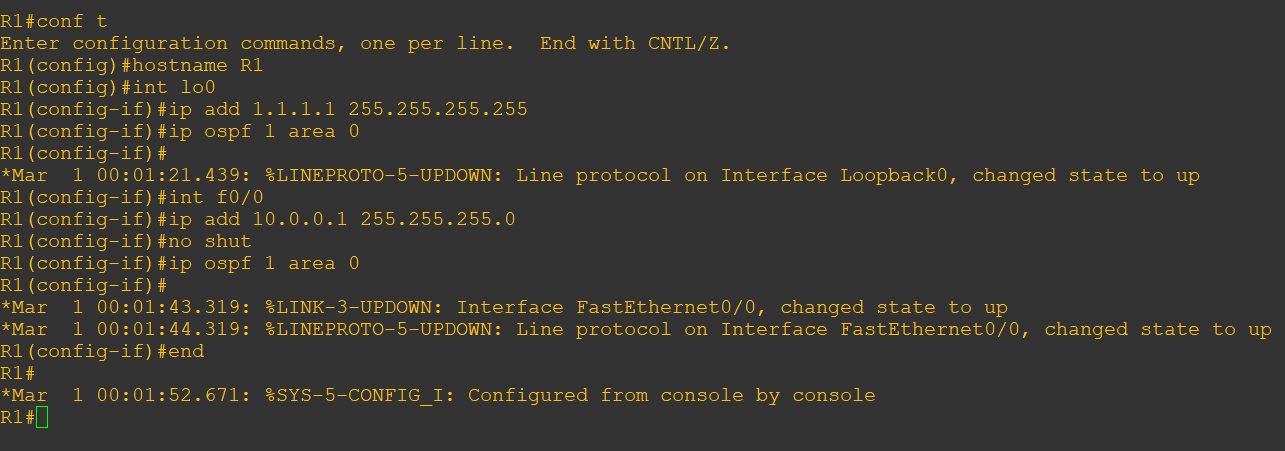
ip ospf 1 area 0

int f0/0

ip add 10.0.0.1 255.255.255.0

no shut

ip ospf 1 area 0



**R2 Console**

conf t

hostname R2

int lo0

ip add 2.2.2.2 255.255.255.255

ip ospf 1 are 0

int f0/0

ip add 10.0.0.2 255.255.255.0

no shut

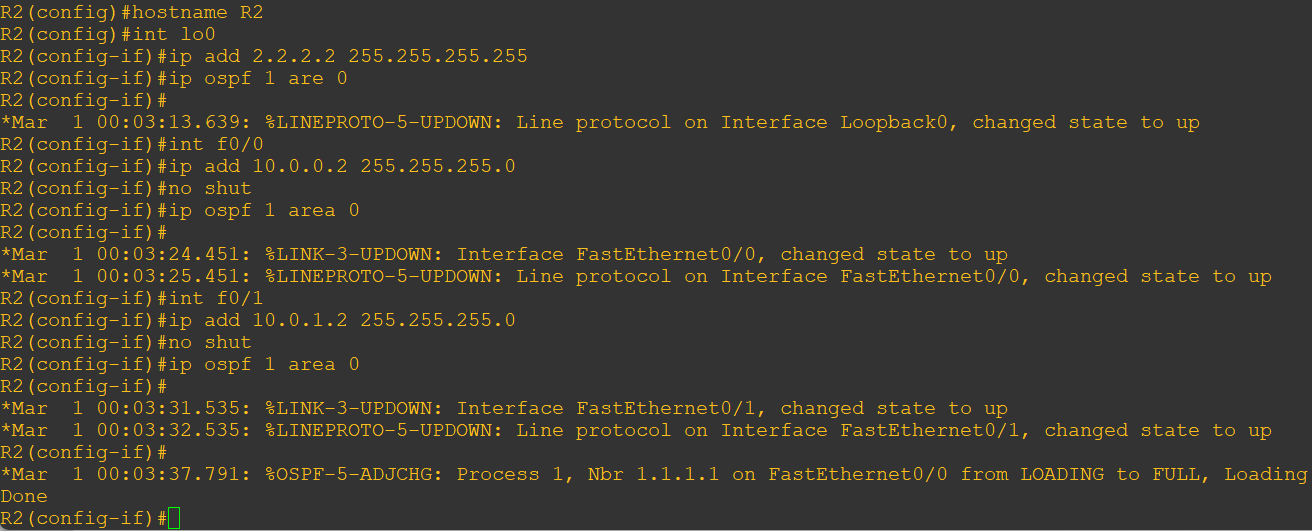
ip ospf 1 area 0

int f0/1

ip add 10.0.1.2 255.255.255.0

no shut

ip ospf 1 area 0



**R3 Console**

conf t

hostname R3

int lo0

ip add 3.3.3.3 255.255.255.255

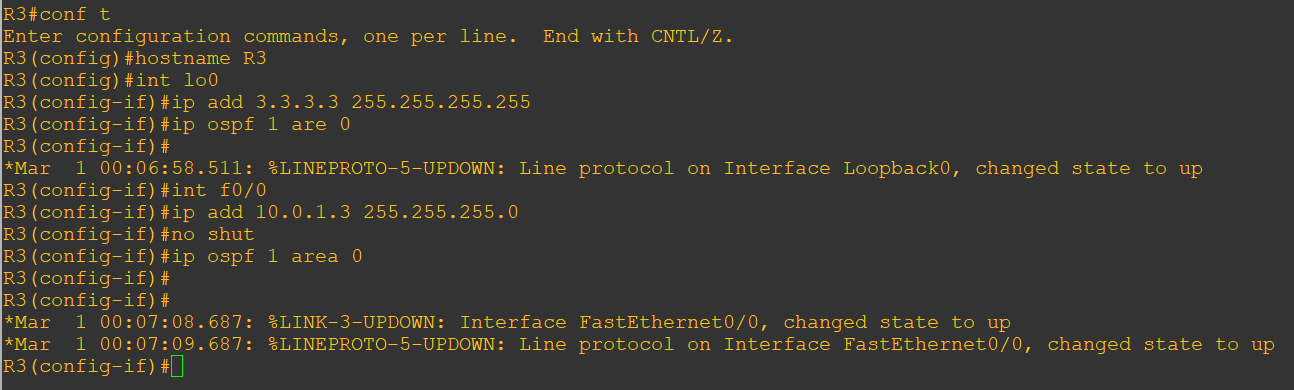
ip ospf 1 are 0

int f0/0

ip add 10.0.1.3 255.255.255.0

no shut

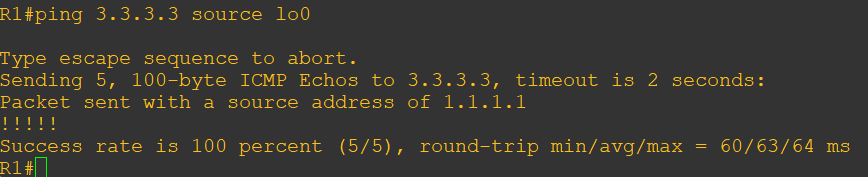
ip ospf 1 area 0



**R1 Console**

Use **ping 3.3.3.3 source lo0** command to check full ip connectivity between R1, R2, R3 to verify if we can ping between the loopbacks of R1 and R3

ping 3.3.3.3 source lo0



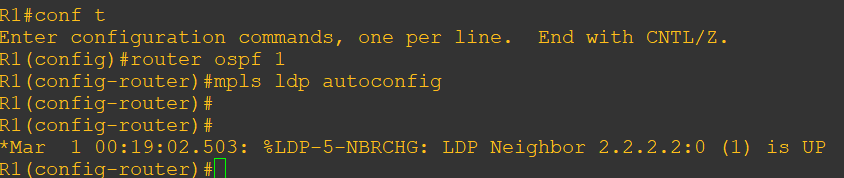
**Step 2 – Configure LDP on all the interfaces in the MPLS Core**

Under the ospf process use the **mpls ldp autoconfig** command this will enable mpls label distribution protocol on every interface running ospf under that specific process.

