Ashwin, Sneha, Nufais, Logan & Arjun

CSCI 5380 - Network Virtualization and Orchestration

Lab 8

BGP SDN

University of Colorado Boulder

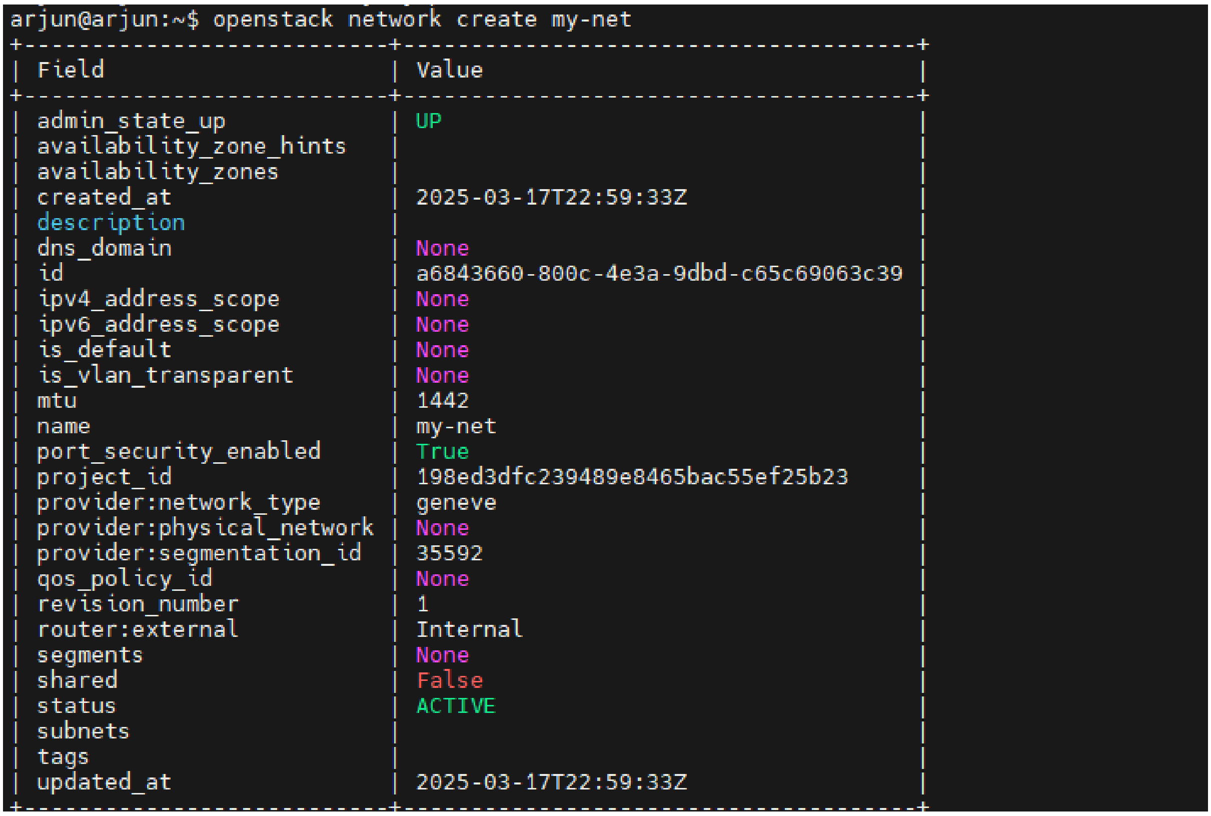
Department of Computer Science

Network Engineering

Professor Levi Perigo, Ph.D.

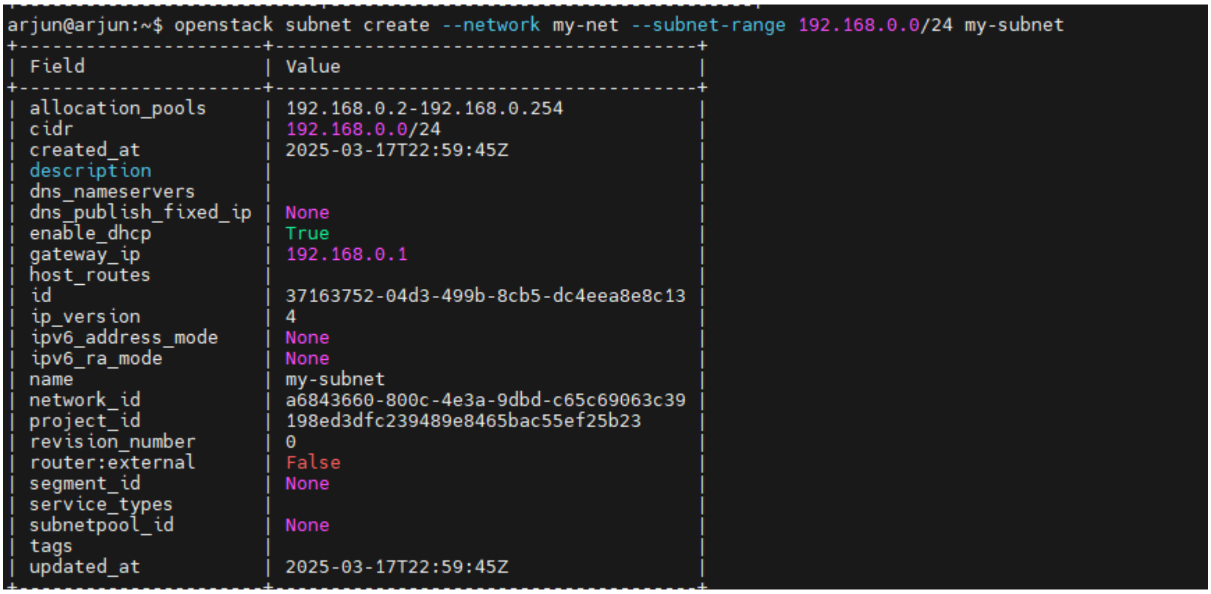
Server 1 Setup:

- Created network to connect vm



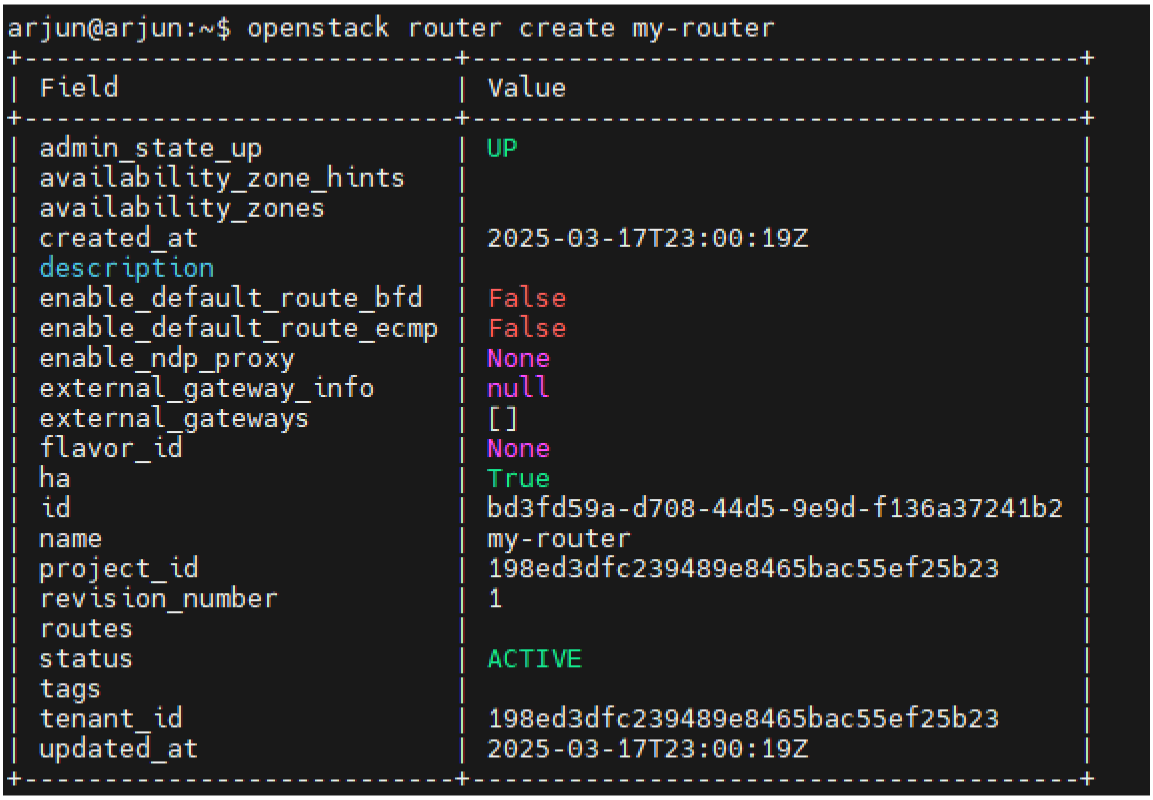
```
arjun@arjun:~$ openstack network create my-net
+---------------------------+--------------------------------------+
| Field                     | Value                                |
+---------------------------+--------------------------------------+
| admin_state_up            | UP                                   |
| availability_zone_hints   |                                      |
| availability_zones        |                                      |
| created_at                | 2025-03-17T22:59:33Z                 |
| description               |                                      |
| dns_domain                | None                                 |
| id                        | a6843660-800c-4e3a-9dbd-c65c69063c39 |
| ipv4_address_scope        | None                                 |
| ipv6_address_scope        | None                                 |
| is_default                | None                                 |
| is_vlan_transparent       | None                                 |
| mtu                       | 1442                                 |
| name                      | my-net                               |
| port_security_enabled     | True                                 |
| project_id                | 198ed3dfc239489e8465bac55ef25b23     |
| provider:network_type     | geneve                               |
| provider:physical_network | None                                 |
| provider:segmentation_id  | 35592                                |
| qos_policy_id             | None                                 |
| revision_number           | 1                                    |
| router:external           | Internal                             |
| segments                  | None                                 |
| shared                    | False                                |
| status                    | ACTIVE                               |
| subnets                   |                                      |
| tags                      |                                      |
| updated_at                | 2025-03-17T22:59:33Z                 |
+---------------------------+--------------------------------------+
```

