CSCI 5380 - Network Virtualization and Orchestration

Lab 9
Automate VM, VN, Docker, and BGP path

University of Colorado Boulder
Department of Computer Science
Network Engineering

Professor Levi Perigo, Ph.D.

Summary:

In this lab, you will use what you have learned in previous labs and automate the processes into a single application.

Required technologies:

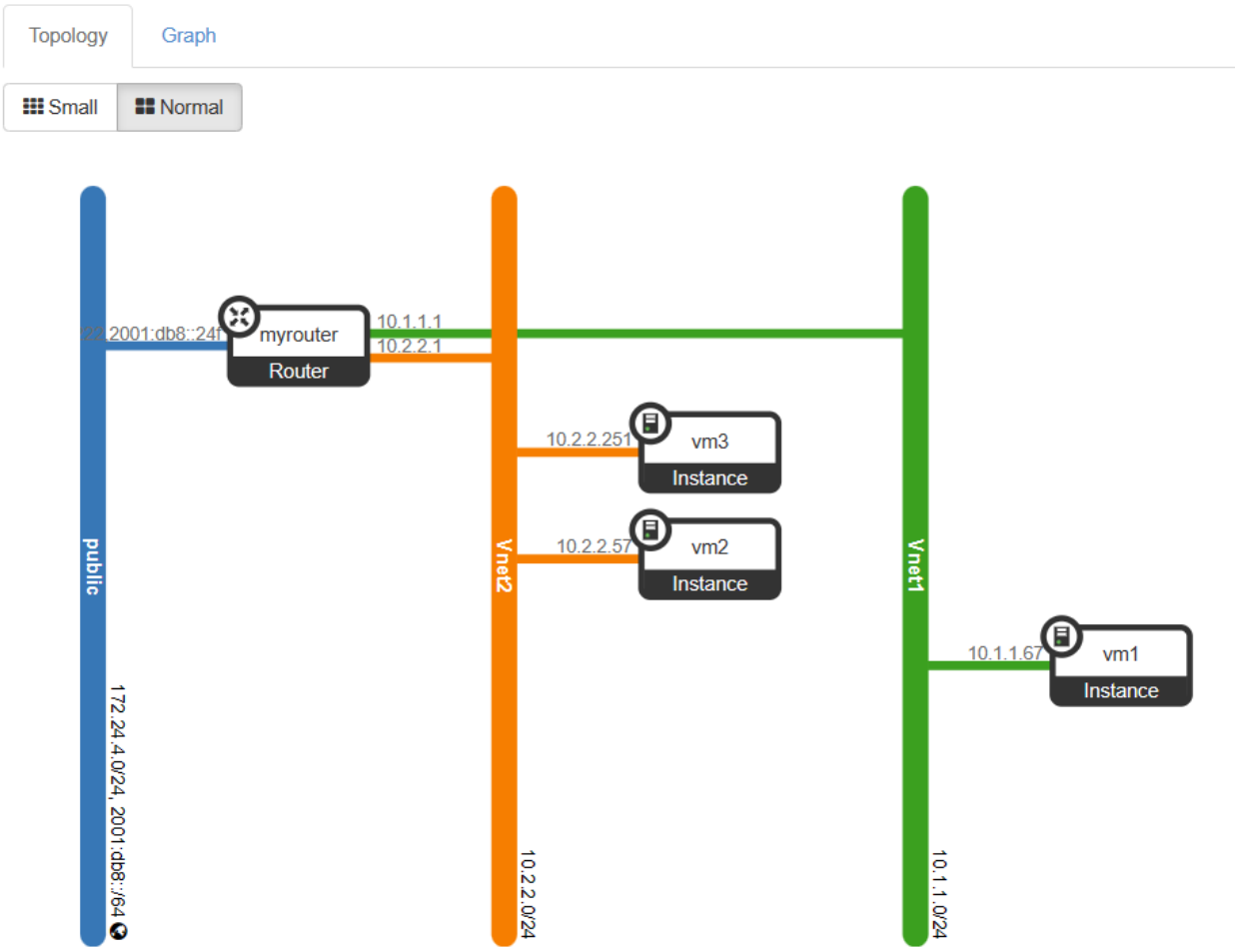
- BGP
- Hypervisor/Orchestrator (such as OpenStack)
- Containers
- SDN Controller
- Hardware server
- Service-chain

Objectives: Virtualized Network Automation

Create an application that meets the following functionality (each objective must be a separate Python module in your code i.e. your main .py file should import the different modules you write):

1) Automate the creation of multiple virtual networks (VNs) within the hypervisor and their connection to the public network.

Network Topology



2) Automate the creation of multiple VMs within the hypervisor-
   a) Both single tenant (same VN) and multi-tenant (different VNs).
   b) All VMs should be accessible from the host server and be able to access the Internet.