**R1 Console**

router ospf 1

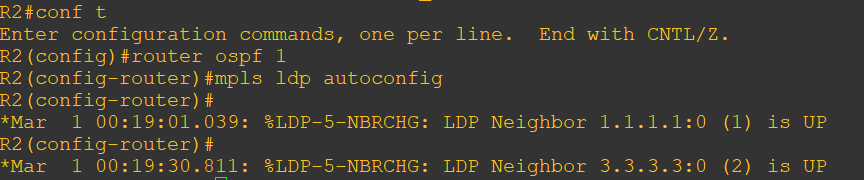
mpls ldp autoconfig



**R2 Console**

router ospf 1

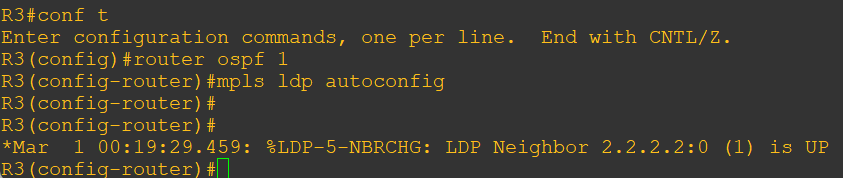
mpls ldp autoconfig



**R3 Console**

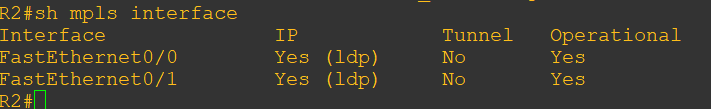
router ospf 1

mpls ldp autoconfig

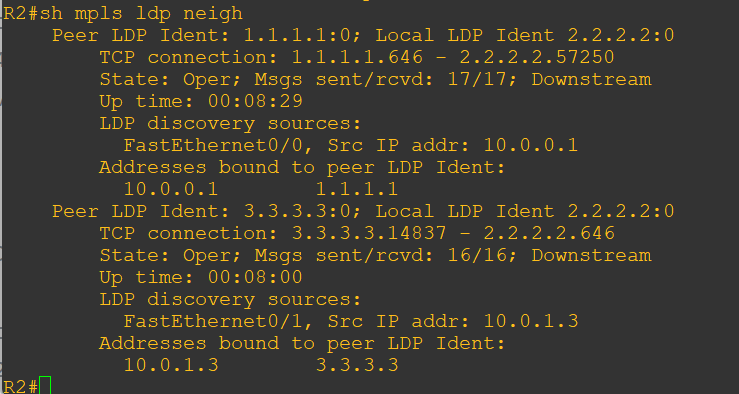


**R2 Console**

verify the mpls interfaces with **sh mpls interface** command.



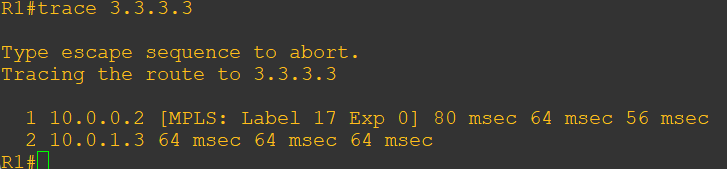
verify the LDP neighbors with the **sh mpls ldp neighbor** command.



One more verification to confirm LDP is running ok is to do a **trace** between R1 and R3 and verify if we get MPLS Labels show up in the trace.

**R1 Console**

trace 3.3.3.3



**Step 3 – MPLS BGP Configuration between R1 and R3**

Establish a Multi Protocol BGP session between R1 and R3 this is done by configuring the vpnv4 address family as below

**R1 Console**

conf t

router bgp 1

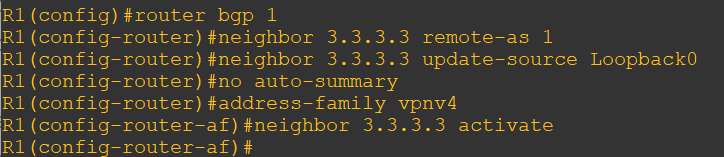
neighbor 3.3.3.3 remote-as 1

neighbor 3.3.3.3 update-source Loopback0

no auto-summary

address-family vpnv4

neighbor 3.3.3.3 activate



**R3 Console**

conf t

router bgp 1

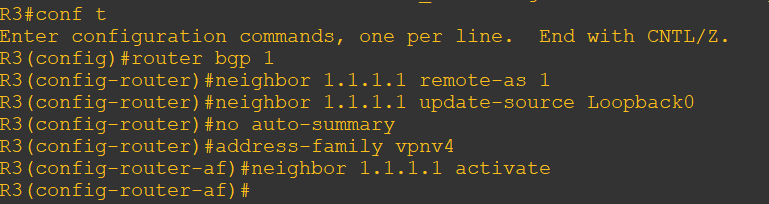
neighbor 1.1.1.1 remote-as 1

neighbor 1.1.1.1 update-source Loopback0

no auto-summary

address-family vpnv4

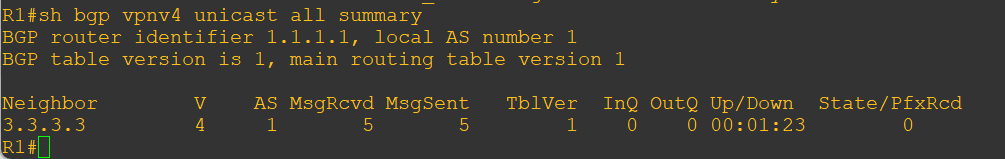
neighbor 1.1.1.1 activate



To verify the BGP session between R1 and R3 issue the command **sh bgp vpnv4** **unicast all summary**