- Adding subnet to that



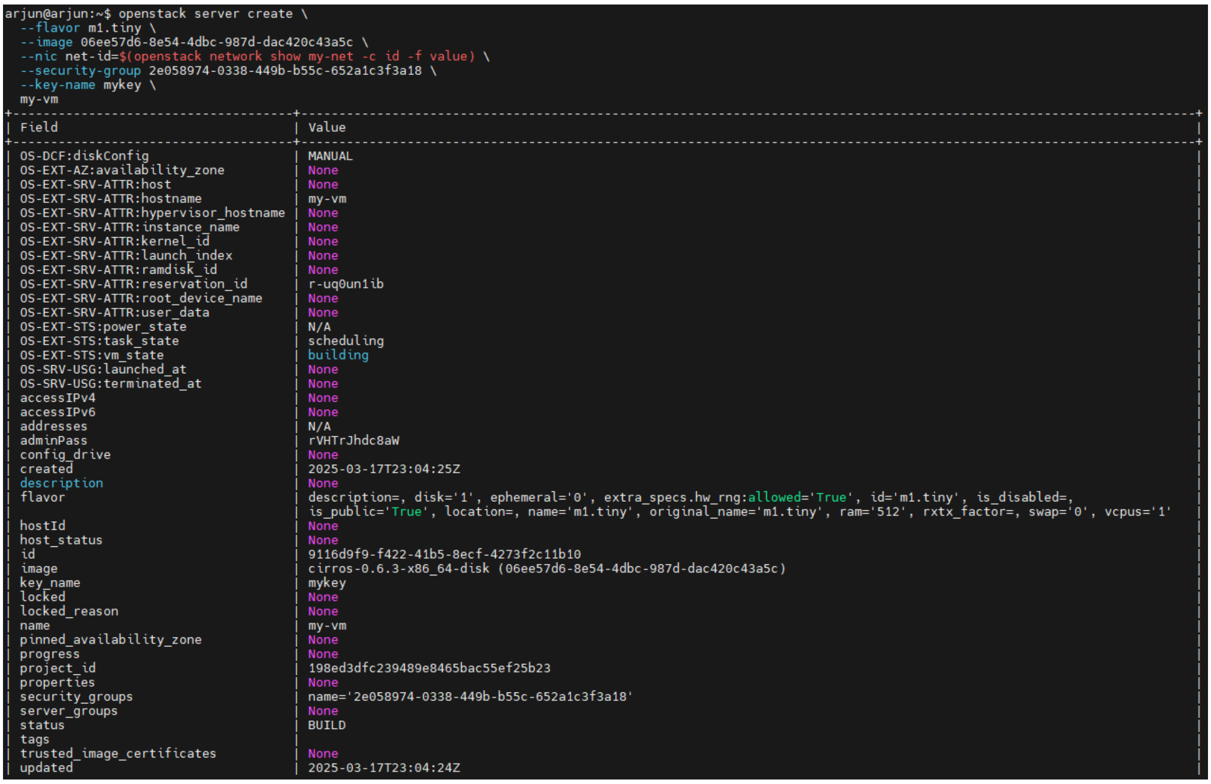
```
arjun@arjun:~$ openstack subnet create --network my-net --subnet-range 192.168.0.0/24 my-subnet
+----------------------+--------------------------------------+
| Field                | Value                                |
+----------------------+--------------------------------------+
| allocation_pools     | 192.168.0.2-192.168.0.254            |
| cidr                 | 192.168.0.0/24                       |
| created_at           | 2025-03-17T22:59:45Z                 |
| description          |                                      |
| dns_nameservers      |                                      |
| dns_publish_fixed_ip | None                                 |
| enable_dhcp          | True                                 |
| gateway_ip           | 192.168.0.1                          |
| host_routes          |                                      |
| id                   | 37163752-04d3-499b-8cb5-dc4eea8e8c13 |
| ip_version           | 4                                    |
| ipv6_address_mode    | None                                 |
| ipv6_ra_mode         | None                                 |
| name                 | my-subnet                            |
| network_id           | a6843660-800c-4e3a-9dbd-c65c69063c39 |
| project_id           | 198ed3dfc239489e8465bac55ef25b23     |
| revision_number      | 0                                    |
| router:external      | False                                |
| segment_id           | None                                 |
| service_types        |                                      |
| subnetpool_id        | None                                 |
| tags                 |                                      |
| updated_at           | 2025-03-17T22:59:45Z                 |
+----------------------+--------------------------------------+
```

- Creating openstack router for connection between new vm and public interface

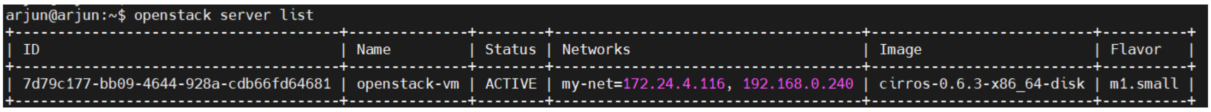


```
arjun@arjun:~$ openstack router create my-router
+---------------------------+--------------------------------------+
| Field                     | Value                                |
+---------------------------+--------------------------------------+
| admin_state_up            | UP                                   |
| availability_zone_hints   |                                      |
| availability_zones        |                                      |
| created_at                | 2025-03-17T23:00:19Z                 |
| description               |                                      |
| enable_default_route_bfd  | False                                |
| enable_default_route_ecmp | False                                |
| enable_ndp_proxy          | None                                 |
| external_gateway_info     | null                                 |
| external_gateways         | []                                   |
| flavor_id                 | None                                 |
| ha                        | True                                 |
| id                        | bd3fd59a-d708-44d5-9e9d-f136a37241b2 |
| name                      | my-router                            |
| project_id                | 198ed3dfc239489e8465bac55ef25b23     |
| revision_number           | 1                                    |
| routes                    |                                      |
| status                    | ACTIVE                               |
| tags                      |                                      |
| tenant_id                 | 198ed3dfc239489e8465bac55ef25b23     |
| updated_at                | 2025-03-17T23:00:19Z                 |
+---------------------------+--------------------------------------+
```

- Creating the VM now

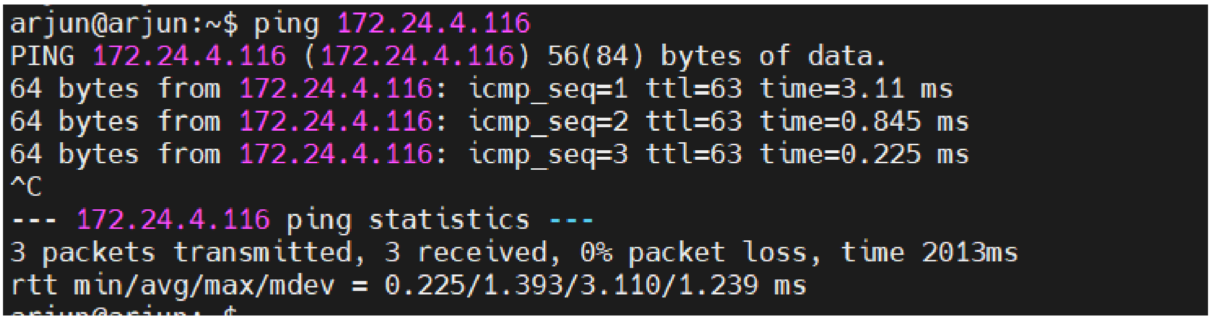


```
arjun@arjun:~$ openstack server create \
  --flavor m1.tiny \
  --image 06ee57d6-8e54-4dbc-987d-dac420c43a5c \
  --nic net-id=$(openstack network show my-net -c id -f value) \
  --security-group 2e058974-0338-449b-b55c-652a1c3f3a18 \
  --key-name mykey \
  my-vm
+-------------------------------------+------------------------------------------------------------------------------------------------------------------------+
| Field                               | Value
+-------------------------------------+------------------------------------------------------------------------------------------------------------------------+
| OS-DCF:diskConfig                   | MANUAL
| OS-EXT-AZ:availability_zone         | None
| OS-EXT-SRV-ATTR:host                | None
| OS-EXT-SRV-ATTR:hostname            | my-vm
| OS-EXT-SRV-ATTR:hypervisor_hostname | None
| OS-EXT-SRV-ATTR:instance_name       | None
| OS-EXT-SRV-ATTR:kernel_id           | None
| OS-EXT-SRV-ATTR:launch_index        | None
| OS-EXT-SRV-ATTR:ramdisk_id          | None
| OS-EXT-SRV-ATTR:reservation_id      | r-uq0un1ib
| OS-EXT-SRV-ATTR:root_device_name    | None
| OS-EXT-SRV-ATTR:user_data           | None
| OS-EXT-STS:power_state              | N/A
| OS-EXT-STS:task_state               | scheduling
| OS-EXT-STS:vm_state                 | building
| OS-SRV-USG:launched_at              | None
| OS-SRV-USG:terminated_at            | None
| accessIPv4                          | None
| accessIPv6                          | None
| addresses                           | N/A
| adminPass                           | rVHTrJhdc8aW
| config_drive                        | None
| created                             | 2025-03-17T23:04:25Z
| description                         | None
| flavor                              | description=, disk='1', ephemeral='0', extra_specs.hw_rng:allowed='True', id='m1.tiny', is_disabled=,
|                                     | is_public='True', location=, name='m1.tiny', original_name='m1.tiny', ram='512', rxtx_factor=, swap='0', vcpus='1'
| hostId                              | None
| host_status                         | None
| id                                  | 9116d9f9-f422-41b5-8ecf-4273f2c11b10
| image                               | cirros-0.6.3-x86_64-disk (06ee57d6-8e54-4dbc-987d-dac420c43a5c)
| key_name                            | mykey
| locked                              | None
| locked_reason                       | None
| name                                | my-vm
| pinned_availability_zone            | None
| progress                            | None
| project_id                          | 198ed3dfc239489e8465bac55ef25b23
| properties                          | None
| security_groups                     | name='2e058974-0338-449b-b55c-652a1c3f3a18'
| server_groups                       | None
| status                              | BUILD
| tags                                |
| trusted_image_certificates          | None
| updated                             | 2025-03-17T23:04:24Z
```



```
arjun@arjun:~$ openstack server list
+--------------------------------------+--------------+--------+----------------------------------------+--------------------------+----------+
| ID                                   | Name         | Status | Networks                               | Image                    | Flavor   |
+--------------------------------------+--------------+--------+----------------------------------------+--------------------------+----------+
| 7d79c177-bb09-4644-928a-cdb66fd64681 | openstack-vm | ACTIVE | my-net=172.24.4.116, 192.168.0.240     | cirros-0.6.3-x86_64-disk | m1.small |
+--------------------------------------+--------------+--------+----------------------------------------+--------------------------+----------+
```

