

**IBGP Implementation Documentation**

Ryan Chen

2/5/2024

**Purpose:**

The primary purpose of this lab is to introduce the uses and configuration of iBGP, while at the same time implementing previously learned protocols such as eBGP and OSPF. It also acted as a review of previously learned concepts such as setting up different AS’s and linking them via BGP.

**Background Information:**

One of the most important protocols that connects the internet is BGP (Border Gateway Protocol), which we previously covered. This protocol is primarily used to connect various different company networks in the real world, which are often running different network protocols each carrying their own attributes and varied information. In order for them to mesh into a single cohesive network, BGP is required as an intermediary routing protocol. Primarily, each of these organizations is referred to as an Autonomous System (AS) and are thus interconnected using BGP.

Previously, the version of BGP that waws implemented was eBGP, which acts as a protocol that connects different networks (AS’s) together. However, there exists another form of BGP that runs strictly within a single internal network and brings numerous different benefits with it. This variant is called iBGP and utilizes a fully meshed network design.

There are numerous reasons why iBGP is used in combination with BGP, but primarily iBGP acts as a “transit AS”. In between two eBGP links, prefixes and data must be sent through the network. Although this is possible by redistributing BGP information into an IGP protocol such as EIGRP or OSPF, it is heavily discouraged for two reasons. First, BGP attribute information is dropped when it crosses an IGP, meaning that crucial information would be lost in transit. Secondly, the actual internet routing prefix table is extremely large, consisting of nearly 500k prefixes. Redistributing such a large amount of routing information to an IGP would almost certainly cause network downtime or slowness.

iBGP is fully meshed, which differs from traditional IGPs. Routers do not have to be directly connected to be peered, and all routers are aware of each other in a topology. This allows for greater efficiency in selecting routing paths and sending packets, increasing overall performance. Furthermore, iBGP preserves attribute information, which fixes issues that IGP’s face.

Typically, iBGP must be implemented in conjunction with an IGP. In order to achieve a fully meshed topology, each router must be able to reach one another. This is only possible by utilizing a protocol like OSPF, which distributes prefix information within the network in order for the routers to be able to reach each other to distribute iBGP information.

**Lab Summary:**

In this lab, we created three different autonomous systems, with two being the endpoints and one being a transit. The primary goal is to achieve connectivity between the two endpoints by configuring iBGP in the transit AS.

To begin, we outlined this topology within Packet Tracer, where we were able to highlight the different AS’s and their respective regions. This allowed us to visualize the role that iBGP has as an intermediary protocol. eBGP connects the iBGP network to each external AS.

After this, we first implemented our transit AS by establishing OSPF adjacencies along the three routers. This was the prerequisite for configuring iBGP, as it requires a fully meshed network, meaning all routers must be able to ping each other. Then, we configured each of the two external AS’s. In order to simplify our topology, we added loopback interfaces on each of the routers and put them inside their own network. This essentially emulated a different network that would allow us to test if our lab worked. Finally, we configured eBGP to connect the link between the transit AS and our two external networks.

iBGP was configured by adding a loopback interface to each existing router inside the transit AS. These prefixes were then propagated using OSPF, and iBGP utilized them as their connection link. Then, iBGP was configured the same way as BGP would, with a few additional commands.

**Lab Commands:**

*router bgp [PROCESS-ID]*

-Initializes the BGP process

*bgp router-id [ROUTER-ID]*

-Sets BGP router-id for neighbors

*neighbor [IPv4/IPv6-ADDRESS] remote-as [AS-NUMBER]*

-Adds a BGP neighbor, with optional BGP weight value attribute

*neighbor [IP-ADDRESS] activate*

-Establishes BGP adjacency with neighbor

*neighbor [ADDRESS] update-source [LOOPBACK]*

-Core command for iBGP which establishes full mesh, must be applied to all routers in the iBGP network

*redistribute [ROUTING-PROTOCOL] [AS-NUMBER]*

-Redistribute routing protocol routes into BGP (OSPF, RIP, EIGRP)