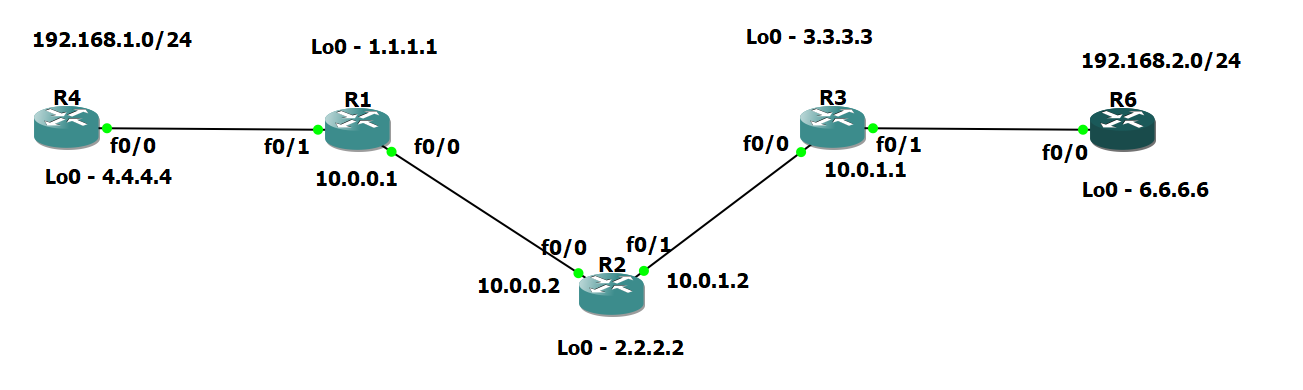
**R1 Console**

sh bgp vpnv4 unicast all summary



**Step 4 – Add two more routers, create VRFs**

Add two more routers into the topology.



**R4 Console**

int lo0

ip add 4.4.4.4 255.255.255.255

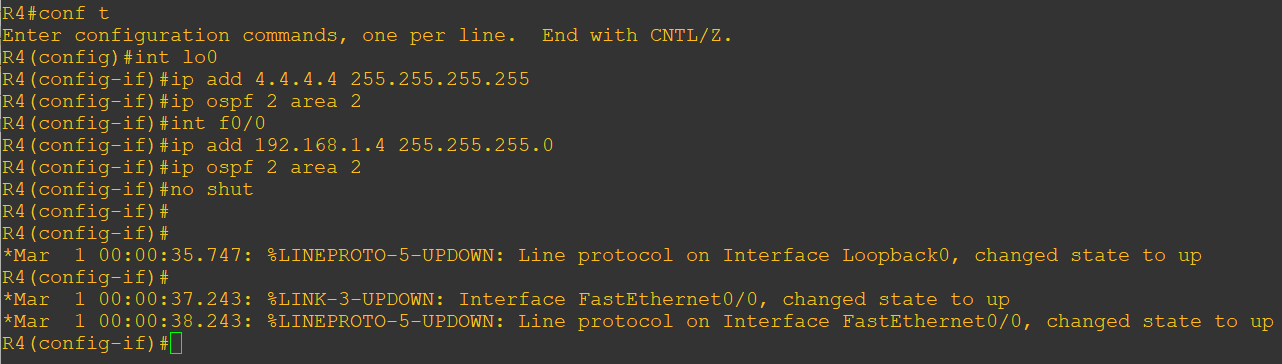
ip ospf 2 area 2

int f0/0

ip add 192.168.1.4 255.255.255.0

ip ospf 2 area 2

no shut

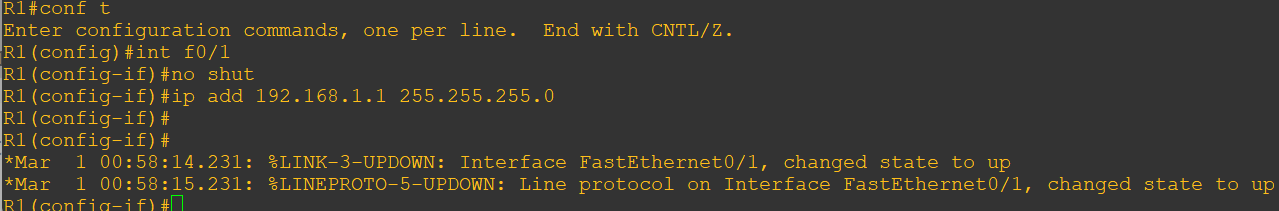


**R1 Console**

int f0/1

no shut

ip add 192.168.1.1 255.255.255.0



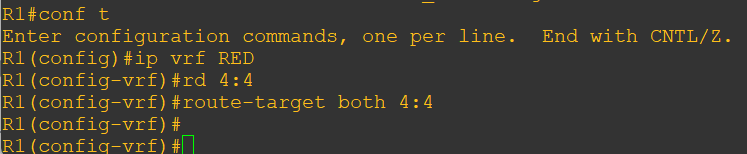
Now create VRF on R1

**R1 Console**

ip vrf RED

rd 4:4

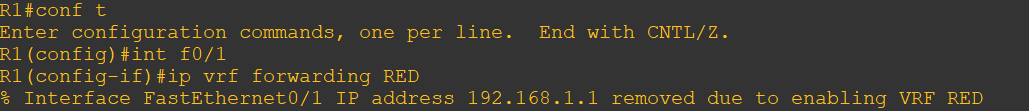
route-target both 4:4



So now we have configured the VRF on R1 we need to move the interface F0/1 into that VRF