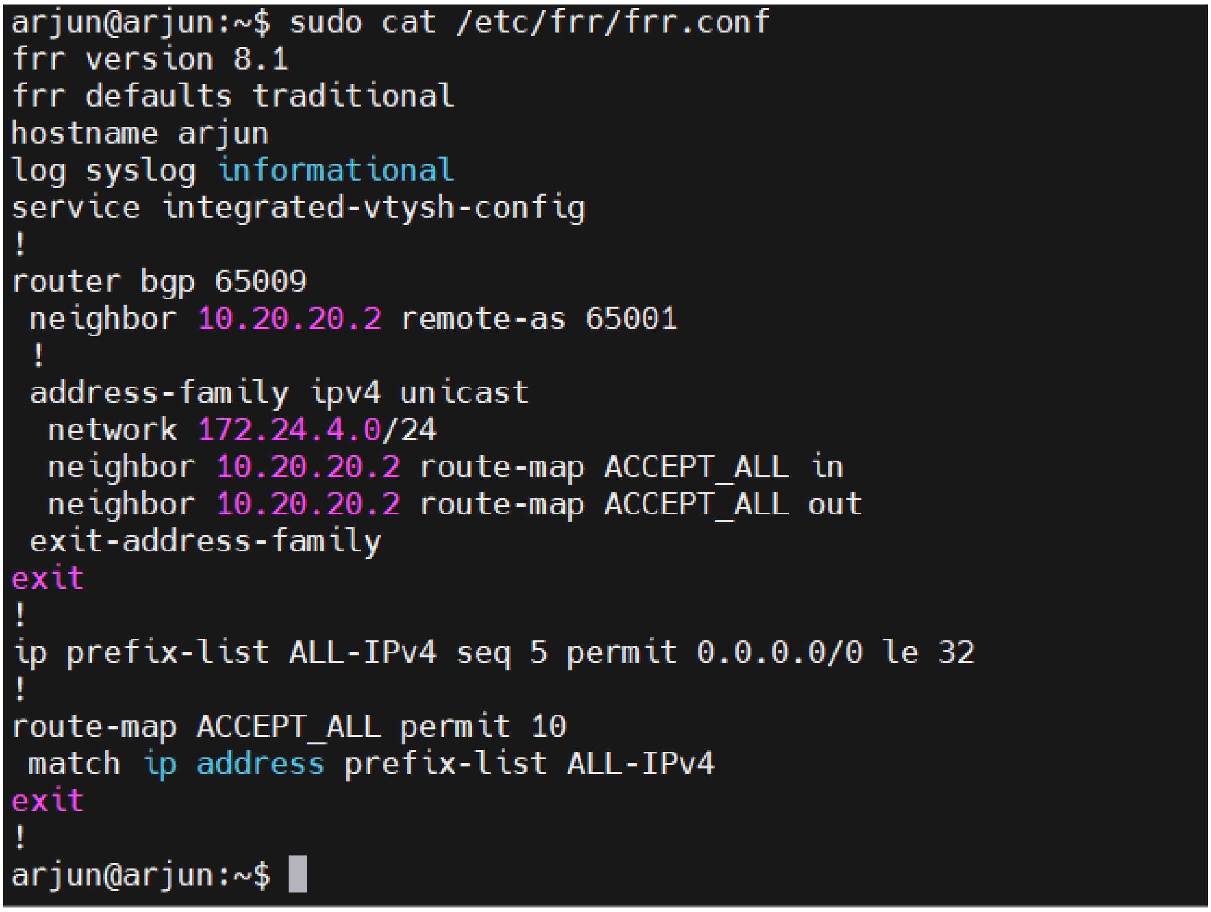
- Assigned floating ip to it and checked the connectivity to it



```
arjun@arjun:~$ ping 172.24.4.116
PING 172.24.4.116 (172.24.4.116) 56(84) bytes of data.
64 bytes from 172.24.4.116: icmp_seq=1 ttl=63 time=3.11 ms
64 bytes from 172.24.4.116: icmp_seq=2 ttl=63 time=0.845 ms
64 bytes from 172.24.4.116: icmp_seq=3 ttl=63 time=0.225 ms
^C
--- 172.24.4.116 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2013ms
rtt min/avg/max/mdev = 0.225/1.393/3.110/1.239 ms
arjun@arjun:~$
```

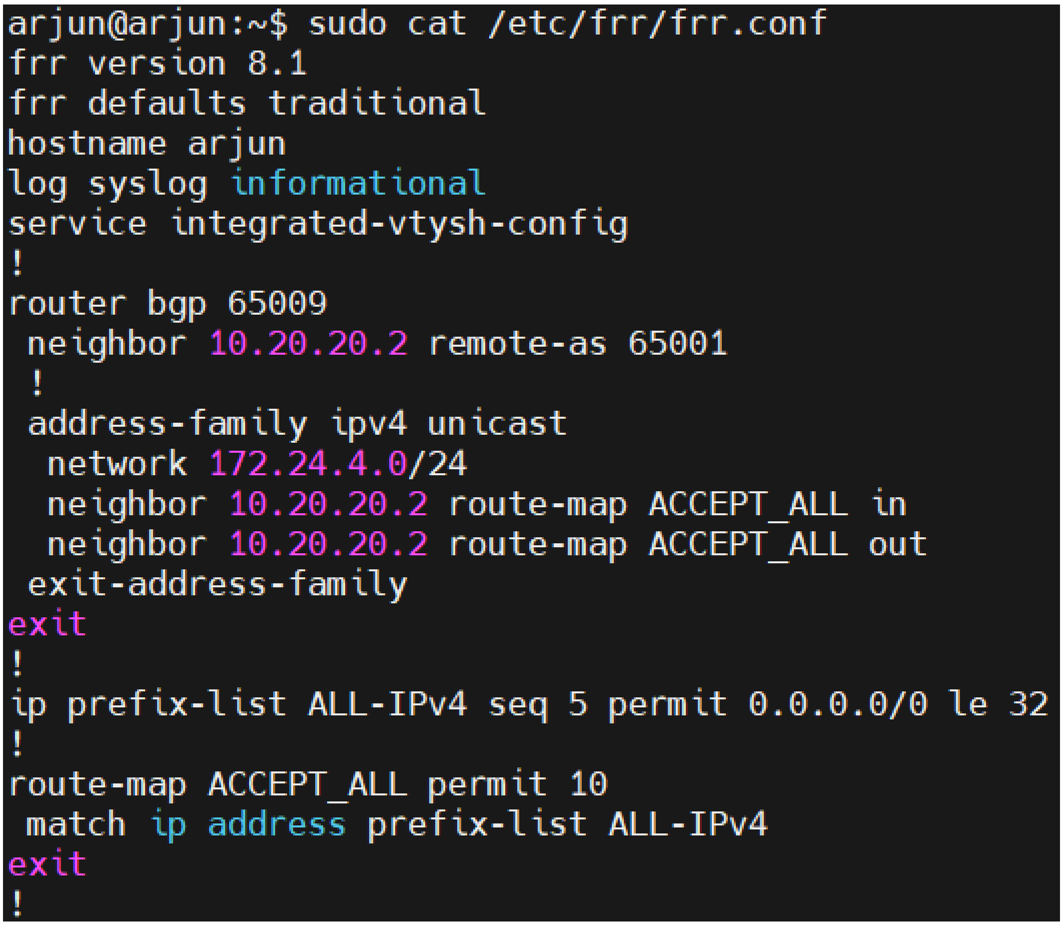
Section 6 – [Extra Credit] BGP via OpenStack

- After installing frr on my server **Nufais** and i configured Configuring frr for ebgp neighborship



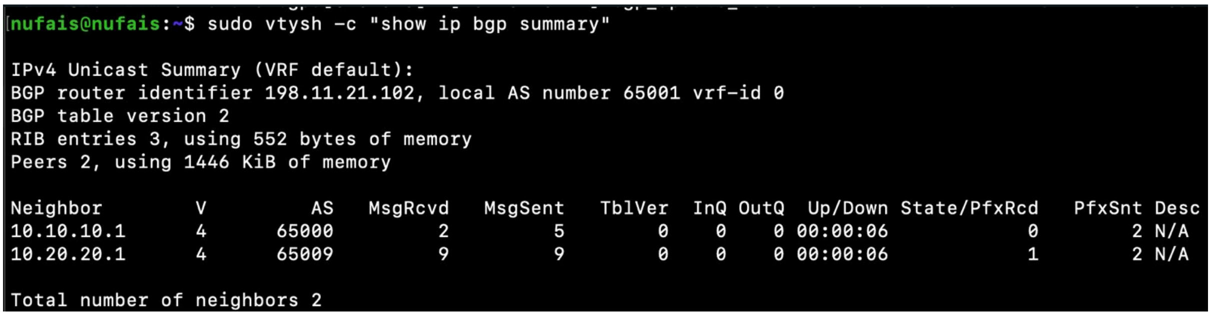
```
arjun@arjun:~$ sudo cat /etc/frr/frr.conf
frr version 8.1
frr defaults traditional
hostname arjun
log syslog informational
service integrated-vtysh-config
!
router bgp 65009
 neighbor 10.20.20.2 remote-as 65001
 !
 address-family ipv4 unicast
  network 172.24.4.0/24
  neighbor 10.20.20.2 route-map ACCEPT_ALL in
  neighbor 10.20.20.2 route-map ACCEPT_ALL out
 exit-address-family
exit
!
ip prefix-list ALL-IPv4 seq 5 permit 0.0.0.0/0 le 32
!
route-map ACCEPT_ALL permit 10
 match ip address prefix-list ALL-IPv4
exit
!
arjun@arjun:~$
```

- Ebgp route map for prefix delegation



```
arjun@arjun:~$ sudo cat /etc/frr/frr.conf
frr version 8.1
frr defaults traditional
hostname arjun
log syslog informational
service integrated-vtysh-config
!
router bgp 65009
 neighbor 10.20.20.2 remote-as 65001
 !
 address-family ipv4 unicast
  network 172.24.4.0/24
  neighbor 10.20.20.2 route-map ACCEPT_ALL in
  neighbor 10.20.20.2 route-map ACCEPT_ALL out
 exit-address-family
exit
!
ip prefix-list ALL-IPv4 seq 5 permit 0.0.0.0/0 le 32
!
route-map ACCEPT_ALL permit 10
 match ip address prefix-list ALL-IPv4
exit
!
```

- Ebgp neighborship with Nufais server



```
nufais@nufais:~$ sudo vtysh -c "show ip bgp summary"

IPv4 Unicast Summary (VRF default):
BGP router identifier 198.11.21.102, local AS number 65001 vrf-id 0
BGP table version 2
RIB entries 3, using 552 bytes of memory
Peers 2, using 1446 KiB of memory

Neighbor        V         AS   MsgRcvd   MsgSent   TblVer  InQ OutQ  Up/Down State/PfxRcd   PfxSnt Desc
10.10.10.1      4      65000         2         5        0    0    0 00:00:06            0        2 N/A
10.20.20.1      4      65009         9         9        0    0    0 00:00:06            1        2 N/A

Total number of neighbors 2
```

Server 2 Setup:

Objective

- Configure an External BGP (eBGP) session between:
  - Server 2 running FRR in Autonomous System (AS) 65001
  - Cisco Router in Autonomous System (AS) 65000

1. Assign IP address

- The **server 4** is connected via interface **eno3** with IP **10.10.10.2/24**.
- The **Cisco router** is connected via interface **FastEthernet2/0** with IP **10.10.10.1/24**.