*address-family [IPv4/IPv6]*

-Specifies IPv4 or IPv6 configurations for BGP

*network [ADDRESS]*

-Adds a network under IPv4 or IPv6

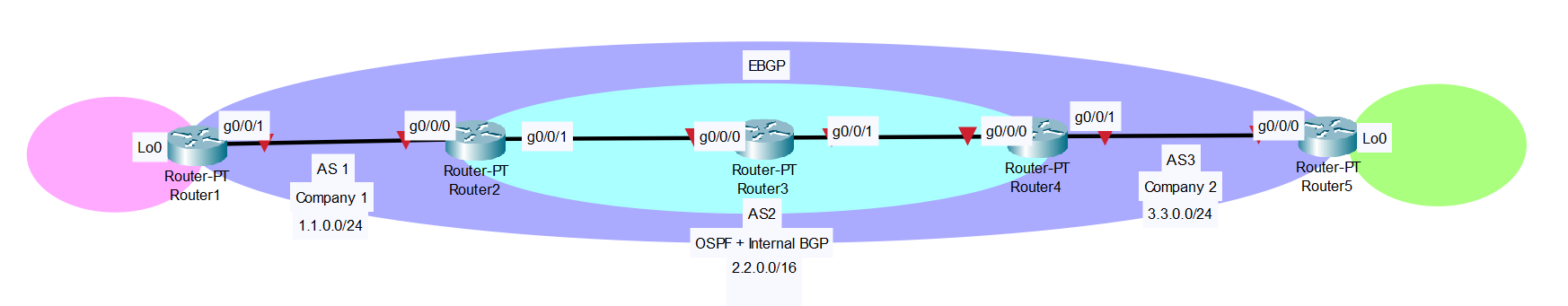
*show ip bgp*

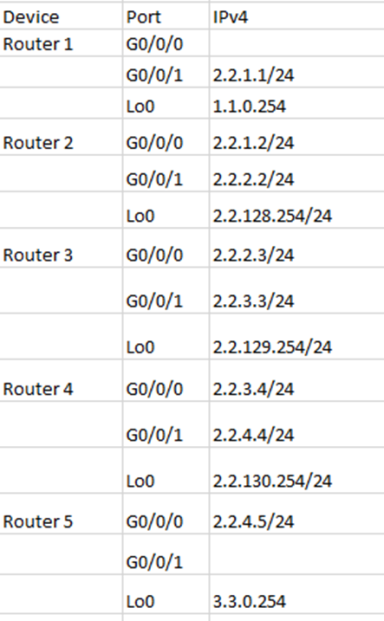
-Shows all BGP routes

*clear ip bgp \**

-Reload the BGP process and delete all stored routes, reestablish neighbors

**Network Diagram and IP Table:**

****



**Configurations:**

**R1:**

hostname R1

boot-start-marker

boot-end-marker

vrf definition Mgmt-intf

address-family ipv4

exit-address-family

address-family ipv6

exit-address-family

no aaa new-model

subscriber templating

multilink bundle-name authenticated

license udi pid ISR4321/K9 sn FDO21491LXV

license accept end user agreement

license boot level securityk9

spanning-tree extend system-id

redundancy

mode none

vlan internal allocation policy ascending

interface Loopback0

ip address 1.1.0.254 255.255.255.0

interface GigabitEthernet0/0/0

no ip address

shutdown

negotiation auto

interface GigabitEthernet0/0/1

ip address 2.2.1.1 255.255.255.0

negotiation auto

interface Serial0/1/0

interface Serial0/1/1

interface GigabitEthernet0

vrf forwarding Mgmt-intf

no ip address

shutdown

negotiation auto

interface Vlan1

no ip address

shutdown

router bgp 1

bgp router-id 1.1.1.1

bgp log-neighbor-changes

network 1.1.0.0 mask 255.255.255.0

neighbor 2.2.1.2 remote-as 2

ip forward-protocol nd

no ip http server

no ip http secure-server

ip tftp source-interface GigabitEthernet0