int f0/1

ip vrf forwarding RED

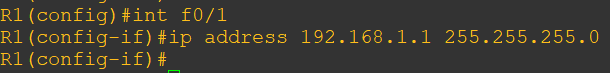


Just reapply below

**R1 Console**

int f0/1

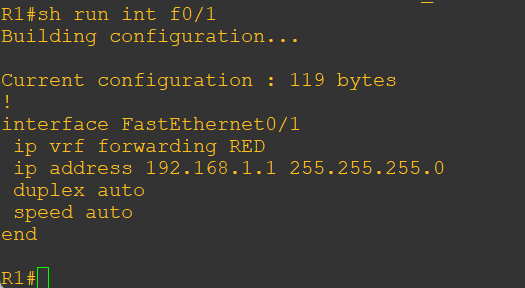
ip address 192.168.1.1 255.255.255.0



Now view the config on R1 int f0/1 we can see the VRF configured

**R1 Console**

sh run int f0/1



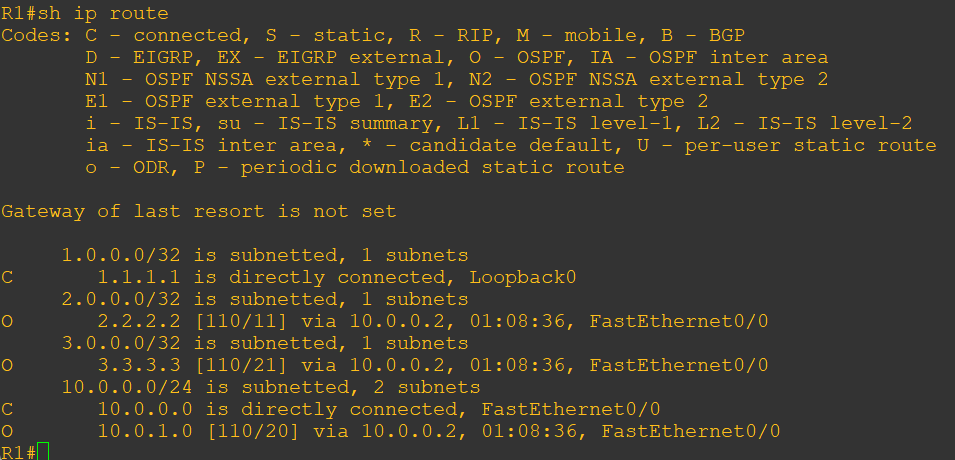
Now that there are 2 routing tables within R1

* The Global Routing Table
* The Routing Table for VRF RED

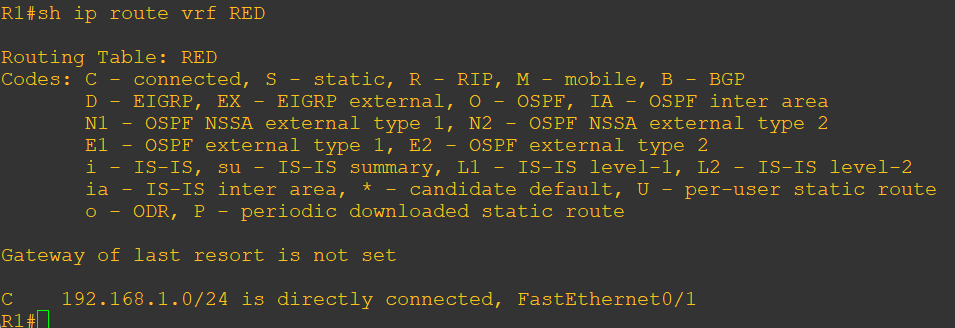
Issue the command sh ip route this shows the routes in the global table and you will notice that you do not see 192.168.1.0/24

**R1 Console**

sh ip route



Issue the command **sh ip route vrf red** – this will show the routes in the routing table for VRF RED

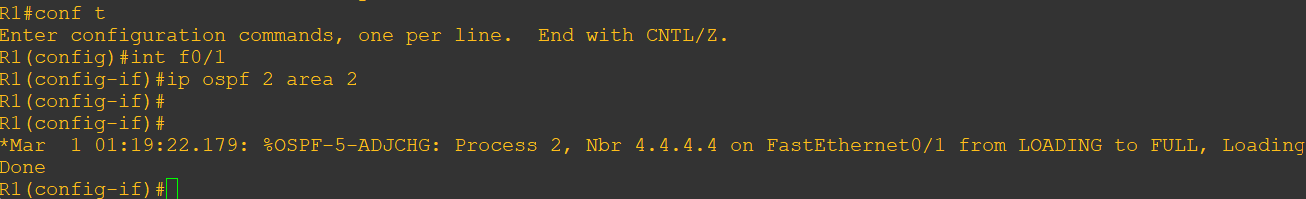


Enable OSPF on this interface and get the loopback address for R4 in the VRF RED routing table.