2. Configure FRRouting (FRR) on Server 2

2.1 Install FRR

1. Update package lists and install FRR:

```
sudo apt update
sudo apt install frr
```

2. Enable the BGP daemon in /etc/frr/daemons:

```
bgpd=yes
```

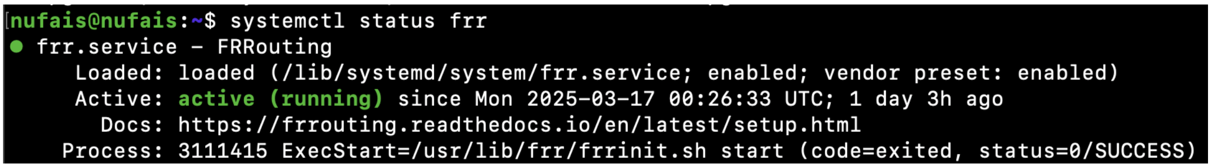
3. Start or restart FRR services:

```
sudo systemctl restart frr
```

2.2 Basic Verification

- Confirm that the FRR services are running:

```
systemctl status frr
```



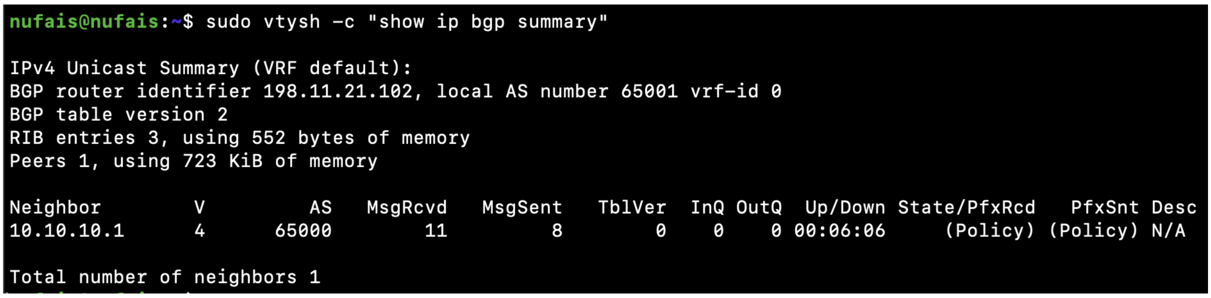
```
nufais@nufais:~$ systemctl status frr
● frr.service - FRRouting
     Loaded: loaded (/lib/systemd/system/frr.service; enabled; vendor preset: enabled)
     Active: active (running) since Mon 2025-03-17 00:26:33 UTC; 1 day 3h ago
       Docs: https://frrouting.readthedocs.io/en/latest/setup.html
    Process: 3111415 ExecStart=/usr/lib/frr/frrinit.sh start (code=exited, status=0/SUCCESS)
```

2.3 Configure BGP in FRR

1. **Edit the main FRR configuration** file **/etc/frr/frr.conf**
2. **Add the following lines** to define BGP with local AS 65001 and peer with 10.10.10.1 in AS 65000:

```
router bgp 65001
 neighbor 10.10.10.1 remote-as 65000
 !
 address-family ipv4 unicast
  network 10.10.10.0/24
  network 10.20.20.0/24
 exit-address-family
exit
```

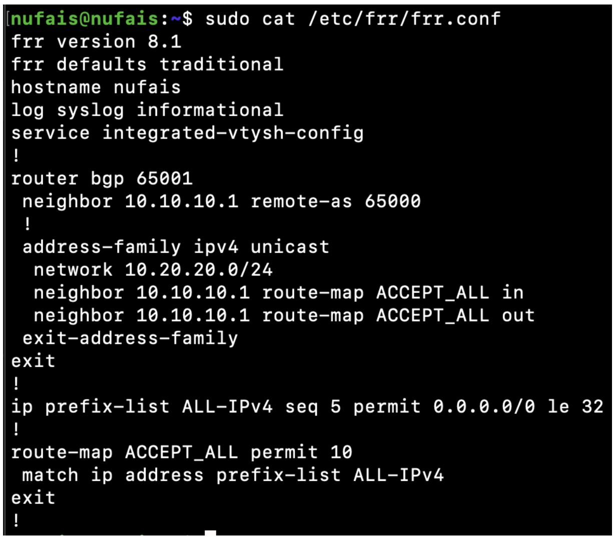
3. Missing Policy:



```
nufais@nufais:~$ sudo vtysh -c "show ip bgp summary"

IPv4 Unicast Summary (VRF default):
BGP router identifier 198.11.21.102, local AS number 65001 vrf-id 0
BGP table version 2
RIB entries 3, using 552 bytes of memory
Peers 1, using 723 KiB of memory

Neighbor        V         AS   MsgRcvd   MsgSent   TblVer  InQ OutQ  Up/Down State/PfxRcd   PfxSnt Desc
10.10.10.1      4      65000        11         8        0    0    0 00:06:06     (Policy) (Policy) N/A

Total number of neighbors 1
```

- FRR may discard updates if it expects route maps or prefix lists. A simple fix is to define and apply a “permit all” route map if needed:

```
ip prefix-list ALL-IPv4 seq 5 permit 0.0.0.0/0 le 32
route-map ACCEPT_ALL permit 10
  match ip address prefix-list ALL-IPv4

router bgp 65001
  address-family ipv4 unicast
    neighbor 10.10.10.1 route-map ACCEPT_ALL in
    neighbor 10.10.10.1 route-map ACCEPT_ALL out
```



```
nufais@nufais:~$ sudo cat /etc/frr/frr.conf
frr version 8.1
frr defaults traditional
hostname nufais
log syslog informational
service integrated-vtysh-config
!
router bgp 65001
 neighbor 10.10.10.1 remote-as 65000
 !
 address-family ipv4 unicast
  network 10.20.20.0/24
  neighbor 10.10.10.1 route-map ACCEPT_ALL in
  neighbor 10.10.10.1 route-map ACCEPT_ALL out
 exit-address-family
exit
!
ip prefix-list ALL-IPv4 seq 5 permit 0.0.0.0/0 le 32
!
route-map ACCEPT_ALL permit 10
 match ip address prefix-list ALL-IPv4
exit
!
```

4. **Save & Restart FRR:**

```
sudo systemctl restart frr
```

3. Configure BGP on the Cisco Router

1. Enter **Global Configuration Mode**:

```
enable
configure terminal
```

2. Configure the router for BGP using AS 65000:

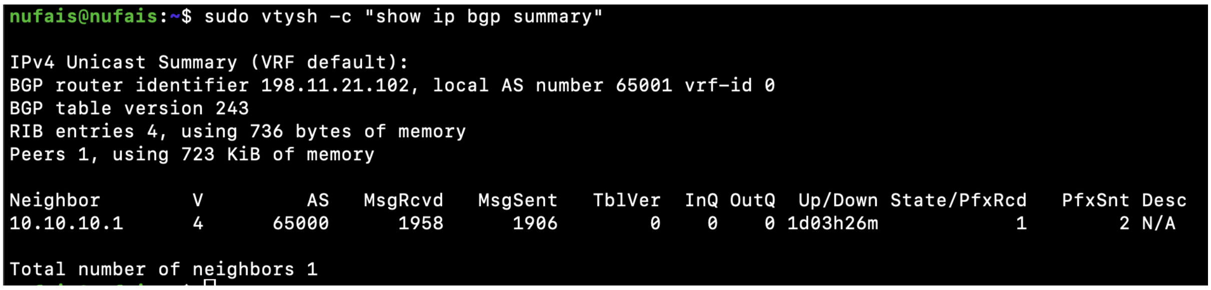
```
router bgp 65000
 neighbor 10.10.10.2 remote-as 65001
 address-family ipv4
  neighbor 10.10.10.2 activate
  network 10.10.10.0 mask 255.255.255.0
```

4. Verifying BGP Session

4.1 On the Ubuntu Server (FRR)

- **BGP Summary**:

```
sudo vtysh -c "show ip bgp summary"
```

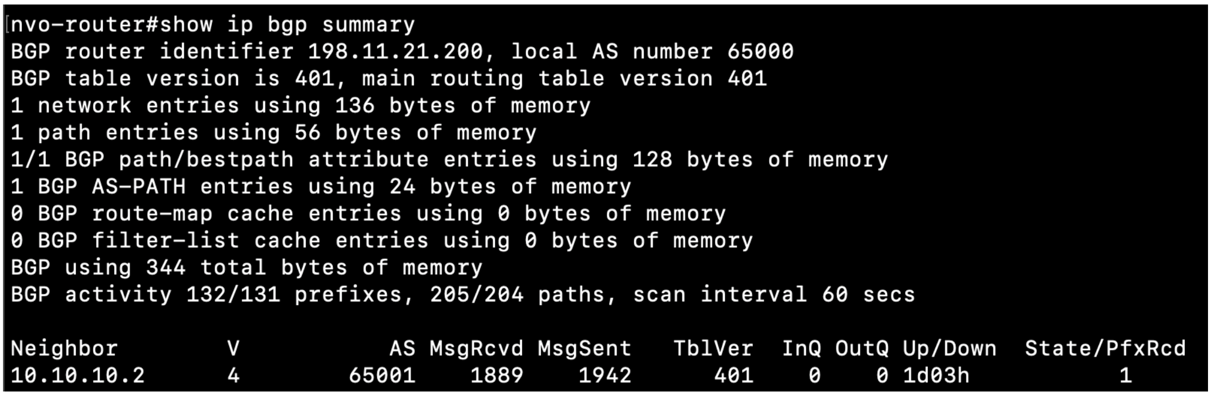


```
nufais@nufais:~$ sudo vtysh -c "show ip bgp summary"

IPv4 Unicast Summary (VRF default):
BGP router identifier 198.11.21.102, local AS number 65001 vrf-id 0
BGP table version 243
RIB entries 4, using 736 bytes of memory
Peers 1, using 723 KiB of memory

Neighbor        V         AS   MsgRcvd   MsgSent   TblVer  InQ OutQ  Up/Down State/PfxRcd   PfxSnt Desc
10.10.10.1      4      65000      1958      1906        0    0    0 1d03h26m            1        2 N/A

Total number of neighbors 1
```

4.2 On the Cisco Router

- **BGP Summary**:

```
show ip bgp summary
```



```
nvo-router#show ip bgp summary
BGP router identifier 198.11.21.200, local AS number 65000
BGP table version is 401, main routing table version 401
1 network entries using 136 bytes of memory
1 path entries using 56 bytes of memory
1/1 BGP path/bestpath attribute entries using 128 bytes of memory
1 BGP AS-PATH entries using 24 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 344 total bytes of memory
BGP activity 132/131 prefixes, 205/204 paths, scan interval 60 secs

Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
10.10.10.2      4        65001    1889    1942      401    0    0 1d03h           1
```

Server 3 Setup:

Tools Used

GoBGP was used for the Open Source routing on Server 3. A python virtual environment was utilized to separate packages/processes from the machine that is also running openstack.