control-plane

line con 0

stopbits 1

line aux 0

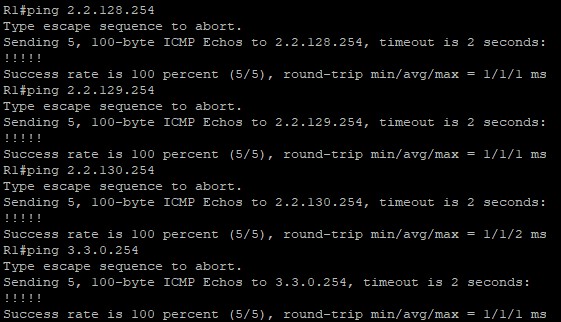
stopbits 1

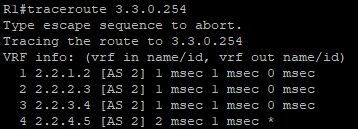
line vty 0 4

login

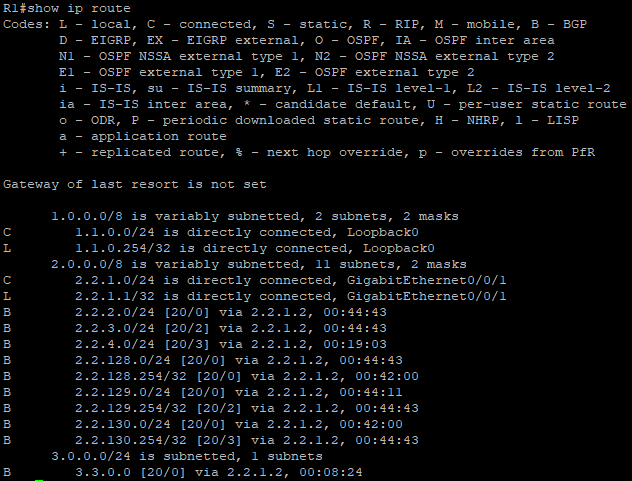
end

**Pings and Traceroute:**

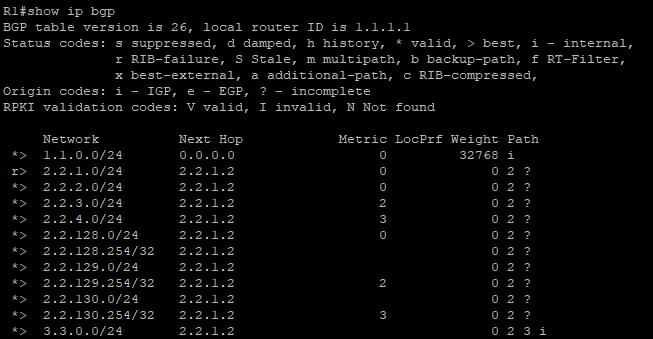


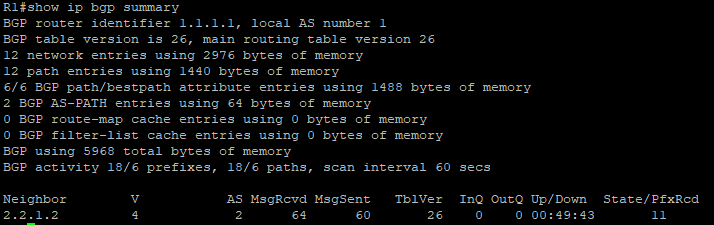


**Routing Table:**



**BGP Info:**





**R2:**

hostname R2

boot-start-marker

boot-end-marker

vrf definition Mgmt-intf

address-family ipv4

exit-address-family

address-family ipv6

exit-address-family

no aaa new-model

subscriber templating

multilink bundle-name authenticated

license udi pid ISR4321/K9 sn FDO214420QQ

license accept end user agreement

license boot level securityk9

spanning-tree extend system-id

redundancy

mode none

vlan internal allocation policy ascending

interface Loopback0

ip address 2.2.128.254 255.255.255.0

interface GigabitEthernet0/0/0

ip address 2.2.1.2 255.255.255.0

negotiation auto

interface GigabitEthernet0/0/1

ip address 2.2.2.2 255.255.255.0

negotiation auto

interface Serial0/1/0

no ip address

shutdown

interface Serial0/1/1

no ip address

shutdown

interface GigabitEthernet0

vrf forwarding Mgmt-intf

no ip address

shutdown

negotiation auto

interface Vlan1

no ip address

shutdown

router ospf 1

router-id 2.2.2.2

redistribute bgp 2 subnets

network 2.2.1.0 0.0.0.255 area 0

network 2.2.2.0 0.0.0.255 area 0

network 2.2.128.0 0.0.0.255 area 0

router bgp 2