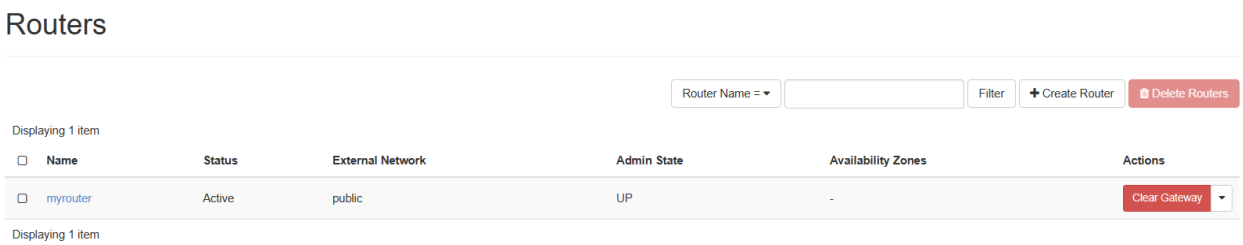
Networks

Name = | Filter | + Create Network | Delete Networks

Displaying 3 items

Name	Subnets Associated	Shared	External	Status	Admin State	Availability Zones	Actions
public	**public-subnet** 172.24.4.0/24 **ipv6-public-subnet** 2001:db8::/64	Yes	Yes	Active	UP	-	Edit Network
Vnet2	**subnet23** 10.2.2.0/24	Yes	No	Active	UP	-	Edit Network
Vnet1	**subnet1** 10.1.1.0/24	Yes	No	Active	UP	-	Edit Network

Displaying 3 items



VM1 to Internet

```
mininet@mininet-ofm:~$ ping 8.8.8.8 -c 3
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=54 time=3.79 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=54 time=5.26 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=54 time=2.49 ms

--- 8.8.8.8 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 2.495/3.852/5.264/1.132 ms
mininet@mininet-ofm:~$
```

Host to VM

```
prajwal@prajwal-PowerEdge-R430:~/Lab9$ ping 172.24.4.107 -c 3
PING 172.24.4.107 (172.24.4.107) 56(84) bytes of data.
64 bytes from 172.24.4.107: icmp_seq=1 ttl=63 time=2.29 ms
64 bytes from 172.24.4.107: icmp_seq=2 ttl=63 time=1.03 ms
64 bytes from 172.24.4.107: icmp_seq=3 ttl=63 time=0.281 ms

--- 172.24.4.107 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 0.281/1.197/2.286/0.827 ms
prajwal@prajwal-PowerEdge-R430:~/Lab9$
```

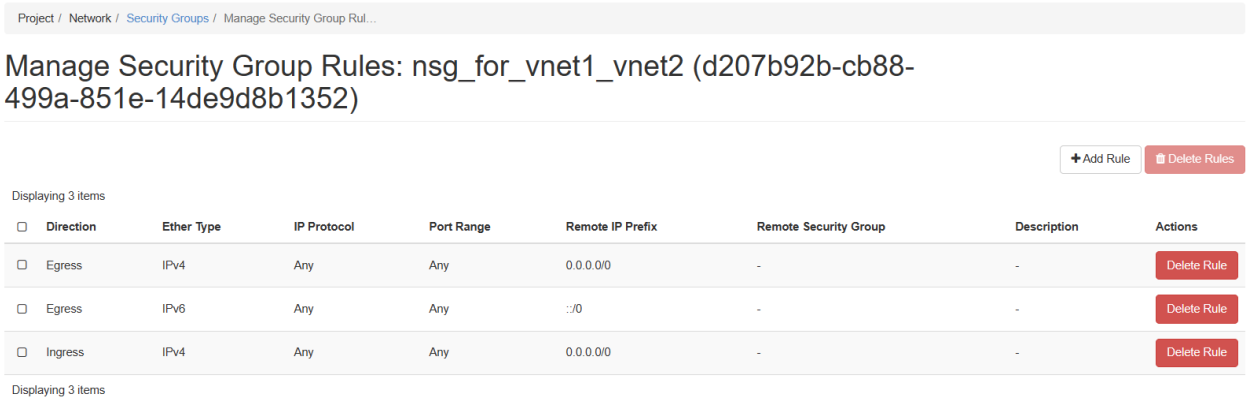
Inter Vnet communication.

```
mininet@mininet-ofm:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1442 qdisc pfifo_fast state UP group default qlen 1000
    link/ether fa:16:3e:b3:47:0c brd ff:ff:ff:ff:ff:ff
    inet 10.1.1.67/24 brd 10.1.1.255 scope global eth0
       valid_lft forever preferred_lft forever
mininet@mininet-ofm:~$
mininet@mininet-ofm:~$ ping 10.2.2.251 -c 3
PING 10.2.2.251 (10.2.2.251) 56(84) bytes of data.
64 bytes from 10.2.2.251: icmp_seq=1 ttl=63 time=2.65 ms
64 bytes from 10.2.2.251: icmp_seq=2 ttl=63 time=1.29 ms
64 bytes from 10.2.2.251: icmp_seq=3 ttl=63 time=0.404 ms

--- 10.2.2.251 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 0.404/1.452/2.657/0.926 ms
mininet@mininet-ofm:~$
```