GoBGP Configuration File

```
[global.config]
  as = 65003
  router-id = "1.1.1.1"

[[neighbors]]
  [neighbors.config]
    neighbor-address = "192.168.4.1"
    peer-as = 65004

[[neighbors]]
  [neighbors.config]
    neighbor-address = "20.0.0.1"
    peer-as = 65000
```

ASN 65003 exists on server 4 which is peering with Cisco Router ASN 65000 and Server 4 65004.

Virtual Environment Setup and Execution

To isolate conflicting packages with my server running openstack, I ran a python virtual environment to start exaBGP.

Ashwin, Sneha, Nufais, Logan & Arjun

Step 1: Create a python virtual environment and set up a exabgp-project directory folder

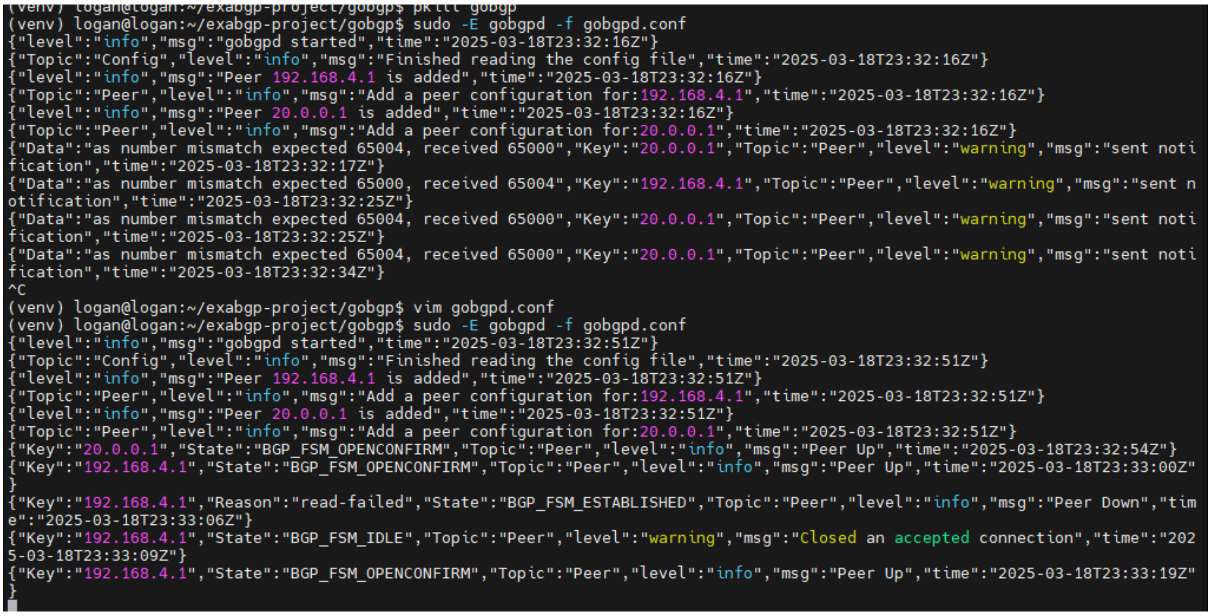
```
mkdir ~/gobgpd-project && cd ~/exabgp-project
python3 -m venv venv
```

Step 2: Create exabgp.conf and announce.py files and add above code

```
mkdir config && touch config/gobgpd.conf
```

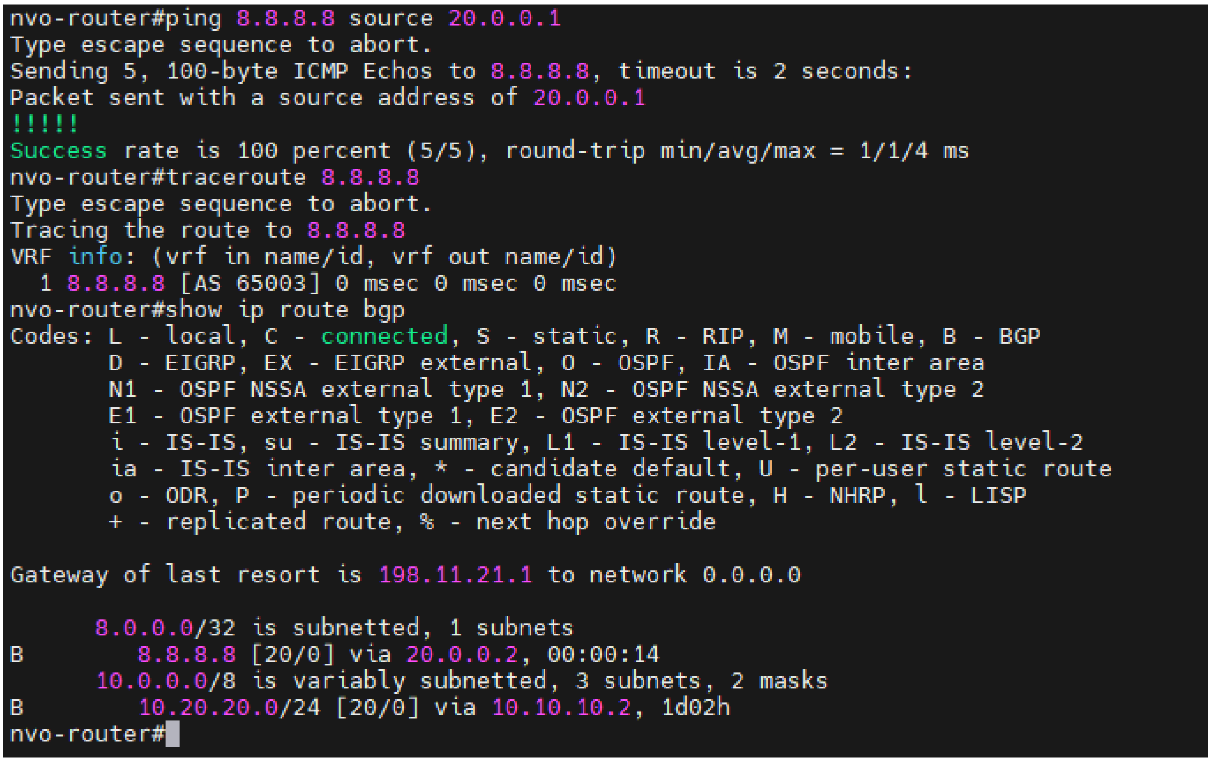
Step 3: Start GoBGP and verify connectivity

Server 3 Logs:



```
(venv) logan@logan:~/exabgp-project/gobgp$ pkill gobgp
(venv) logan@logan:~/exabgp-project/gobgp$ sudo -E gobgpd -f gobgpd.conf
{"level":"info","msg":"gobgpd started","time":"2025-03-18T23:32:16Z"}
{"Topic":"Config","level":"info","msg":"Finished reading the config file","time":"2025-03-18T23:32:16Z"}
{"level":"info","msg":"Peer 192.168.4.1 is added","time":"2025-03-18T23:32:16Z"}
{"Topic":"Peer","level":"info","msg":"Add a peer configuration for:192.168.4.1","time":"2025-03-18T23:32:16Z"}
{"level":"info","msg":"Peer 20.0.0.1 is added","time":"2025-03-18T23:32:16Z"}
{"Topic":"Peer","level":"info","msg":"Add a peer configuration for:20.0.0.1","time":"2025-03-18T23:32:16Z"}
{"Data":"as number mismatch expected 65004, received 65000","Key":"20.0.0.1","Topic":"Peer","level":"warning","msg":"sent noti
fication","time":"2025-03-18T23:32:17Z"}
{"Data":"as number mismatch expected 65000, received 65004","Key":"192.168.4.1","Topic":"Peer","level":"warning","msg":"sent n
otification","time":"2025-03-18T23:32:25Z"}
{"Data":"as number mismatch expected 65004, received 65000","Key":"20.0.0.1","Topic":"Peer","level":"warning","msg":"sent noti
fication","time":"2025-03-18T23:32:25Z"}
{"Data":"as number mismatch expected 65004, received 65000","Key":"20.0.0.1","Topic":"Peer","level":"warning","msg":"sent noti
fication","time":"2025-03-18T23:32:34Z"}
^C
(venv) logan@logan:~/exabgp-project/gobgp$ vim gobgpd.conf
(venv) logan@logan:~/exabgp-project/gobgp$ sudo -E gobgpd -f gobgpd.conf
{"level":"info","msg":"gobgpd started","time":"2025-03-18T23:32:51Z"}
{"Topic":"Config","level":"info","msg":"Finished reading the config file","time":"2025-03-18T23:32:51Z"}
{"level":"info","msg":"Peer 192.168.4.1 is added","time":"2025-03-18T23:32:51Z"}
{"Topic":"Peer","level":"info","msg":"Add a peer configuration for:192.168.4.1","time":"2025-03-18T23:32:51Z"}
{"level":"info","msg":"Peer 20.0.0.1 is added","time":"2025-03-18T23:32:51Z"}
{"Topic":"Peer","level":"info","msg":"Add a peer configuration for:20.0.0.1","time":"2025-03-18T23:32:51Z"}
{"Key":"20.0.0.1","State":"BGP_FSM_OPENCONFIRM","Topic":"Peer","level":"info","msg":"Peer Up","time":"2025-03-18T23:32:54Z"}
{"Key":"192.168.4.1","State":"BGP_FSM_OPENCONFIRM","Topic":"Peer","level":"info","msg":"Peer Up","time":"2025-03-18T23:33:00Z"
}
{"Key":"192.168.4.1","Reason":"read-failed","State":"BGP_FSM_ESTABLISHED","Topic":"Peer","level":"info","msg":"Peer Down","tim
e":"2025-03-18T23:33:06Z"}
{"Key":"192.168.4.1","State":"BGP_FSM_IDLE","Topic":"Peer","level":"warning","msg":"Closed an accepted connection","time":"202
5-03-18T23:33:09Z"}
{"Key":"192.168.4.1","State":"BGP_FSM_OPENCONFIRM","Topic":"Peer","level":"info","msg":"Peer Up","time":"2025-03-18T23:33:19Z"
}
```