**R1 Console**

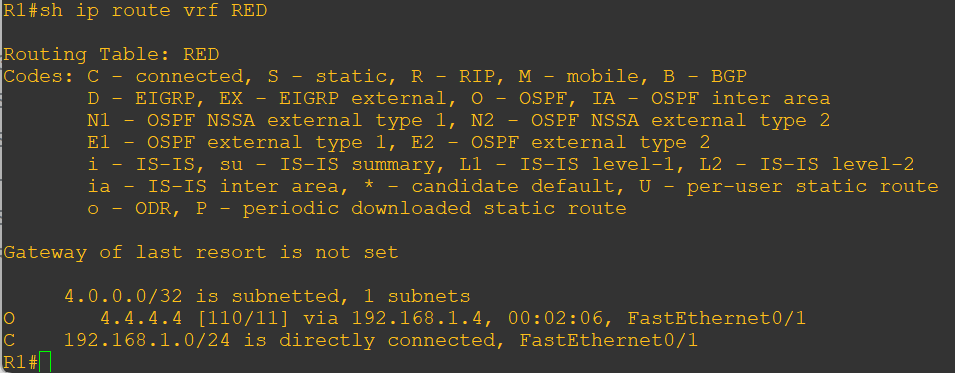
int f0/1

ip ospf 2 area 2



Check the routes in the VRF RED routing table we should see 4.4.4.4 in there as well.

sh ip route vrf RED



Repeat this process for R3 and R6 as below:

**R6 Console**

int lo0

ip add 6.6.6.6 255.255.255.255

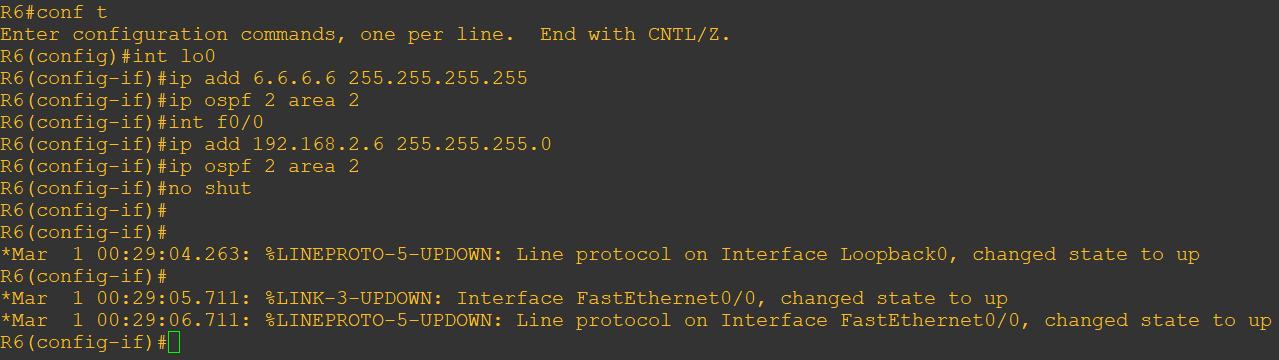
ip ospf 2 area 2

int f0/0

ip add 192.168.2.6 255.255.255.0

ip ospf 2 area 2

no shut

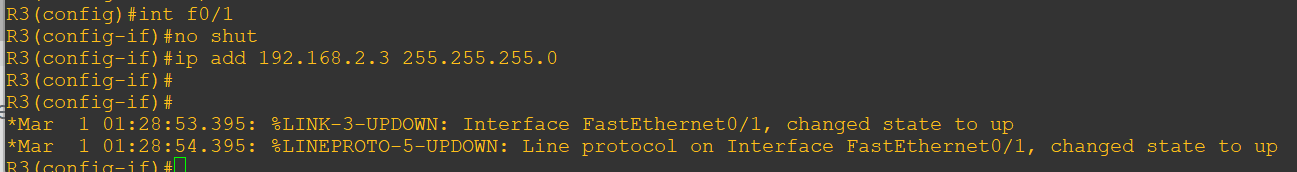


**R3 Console**

int f0/1

no shut

ip add 192.168.2.3 255.255.255.0



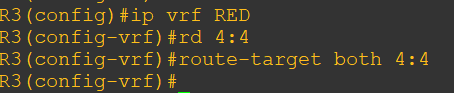
create VRF on R3

**R3 Console**

ip vrf RED

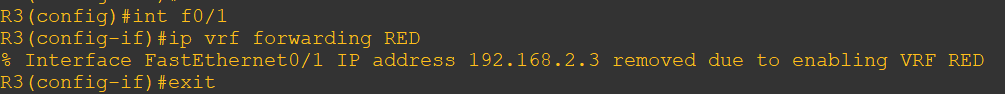
rd 4:4

route-target both 4:4



int f0/1

ip vrf forwarding RED

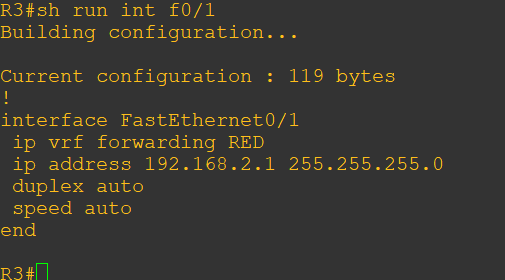


int f0/1

ip address 192.168.2.1 255.255.255.0

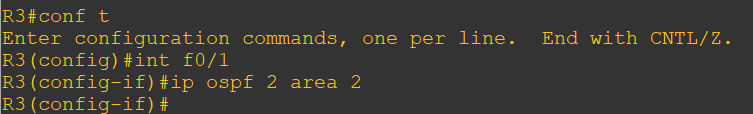


sh run int f0/1

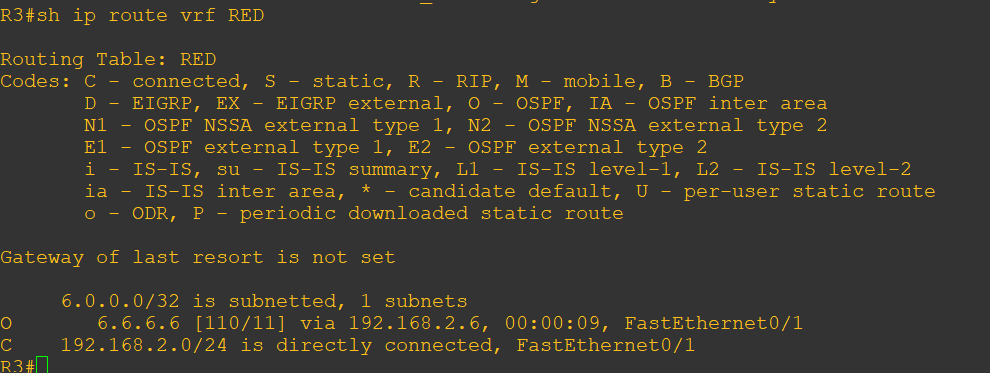


int f0/1

ip ospf 2 area 2