Intra-Vnet communication

```
mininet@mininet-ofm:~$ ping 10.2.2.251 -c 3
PING 10.2.2.251 (10.2.2.251) 56(84) bytes of data.
64 bytes from 10.2.2.251: icmp_seq=1 ttl=64 time=1.24 ms
64 bytes from 10.2.2.251: icmp_seq=2 ttl=64 time=1.98 ms
64 bytes from 10.2.2.251: icmp_seq=3 ttl=64 time=0.381 ms

--- 10.2.2.251 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 0.381/1.205/1.985/0.655 ms
mininet@mininet-ofm:~$ _
```

3) Automate the security groups and port security configuration to make intra-VN and inter-VN communication possible.



Final code

```
prajwal@prajwal-PowerEdge-R430:~/Lab9/Final1_Lab9$ python3 main1.py
Executing createVnet.py
Creating network: Vnet1
Creating subnet: subnet1 for network: Vnet1
Creating network: Vnet2
Creating subnet: subnet2 for network: Vnet2
Executing createInstance.py
Creating instance: VM1 in network: Vnet1
Creating instance: VM2 in network: Vnet2
Creating instance: VM3 in network: Vnet2
Executing nsg.py
Creating security group rule for security group: efe652cd-dd13-44a8-beb0-027a568cc8a4
```

4) Automate spinning up and configuring a Quagga/FRR BGP router as a Docker container.

   a) Automate its BGP configuration to peer with the SDN controller in the next objective.

```
mininet@mininet-ofm:~$ sudo docker build -t frr_container -f Dockerfile_frr .
Sending build context to Docker daemon 121.3 MB
Sending build context to Docker daemon
Step 0 : FROM frrouting/frr:latest
latest: Pulling from frrouting/frr
9d122a1e7a29: Pull complete
3a3931816da5: Pull complete
e83182fe0379: Pull complete
6fd092a8445c: Pull complete
a09fad7c2ad8: Pull complete
5b6b5574e34a: Pull complete
491821560710: Pull complete
6d9c5924bc5d: Pull complete
a158d3284469: Pull complete
1ef8d24eb9dd: Pull complete
Digest: sha256:4d9e09031d0b1ec4e0698114895371a0a254672cb9432b4c6319d2f4b3feecc5
Status: Downloaded newer image for frrouting/frr:latest
 ---> 1ef8d24eb9dd
Step 1 : CMD frr -d
 ---> Running in a46164b6854c
 ---> f176649f86fd
Removing intermediate container a46164b6854c
Successfully built f176649f86fd
```

```
mininet@mininet-ofm:~$ sudo docker build -t sdn_controller_container -f Dockerfile_sdn_controller .
[sudo] password for mininet:
Sending build context to Docker daemon 121.3 MB
Sending build context to Docker daemon
Step 0 : FROM osrg/ryu
latest: Pulling from osrg/ryu
9d0f3981af9f: Pull complete
df35d9906b5f: Pull complete
f50c9170ef66: Pull complete
a3dee92ce2e9: Pull complete
de4ff88d37b1: Pull complete
469a1841be3d: Pull complete
18c60132af06: Pull complete
901926960982: Pull complete
904c7e330a1d: Pull complete
Digest: sha256:4ce68bbb44ac000268cd1f6ff15a26ff2fbbe2bf63decec2897df0b958481b16
Status: Downloaded newer image for osrg/ryu:latest
 ---> 904c7e330a1d
Step 1 : CMD ryu-manager
 ---> Running in 697894d93eba
 ---> 045b8ba03a46
Removing intermediate container 697894d93eba
Successfully built 045b8ba03a46
mininet@mininet-ofm:~$
```

5) Automate spinning up and configuring an SDN controller as another Docker container.
    a) Automate its BGP speaker configuration to peer with Quagga/FRR.

To build Docker images from these Dockerfiles, we can use the docker build command.

docker build -t frr_container -f Dockerfile_frr .

docker build -t sdn_controller_container -f Dockerfile_sdn_controller .

Run these containers using the docker run command:

docker run --name frr_container -d frr_container

docker run --name sdn_controller_container -d sdn_controller_container

Deliverable:

Create a personal GitHub page that demonstrates the required functionality.
https://github.com/klprajwal/Lab9-NVO.git