Connectivity via Cisco Router:



```
nvo-router#ping 8.8.8.8 source 20.0.0.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 8.8.8.8, timeout is 2 seconds:
Packet sent with a source address of 20.0.0.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms
nvo-router#traceroute 8.8.8.8
Type escape sequence to abort.
Tracing the route to 8.8.8.8
VRF info: (vrf in name/id, vrf out name/id)
  1 8.8.8.8 [AS 65003] 0 msec 0 msec 0 msec
nvo-router#show ip route bgp
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       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
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       + - replicated route, % - next hop override

Gateway of last resort is 198.11.21.1 to network 0.0.0.0

      8.0.0.0/32 is subnetted, 1 subnets
B        8.8.8.8 [20/0] via 20.0.0.2, 00:00:14
      10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
B        10.20.20.0/24 [20/0] via 10.10.10.2, 1d02h
nvo-router#
```

Connectivity Caveats

GoBGP basically works as an application that can read TCP179 (BGP) packets. It comprehends and adds neighborships, and routes where you can then monitor via its CLI different neighborships/networks. The only thing we couldn’t figure out is that it does not process or forward traffic, it only has a routing information base of where everything is. To allow for connectivity to happen, I added static routes pointing to Server 4 and Server 1 networks. This allowed for ARP, L3 IP and ICMP to happen and pass through based on the BGP paths/policies that were set up from GoBGP.

Static route commands:

```
sudo ip route add 172.24.4.0/24 dev eno4
sudo ip route add 4.4.4.0/24 dev eno3
```

Server 4 Setup:

Objective

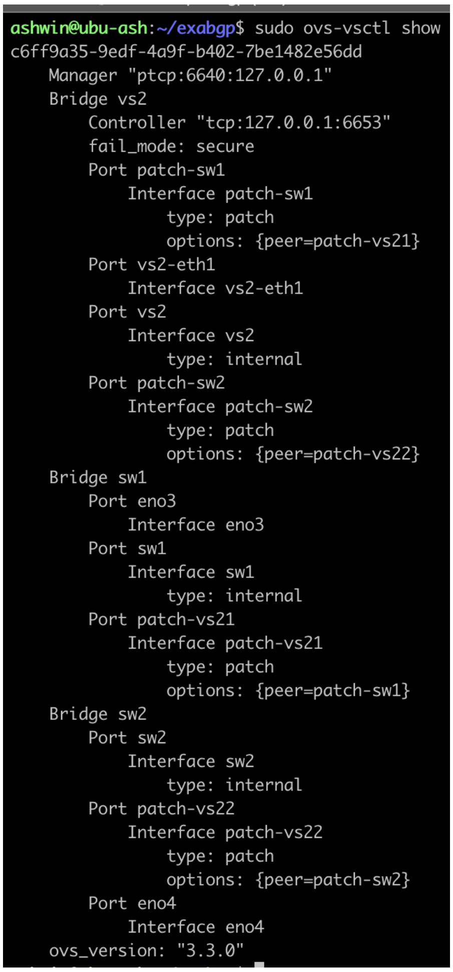
- Configure an eBGP session between:
  - Server 3 running ExaBGP in Autonomous System (AS) 65003
  - Cisco Router in Autonomous System (AS) 65000
  - Server 4 running Ryu BGP app in Autonomous System (AS) 65004

1. Assign IP address

- The **server 4** is connected to **server 3** via **eno3** with IP **192.168.4.1/24**
- The **server 4** is connected to **Cisco router** via **eno4** with IP **192.168.3.1/24**

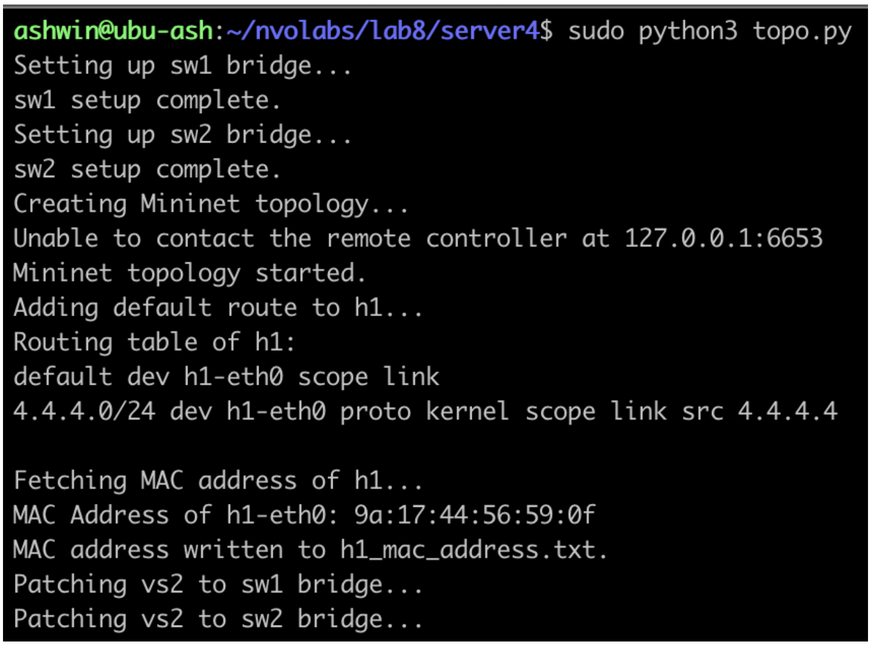
2. Steps to configure

- Creating OVS **sw1** and **sw2** with initial flows installed - for BGP peering.



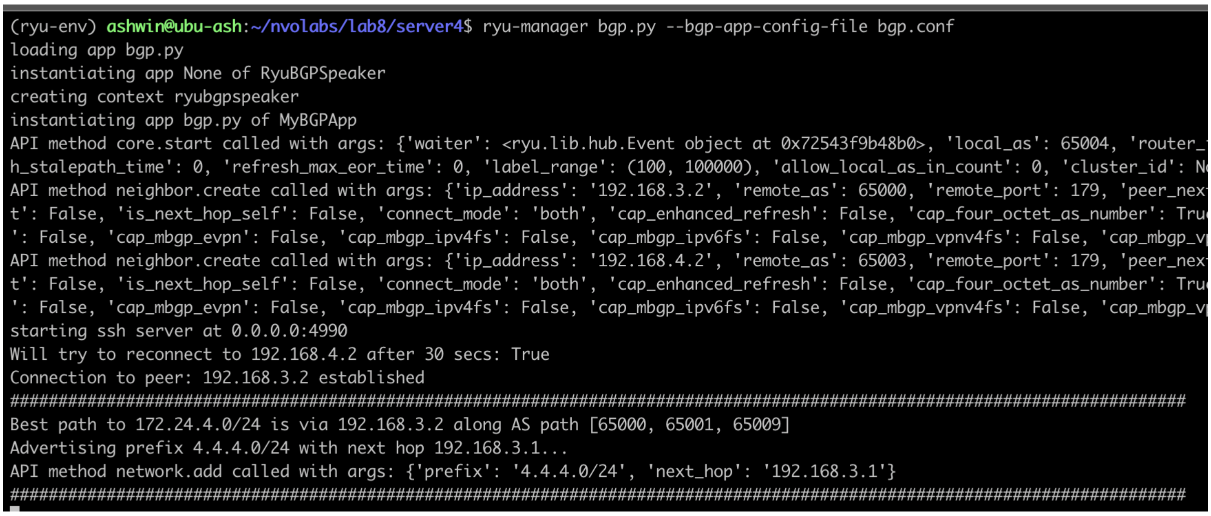
```
ashwin@ubu-ash:~/exabgp$ sudo ovs-vsctl show
c6ff9a35-9edf-4a9f-b402-7be1482e56dd
    Manager "ptcp:6640:127.0.0.1"
    Bridge vs2
        Controller "tcp:127.0.0.1:6653"
        fail_mode: secure
        Port patch-sw1
            Interface patch-sw1
                type: patch
                options: {peer=patch-vs21}
        Port vs2-eth1
            Interface vs2-eth1
        Port vs2
            Interface vs2
                type: internal
        Port patch-sw2
            Interface patch-sw2
                type: patch
                options: {peer=patch-vs22}
    Bridge sw1
        Port eno3
            Interface eno3
        Port sw1
            Interface sw1
                type: internal
        Port patch-vs21
            Interface patch-vs21
                type: patch
                options: {peer=patch-sw1}
    Bridge sw2
        Port sw2
            Interface sw2
                type: internal
        Port patch-vs22
            Interface patch-vs22
                type: patch
                options: {peer=patch-sw2}
        Port eno4
            Interface eno4
    ovs_version: "3.3.0"
```

- Setting up mininet hosts **(h1)** and mininet switch **(vs2)**.
- Patching the vs2 with sw1 and sw2, eno3 with sw1 and eno4 with sw2.