sh ip route vrf RED



Perform verification on R1 and R4

**R1 and R4 Console**

sh ip route

**R1 Console**

sh ip route vrf RED

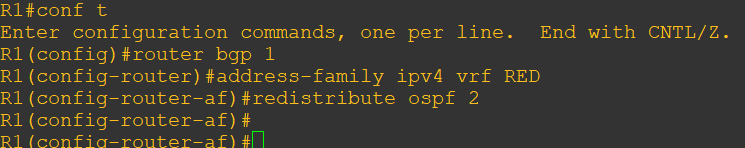
**Step - Redistribute OSPF into MP-BGP on R1**

**R1 Console**

router bgp 1

address-family ipv4 vrf RED

redistribute ospf 2



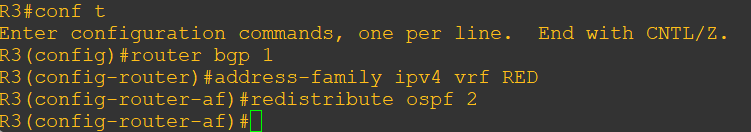
**Step- Redistribute OSPF into MP-BGP on R3**

**R3 Console**

router bgp 1

address-family ipv4 vrf RED

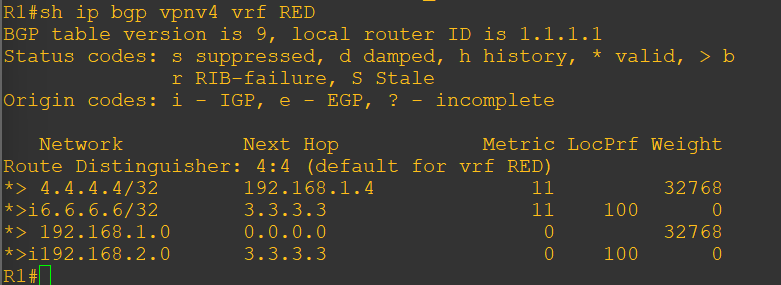
redistribute ospf 2

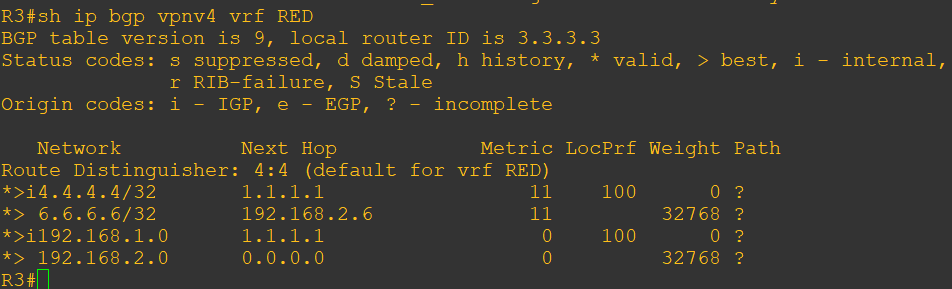


This has enabled redistribution of the OSPF routes into BGP. We can check the routes from R4 and R6 are now showing in the BGP table for their VRF with below command

