# PART 1: Service chaining

# Create networks:

Create Left VM network:

openstack network create left\_network

openstack subnet create --network left\_network --subnet-range 10.10.10.0/24 VM\_left\_network

Create Right VM network:

openstack network create right\_network

openstack subnet create --network right\_network --subnet-range 2.2.2.0/24 VM\_right\_network

# Create Fixed IPs:

Create Left side fixed Ips:

openstack port create --network left\_network --fixed-ip subnet=VM\_left\_network,ip-address=10.10.10.1 RTR\_port\_left

openstack port create --network left\_network --fixed-ip subnet=VM\_left\_network,ip-address=10.10.10.10 NAT\_port\_left

openstack port create --network left\_network --fixed-ip subnet=VM\_left\_network,ip-address=10.10.10.5 VM\_port\_left

Create Right side fixed Ips:

openstack port create --network right\_network --fixed-ip subnet=VM\_right\_network,ip-address=2.2.2.1 RTR\_port\_right

openstack port create --network right\_network --fixed-ip subnet=VM\_right\_network,ip-address=2.2.2.10 NAT\_port\_right

openstack port create --network right\_network --fixed-ip subnet=VM\_right\_network,ip-address=2.2.2.5 VM\_port\_right

# Create Router:

openstack router create RTR

openstack router add port RTR RTR\_port\_left

openstack router add port RTR RTR\_port\_right

openstack router set --external-gateway public --enable-snat RTR

# Create VMs:

Create Left VM:

openstack server create --flavor m1.tiny --image cirros-0.6.3-x86\_64-disk \

--port VM\_port\_left VM\_left

Create Right VM:

openstack server create --flavor m1.tiny --image cirros-0.6.3-x86\_64-disk \

--port VM\_port\_right VM\_right

Create NAT\_VM:

First, download the specific image: [**debian-10-openstack-amd64.qcow2**](https://cdimage.debian.org/images/cloud/OpenStack/current-10/debian-10-openstack-amd64.qcow2) at this [link](https://cdimage.debian.org/images/cloud/OpenStack/current-10/)

Then to create the image in Openstack go to **Project 🡪 Images 🡪 Create Image**

Upload the file you just downloaded, name it, and chose the format as **QCOW2.**

Then run this command to create the instance:

openstack server create --flavor m1.small --image Debian-10-Openstack --port NAT\_port\_left --port NAT\_port\_right NAT

# Add Floating IPs to VMs

openstack server add floating ip <instance> <floating\_ip>

# Allow address pairs & Disable Network Wide Port Security:

openstack port set --allowed-address ip-address=2.2.2.0/24 NAT\_port\_left

openstack port set --allowed-address ip-address=10.10.10.0/24 NAT\_port\_right

openstack network set --disable-port-security left\_network

openstack network set --disable-port-security right\_network

# CLI Configurations for connectivity:

**VM\_Left:**

sudo ip route add default via 10.10.10.10

**VM\_Right:**

sudo ip route add default via 2.2.2.10

**NAT\_VM:**

# NAT traffic from VM1 (10.10.10.5) going to VM3 so it appears from 2.2.2.10

iptables -t nat -A POSTROUTING -o eth1 -s 10.10.10.0/24 -j SNAT --to-source 2.2.2.10

# NAT traffic from VM3 (2.2.2.5) going to VM1 so it appears from 10.10.10.10

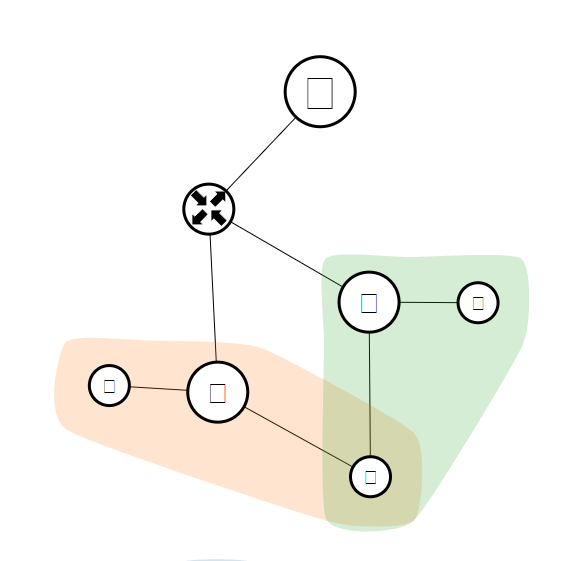
iptables -t nat -A POSTROUTING -o eth0 -s 2.2.2.0/24 -j SNAT --to-source 10.10.10.10

# Allow forwarding between eth0 (VM1) and eth1 (VM3)

iptables -A FORWARD -i eth0 -o eth1 -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT

iptables -A FORWARD -i eth1 -o eth0 -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT

# Network Topology:



# Proof of connectivity:

VM\_left 🡪VM\_right ping and traceroute:

A screenshot of a computer program

AI-generated content may be incorrect.

Tcpdump on VM\_right:

A screenshot of a computer screen

AI-generated content may be incorrect.

# PART2: Transparent Service Chaining:

# Create L2FW\_VM:

We keep everything the same, but instead we do port pairs and chaining to connect this L2 firewall to VM\_left, first let’s set up L2FW\_VM first:

openstack port create --network left\_network --fixed-ip subnet=VM\_left\_network,ip-address=10.10.10.100 L2\_port\_left

openstack port create --network left\_network --fixed-ip subnet=VM\_left\_network,ip-address=10.10.10.200 L2\_port\_right

openstack server create --flavor m1.logan --image Debian-10-Openstack --port L2\_port\_left --port L2\_port\_right L2FW\_VM

# Create IP Table Rules for L2 Traffic:

# Block incoming SSH traffic from VM1 (10.10.10.5) to VM2 (10.10.10.10)

iptables -A INPUT -p tcp --dport 22 -s 10.10.10.5 -j DROP

# Block SSH forwarding from VM1 to VM3

iptables -A FORWARD -p tcp --dport 22 -s 10.10.10.5 -j DROP

# Create port pairs, chain, and SFC:

openstack sfc port pair group create --ingress VM\_port\_left --egress L2\_port\_left VMleft\_L2FW

openstack sfc port pair group create --ingress L2\_port\_right --egress NAT\_port\_left L2FW\_NAT

openstack sfc port pair group create --port-pair WMleft\_L2FW --port-pair L2FW\_NAT L2FW\_flow

# Network Topology:

A diagram of a network

AI-generated content may be incorrect.