bgp router-id 2.2.2.2

bgp log-neighbor-changes

neighbor 2.2.1.1 remote-as 1

neighbor 2.2.2.3 remote-as 2

neighbor 2.2.2.3 update-source Loopback0

neighbor 2.2.3.4 remote-as 2

neighbor 2.2.3.4 update-source Loopback0

address-family ipv4

network 2.2.1.0

network 2.2.2.0

network 2.2.128.0

redistribute ospf 1

neighbor 2.2.1.1 activate

neighbor 2.2.2.3 activate

neighbor 2.2.3.4 activate

exit-address-family

ip forward-protocol nd

no ip http server

no ip http secure-server

ip tftp source-interface GigabitEthernet0

control-plane

line con 0

stopbits 1

line aux 0

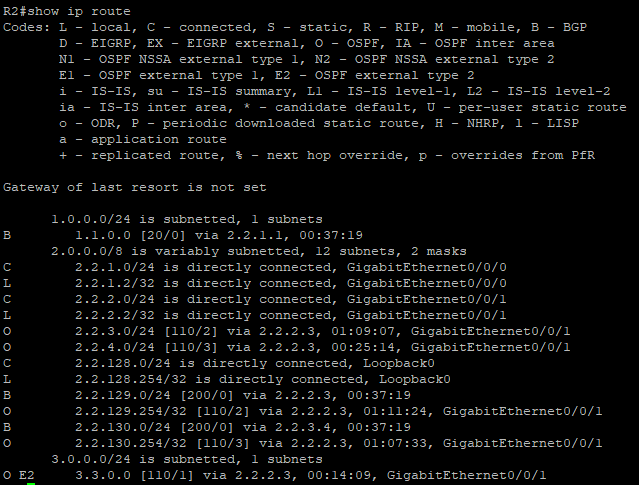
stopbits 1

line vty 0 4

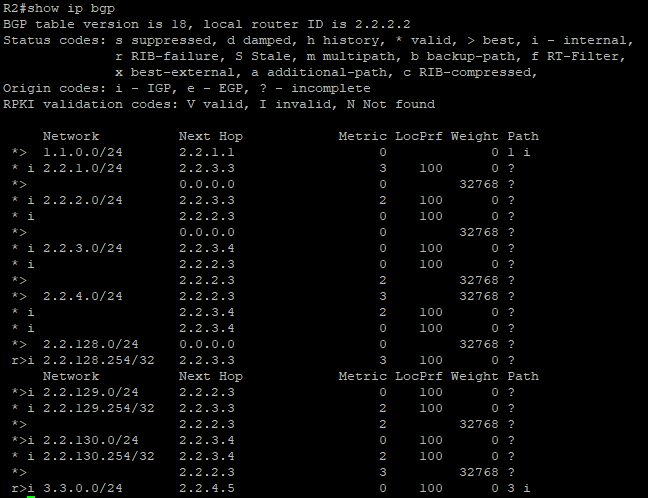
login

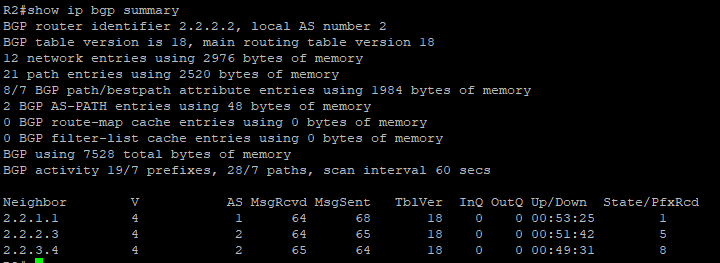
end

**Routing Table:**



**BGP Info:**





**R3:**

hostname R3

boot-start-marker

boot-end-marker

vrf definition Mgmt-intf

address-family ipv4

exit-address-family

address-family ipv6

exit-address-family

no aaa new-model

subscriber templating

vtp domain cisco

vtp mode transparent

multilink bundle-name authenticated

license udi pid ISR4321/K9 sn FDO214420HY

license boot level securityk9

spanning-tree extend system-id

redundancy

mode none

vlan internal allocation policy ascending

vlan 10,20

interface Loopback0

ip address 2.2.129.254 255.255.255.0

interface GigabitEthernet0/0/0

ip address 2.2.2.3 255.255.255.0

negotiation auto

interface GigabitEthernet0/0/1

ip address 2.2.3.3 255.255.255.0

negotiation auto

interface Serial0/1/0