**R1 and R3 Console**

sh ip bgp vpnv4 vrf RED



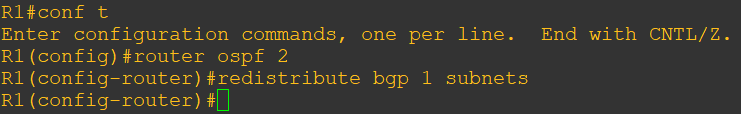


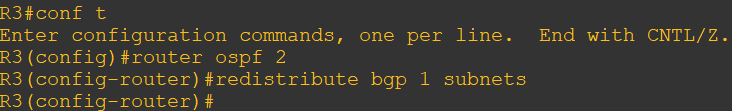
The final step is to get the routes that have come across the MPLS back into OSPF and then we can get end to end connectivity.

**R1 and R3 Console**

router ospf 2

redistribute bgp 1 subnets

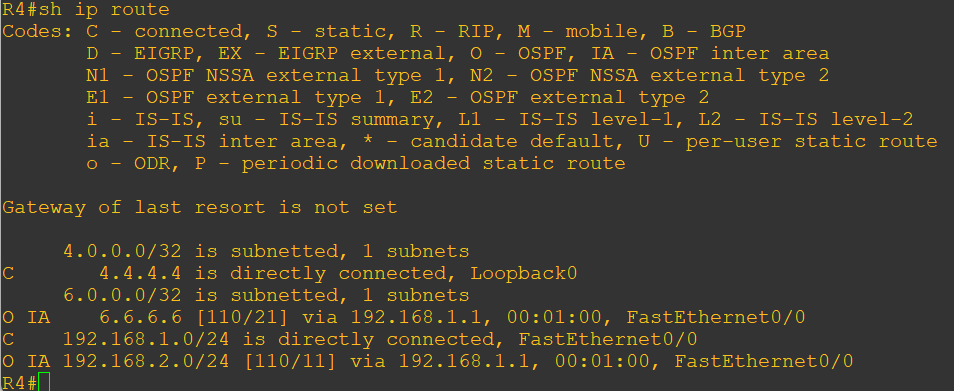


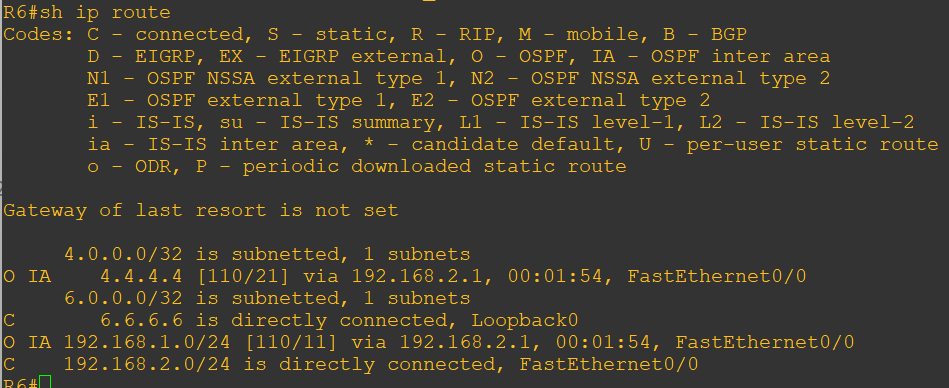


If all has worked we should be now able to ping 6.6.6.6 from R4

**R4 and R6 Console**

sh ip route

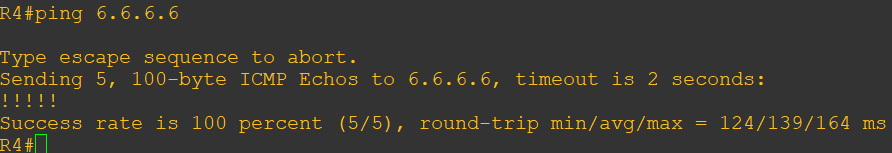




we have 4.4.4.4 in there so we should be able to ping across the MPLS

**R4 Console**

ping 6.6.6.6



Add below command in R4 and R6 Console:

router ospf 2

mpls ldp autoconfig

**R4 Console**

After that perform trace we can able to see lables

trace 6.6.6.6