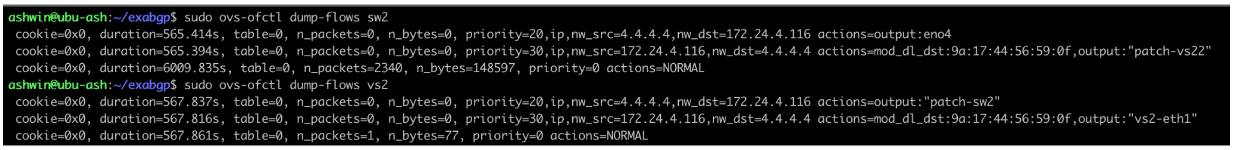
```
ashwin@ubu-ash:~/nvolabs/lab8/server4$ sudo python3 topo.py
Setting up sw1 bridge...
sw1 setup complete.
Setting up sw2 bridge...
sw2 setup complete.
Creating Mininet topology...
Unable to contact the remote controller at 127.0.0.1:6653
Mininet topology started.
Adding default route to h1...
Routing table of h1:
default dev h1-eth0 scope link
4.4.4.0/24 dev h1-eth0 proto kernel scope link src 4.4.4.4

Fetching MAC address of h1...
MAC Address of h1-eth0: 9a:17:44:56:59:0f
MAC address written to h1_mac_address.txt.
Patching vs2 to sw1 bridge...
Patching vs2 to sw2 bridge...
```

- Ryu controller requires BGP application configuration for the neighborship. So wrote a python file



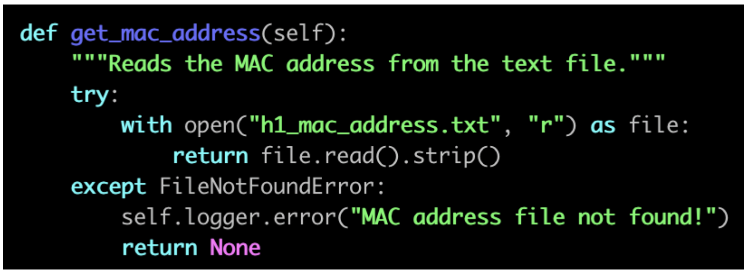
```
(ryu-env) ashwin@ubu-ash:~/nvolabs/lab8/server4$ ryu-manager bgp.py --bgp-app-config-file bgp.conf
loading app bgp.py
instantiating app None of RyuBGPSpeaker
creating context ryubgpspeaker
instantiating app bgp.py of MyBGPApp
API method core.start called with args: {'waiter': <ryu.lib.hub.Event object at 0x72543f9b48b0>, 'local_as': 65004, 'router_
h_stalepath_time': 0, 'refresh_max_eor_time': 0, 'label_range': (100, 100000), 'allow_local_as_in_count': 0, 'cluster_id': N
API method neighbor.create called with args: {'ip_address': '192.168.3.2', 'remote_as': 65000, 'remote_port': 179, 'peer_nex
t': False, 'is_next_hop_self': False, 'connect_mode': 'both', 'cap_enhanced_refresh': False, 'cap_four_octet_as_number': Tru
': False, 'cap_mbgp_evpn': False, 'cap_mbgp_ipv4fs': False, 'cap_mbgp_ipv6fs': False, 'cap_mbgp_vpnv4fs': False, 'cap_mbgp_v
API method neighbor.create called with args: {'ip_address': '192.168.4.2', 'remote_as': 65003, 'remote_port': 179, 'peer_nex
t': False, 'is_next_hop_self': False, 'connect_mode': 'both', 'cap_enhanced_refresh': False, 'cap_four_octet_as_number': Tru
': False, 'cap_mbgp_evpn': False, 'cap_mbgp_ipv4fs': False, 'cap_mbgp_ipv6fs': False, 'cap_mbgp_vpnv4fs': False, 'cap_mbgp_v
starting ssh server at 0.0.0.0:4990
Will try to reconnect to 192.168.4.2 after 30 secs: True
Connection to peer: 192.168.3.2 established
###########################################################################################################################
Best path to 172.24.4.0/24 is via 192.168.3.2 along AS path [65000, 65001, 65009]
Advertising prefix 4.4.4.0/24 with next hop 192.168.3.1...
API method network.add called with args: {'prefix': '4.4.4.0/24', 'next_hop': '192.168.3.1'}
###########################################################################################################################
```

- Dynamically adding and removing flows based on the path:



```
ashwin@ubu-ash:~/exabgp$ sudo ovs-ofctl dump-flows sw2
 cookie=0x0, duration=565.414s, table=0, n_packets=0, n_bytes=0, priority=20,ip,nw_src=4.4.4.4,nw_dst=172.24.4.116 actions=output:eno4
 cookie=0x0, duration=565.394s, table=0, n_packets=0, n_bytes=0, priority=30,ip,nw_src=172.24.4.116,nw_dst=4.4.4.4 actions=mod_dl_dst:9a:17:44:56:59:0f,output:"patch-vs22"
 cookie=0x0, duration=6009.835s, table=0, n_packets=2340, n_bytes=148597, priority=0 actions=NORMAL
ashwin@ubu-ash:~/exabgp$ sudo ovs-ofctl dump-flows vs2
 cookie=0x0, duration=567.837s, table=0, n_packets=0, n_bytes=0, priority=20,ip,nw_src=4.4.4.4,nw_dst=172.24.4.116 actions=output:"patch-sw2"
 cookie=0x0, duration=567.816s, table=0, n_packets=0, n_bytes=0, priority=30,ip,nw_src=172.24.4.116,nw_dst=4.4.4.4 actions=mod_dl_dst:9a:17:44:56:59:0f,output:"vs2-eth1"
 cookie=0x0, duration=567.861s, table=0, n_packets=1, n_bytes=77, priority=0 actions=NORMAL
```

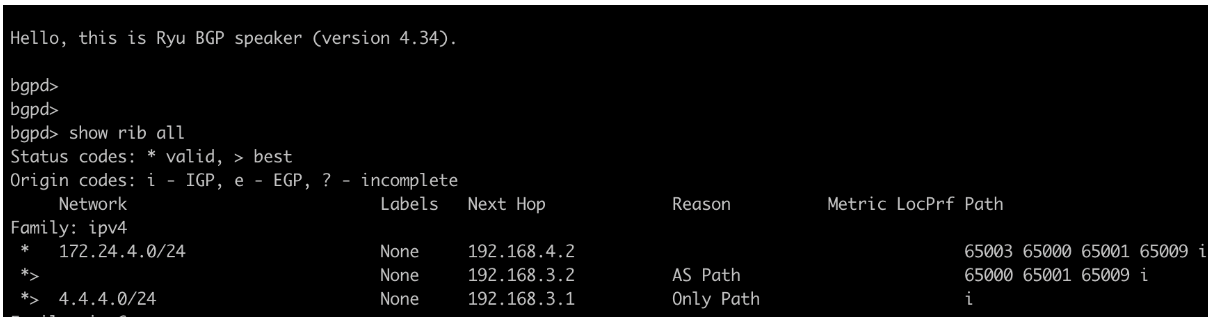
- Dynamically modifying the MAC address of the flows from the previous picture by capturing it on a txt file and redirecting it to bgp app.



```python
def get_mac_address(self):
    """Reads the MAC address from the text file."""
    try:
        with open("h1_mac_address.txt", "r") as file:
            return file.read().strip()
    except FileNotFoundError:
        self.logger.error("MAC address file not found!")
        return None
```

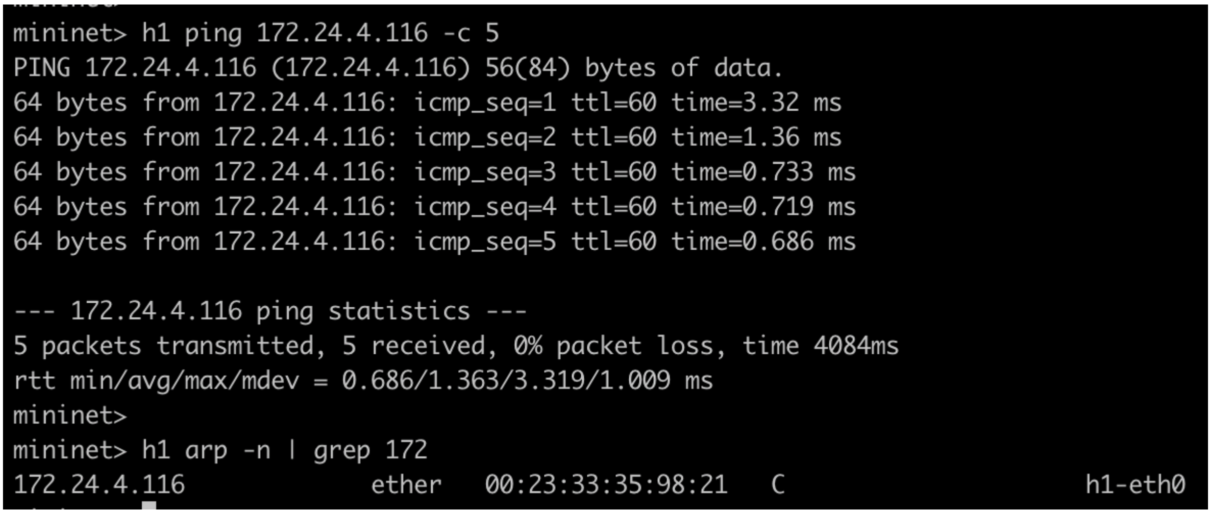
3. Testing

- Checking the rib of server 4 bgp engine to check the best path. And it chooses the path through cisco router



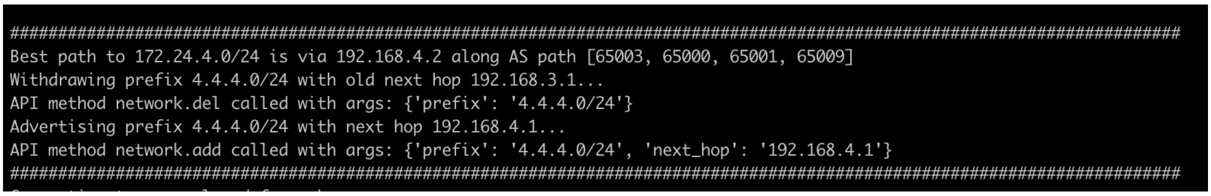
```
Hello, this is Ryu BGP speaker (version 4.34).

bgpd>
bgpd>
bgpd> show rib all
Status codes: * valid, > best
Origin codes: i - IGP, e - EGP, ? - incomplete
     Network                          Labels   Next Hop             Reason          Metric LocPrf Path
Family: ipv4
 *   172.24.4.0/24                    None     192.168.4.2                                        65003 65000 65001 65009 i
 *>                                   None     192.168.3.2          AS Path                       65000 65001 65009 i
 *>  4.4.4.0/24                       None     192.168.3.1          Only Path                     i
```

- Ping test to the host through path 1 [65004—> 65000 —-> 65001 —-> 65009]
- Since I run SDN, the host does not have a default gateway, so the destination ip arps to the next hop (cisco router) mac address



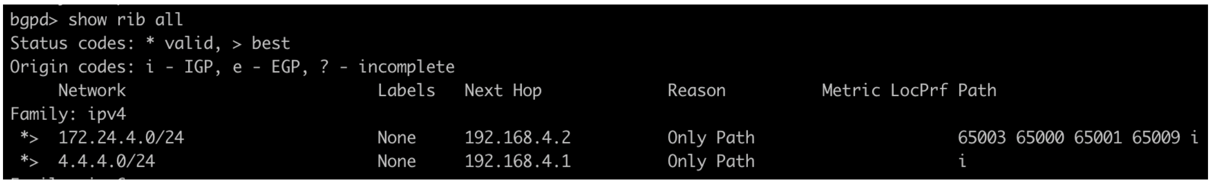
```
mininet> h1 ping 172.24.4.116 -c 5
PING 172.24.4.116 (172.24.4.116) 56(84) bytes of data.
64 bytes from 172.24.4.116: icmp_seq=1 ttl=60 time=3.32 ms
64 bytes from 172.24.4.116: icmp_seq=2 ttl=60 time=1.36 ms
64 bytes from 172.24.4.116: icmp_seq=3 ttl=60 time=0.733 ms
64 bytes from 172.24.4.116: icmp_seq=4 ttl=60 time=0.719 ms
64 bytes from 172.24.4.116: icmp_seq=5 ttl=60 time=0.686 ms

--- 172.24.4.116 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4084ms
rtt min/avg/max/mdev = 0.686/1.363/3.319/1.009 ms
mininet>
mininet> h1 arp -n | grep 172
172.24.4.116             ether   00:23:33:35:98:21   C                     h1-eth0
```