no ip address

interface Serial0/1/1

no ip address

interface GigabitEthernet0

vrf forwarding Mgmt-intf

no ip address

negotiation auto

interface Vlan1

no ip address

router ospf 1

router-id 3.3.3.3

redistribute bgp 2 subnets

network 2.2.2.0 0.0.0.255 area 0

network 2.2.3.0 0.0.0.255 area 0

network 2.2.129.0 0.0.0.255 area 0

router bgp 2

bgp router-id 3.3.3.3

bgp log-neighbor-changes

neighbor 2.2.2.2 remote-as 2

neighbor 2.2.3.4 remote-as 2

address-family ipv4

network 2.2.2.0

network 2.2.3.0

network 2.2.129.0

redistribute ospf 1

neighbor 2.2.2.2 activate

neighbor 2.2.3.4 activate

exit-address-family

ip forward-protocol nd

no ip http server

no ip http secure-server

control-plane

line con 0

stopbits 1

line aux 0

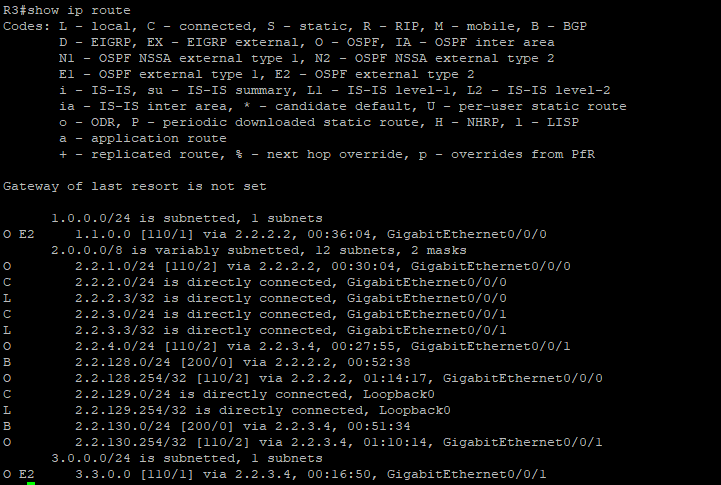
stopbits 1

line vty 0 4

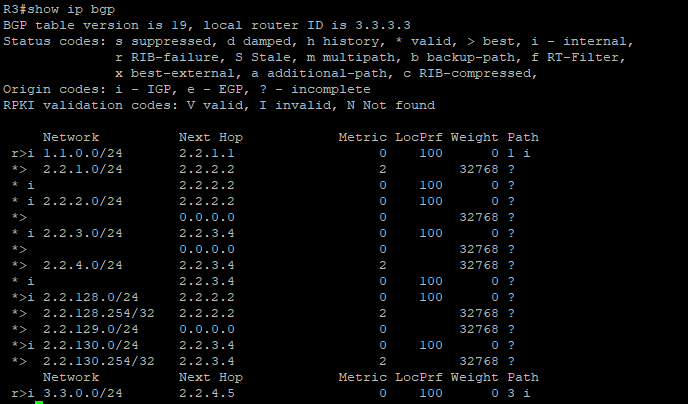
login

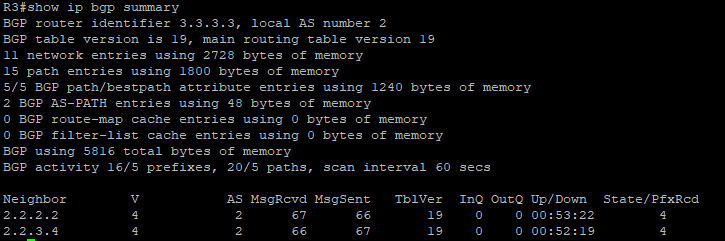
end

**Routing Table:**



**BGP Info:**





**R4:**

hostname R4

boot-start-marker

boot-end-marker

vrf definition Mgmt-intf

address-family ipv4

exit-address-family

address-family ipv6

exit-address-family

no aaa new-model

subscriber templating

vtp domain cisco

vtp mode transparent

multilink bundle-name authenticated

license udi pid ISR4321/K9 sn FDO214333H6

license boot level securityk9

spanning-tree extend system-id

redundancy

mode none

vlan internal allocation policy ascending

vlan 10,20

interface Loopback0

ip address 2.2.130.254 255.255.255.0

interface GigabitEthernet0/0/0

ip address 2.2.3.4 255.255.255.0

negotiation auto

interface GigabitEthernet0/0/1

ip address 2.2.4.4 255.255.255.0

negotiation auto

interface Serial0/1/0

no ip address

shutdown

interface Serial0/1/1

no ip address

shutdown

interface GigabitEthernet0

vrf forwarding Mgmt-intf

no ip address

shutdown

negotiation auto

interface Vlan1

no ip address

shutdown

router ospf 1

router-id 4.4.4.4

redistribute bgp 2 subnets

network 2.2.3.0 0.0.0.255 area 0

network 2.2.4.0 0.0.0.255 area 0

network 2.2.130.0 0.0.0.255 area 0

router ospf 2

router bgp 2

bgp router-id 4.4.4.4

bgp log-neighbor-changes

network 2.2.3.0

network 2.2.4.0

network 2.2.130.0

redistribute ospf 1

neighbor 2.2.2.2 remote-as 2

neighbor 2.2.2.2 update-source Loopback0

neighbor 2.2.3.3 remote-as 2

neighbor 2.2.3.3 update-source Loopback0

neighbor 2.2.4.5 remote-as 3

ip forward-protocol nd

no ip http server

no ip http secure-server

ip tftp source-interface GigabitEthernet0

control-plane

line con 0

stopbits 1

line aux 0

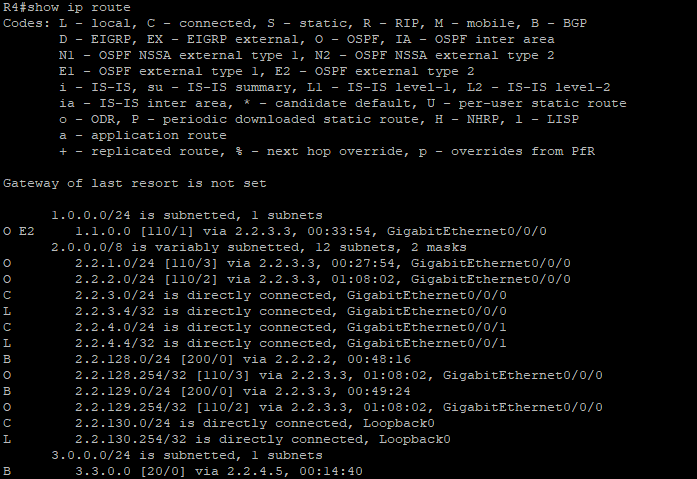
stopbits 1

line vty 0 4

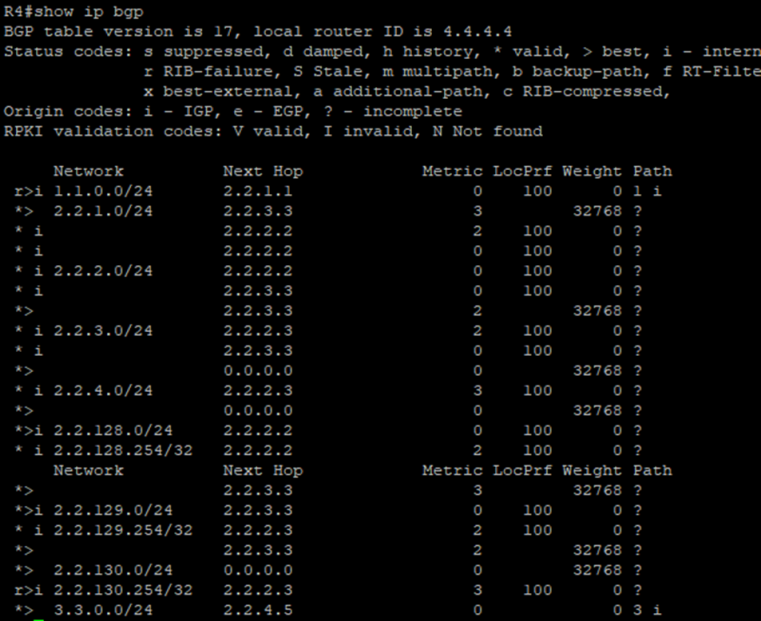
login

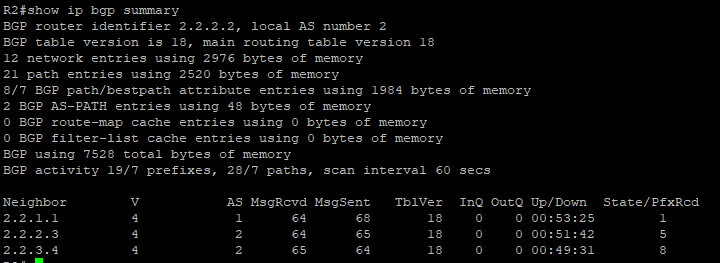
end

**Routing Table:**



**BGP Info:**



****

**R5:**

hostname R5

boot-start-marker

boot-end-marker

vrf definition Mgmt-intf

address-family ipv4

exit-address-family

address-family ipv6

exit-address-family

no aaa new-model

subscriber templating

vtp domain cisco