- Shutting down cisco interface to check the failover path



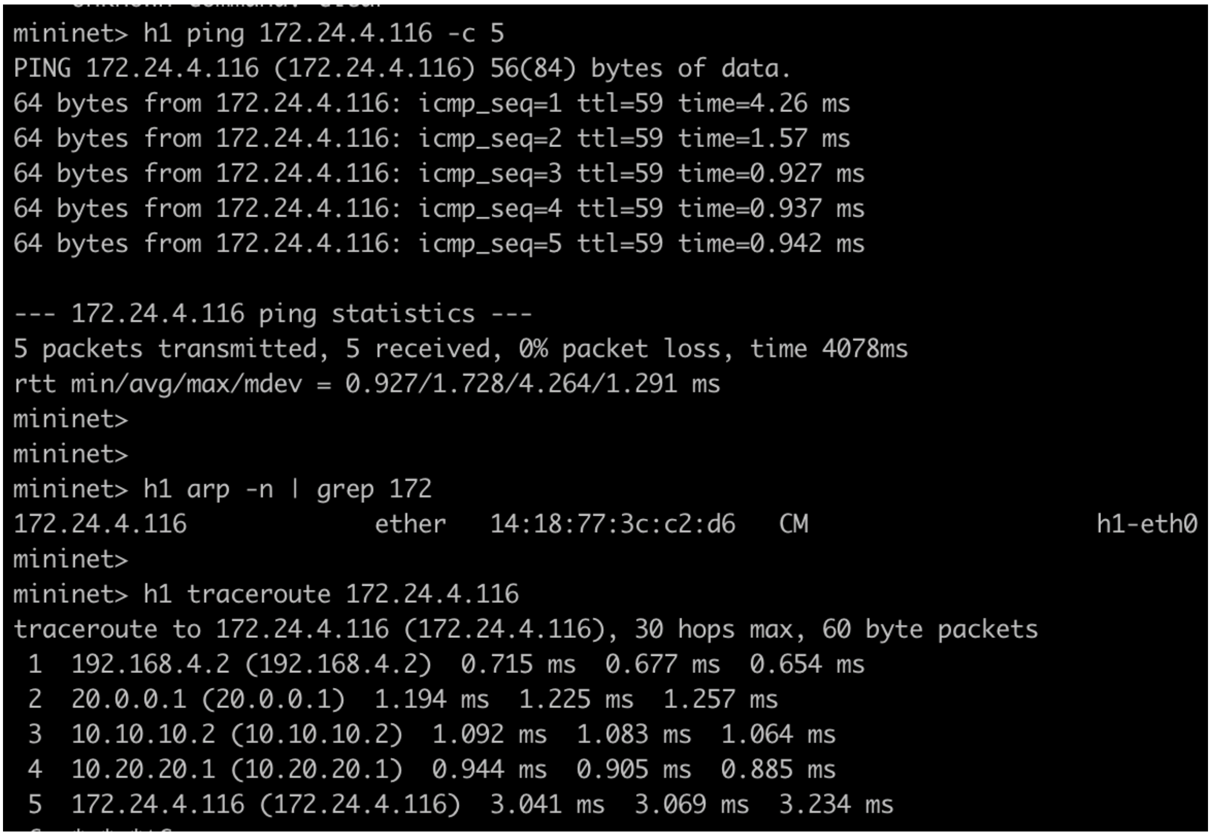
```
##########################################################################################################################
Best path to 172.24.4.0/24 is via 192.168.4.2 along AS path [65003, 65000, 65001, 65009]
Withdrawing prefix 4.4.4.0/24 with old next hop 192.168.3.1...
API method network.del called with args: {'prefix': '4.4.4.0/24'}
Advertising prefix 4.4.4.0/24 with next hop 192.168.4.1...
API method network.add called with args: {'prefix': '4.4.4.0/24', 'next_hop': '192.168.4.1'}
##########################################################################################################################
```

- New RIB entry for the updated route



```
bgpd> show rib all
Status codes: * valid, > best
Origin codes: i - IGP, e - EGP, ? - incomplete
    Network                         Labels   Next Hop             Reason          Metric LocPrf Path
Family: ipv4
 *>  172.24.4.0/24                  None     192.168.4.2          Only Path                     65003 65000 65001 65009 i
 *>  4.4.4.0/24                     None     192.168.4.1          Only Path                     i
```

- Ping and traceroute through server 3. The arp for the destination ip is resolved for the server 3 physical interface [65003 —> 65004—> 65000 —-> 65001 —-> 65009]



```
mininet> h1 ping 172.24.4.116 -c 5
PING 172.24.4.116 (172.24.4.116) 56(84) bytes of data.
64 bytes from 172.24.4.116: icmp_seq=1 ttl=59 time=4.26 ms
64 bytes from 172.24.4.116: icmp_seq=2 ttl=59 time=1.57 ms
64 bytes from 172.24.4.116: icmp_seq=3 ttl=59 time=0.927 ms
64 bytes from 172.24.4.116: icmp_seq=4 ttl=59 time=0.937 ms
64 bytes from 172.24.4.116: icmp_seq=5 ttl=59 time=0.942 ms

--- 172.24.4.116 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4078ms
rtt min/avg/max/mdev = 0.927/1.728/4.264/1.291 ms
mininet>
mininet>
mininet> h1 arp -n | grep 172
172.24.4.116             ether   14:18:77:3c:c2:d6   CM                    h1-eth0
mininet>
mininet> h1 traceroute 172.24.4.116
traceroute to 172.24.4.116 (172.24.4.116), 30 hops max, 60 byte packets
 1  192.168.4.2 (192.168.4.2)  0.715 ms  0.677 ms  0.654 ms
 2  20.0.0.1 (20.0.0.1)  1.194 ms  1.225 ms  1.257 ms
 3  10.10.10.2 (10.10.10.2)  1.092 ms  1.083 ms  1.064 ms
 4  10.20.20.1 (10.20.20.1)  0.944 ms  0.905 ms  0.885 ms
 5  172.24.4.116 (172.24.4.116)  3.041 ms  3.069 ms  3.234 ms
```

In this Lab, you will configure BGP in a virtualized, traditional, and SDN. See Figure 1.

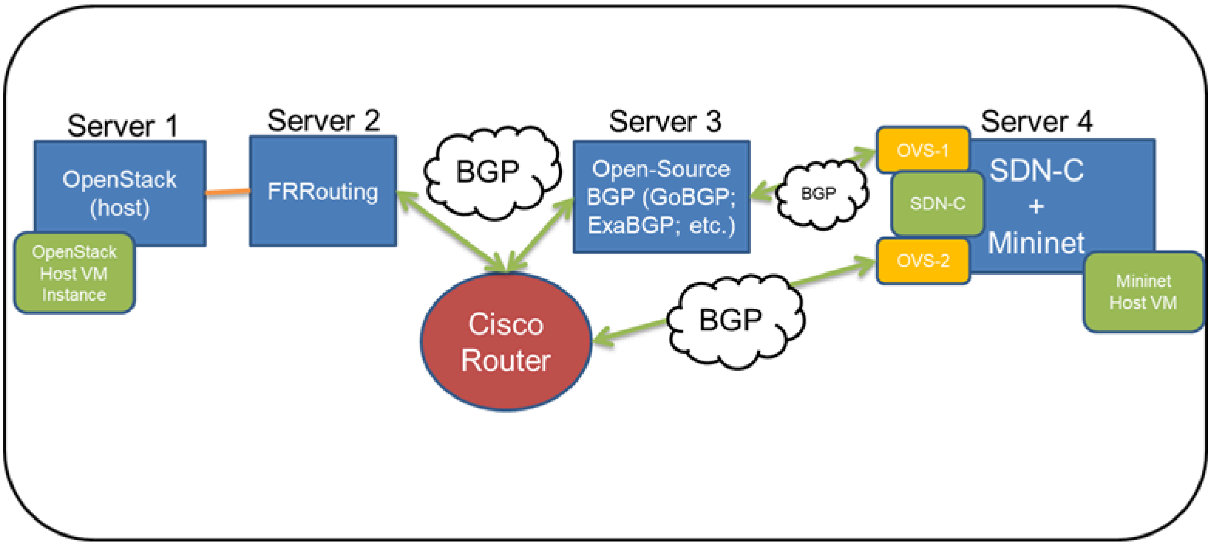


Figure 1: Lab environment

Each team will use one Cisco router and four servers running-

1. OpenStack
2. FRRouting [https://frrouting.org/]
3. SDN Controller (Ryu) + Mininet
4. Open-Source BGP (your choice)

Section 1 - Establish BGP Peering between FRRouting and Cisco router:

Configure BGP on FRRouting server and establish peering with the Cisco router.

[FRRouting is forked from Quagga and the installation process is the same. You can refer this document and replace Quagga with FRRouting -

http://www.brianlinkletter.com/how-to-build-a-network-of-linux-routers-using-quagga/]

Section 2 - Establish BGP Peering between Cisco Router and Open-Source BGP speaker:

Configure BGP on Cisco Router and establish a peering relationship of the Open-Source BGP speaker of your choice (GoBGP or ExaBGP recommended).

Section 3 – Establish BGP Peering between Open-Source BGP speaker and SDN Controller AND between Cisco Router and SDN Controller

In this objective, the SDN controller will need to dynamically add flow table entries (“routes”) to the Mininet OvS based on the peers. Failover should be accounted for (i.e. if the Open-Source speaker peering sessions goes down, all traffic should be routed through the other BGP peer).

Section 4 - IP Connectivity from VM to SDN:

The objective is to achieve communication between a Mininet host (Server 4) and an OpenStack instance (Server 1).

Final Deliverable

- Indicate via code that OpenStack host machine (host1) has connectivity to Mininet Host (host2), noting the correct hop/paths along the way.
    - BGP failover path must be accounted for.

Section 5 – [Extra Credit] BGP via Containers

Implement one of the sections/objectives in containerized form.

Section 6 – [Extra Credit] BGP via OpenStack