vtp mode transparent

multilink bundle-name authenticated

license udi pid ISR4321/K9 sn FDO21482HYV

spanning-tree extend system-id

redundancy

mode none

vlan internal allocation policy ascending

vlan 2,10,20

vlan 996

name CUSTOMER\_NATIVE

interface Loopback0

ip address 3.3.0.254 255.255.255.0

interface GigabitEthernet0/0/0

ip address 2.2.4.5 255.255.255.0

negotiation auto

interface GigabitEthernet0/0/1

no ip address

shutdown

negotiation auto

interface Serial0/1/0

no ip address

shutdown

interface Serial0/1/1

no ip address

shutdown

interface GigabitEthernet0/2/0

no ip address

shutdown

negotiation auto

interface GigabitEthernet0/2/1

no ip address

shutdown

negotiation auto

interface GigabitEthernet0

vrf forwarding Mgmt-intf

no ip address

shutdown

negotiation auto

interface Vlan1

no ip address

shutdown

router bgp 3

bgp router-id 5.5.5.5

bgp log-neighbor-changes

network 3.3.0.0 mask 255.255.255.0

neighbor 2.2.4.4 remote-as 2

ip forward-protocol nd

no ip http server

no ip http secure-server

ip tftp source-interface GigabitEthernet0

control-plane

line con 0

stopbits 1

line aux 0

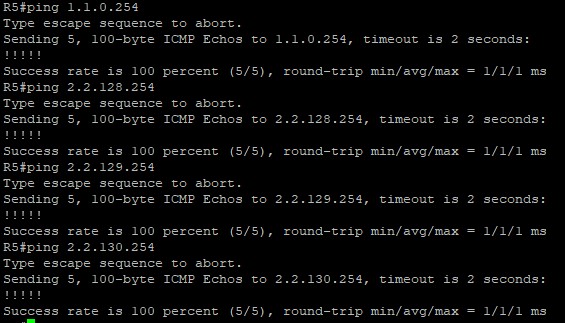
stopbits 1

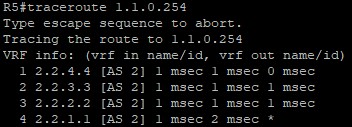
line vty 0 4

login

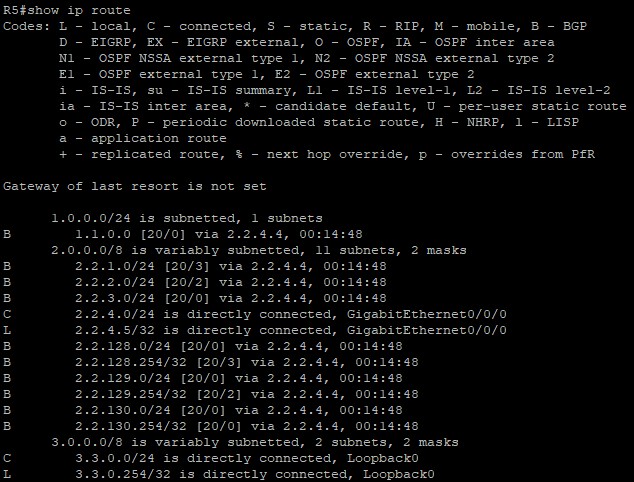
end

**Pings and Traceroute:**

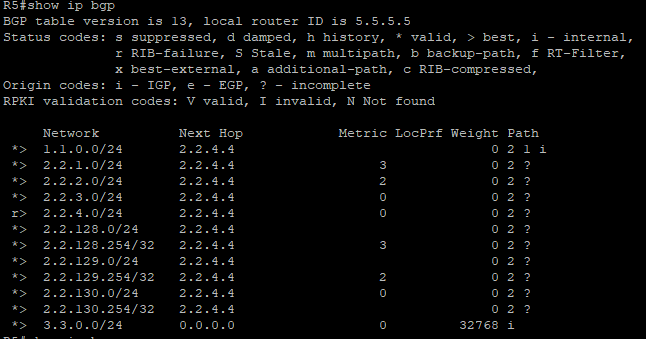
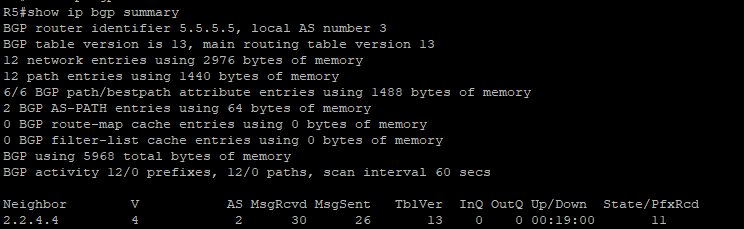




**Routing Table:**



**BGP Info:**

**Problems:**

1. Fully meshed network

eBGP does not require that all routers in a network have to be fully meshed, meaning each router only needs an adjacency with its directly connected neighbor. However, iBGP requires this, and we originally forgot to set this. Later on, we realized that pings were unable to go through and after adding neighbors with all routers in the network, we were able to fix it.

1. OSPF network setup

Originally, we forgot to distribute some of the OSPF networks in our topology, meaning that they were also not distributed into BGP. As a result, we were unable to ping across our topology. After adding these network statements into OSPF, it was fixed.

1. OSPF routes taking priority over BGP

Due to incorrect configurations with network statements and redistributing routes, our OSPF routes initially had more precedence over BGP, meaning that we were unable to demonstrate that iBGP had an effect.

1. Redid our entire lab

Since there were many miscellaneous problems in our config, we redid our lab and realized that we forgot to redistribute our iBGP into OSPF. After fixing this, our entire worked easily.

**Conclusion:**

Overall, this lab served as a useful introduction to the configuration and uses of iBGP, which we were able to combine with eBGP in order to create a fully functioning network. Furthermore, we were able to review previous concepts such as network subnetting and IGP protocol configuration using OSPF.

IBGP Signoff Sheet

Ryan Chen, P3-4 Cisco CCNP, Mr. Mason

