Team:

Khantil Choksi - khchoksi	Shubhankar Reddy - skatta2

Staging Server IP			VM IP		
152.14.83.156	ece792	EcE792net!	192.168.124.15	ece792	EcE792net!

PROBLEM 1:

	500 Mbps	1000 Mbps	5000 Mbps
CPU	1 vCPU	1 vCPU (2 for the AX tech stack)	1/2/8 vCPU (for IP base, Security and AppX)
Memory	4 GB	4 GB	4 GB

a) Networking: BGP, OSPF, EIGRP, Routing Information Protocol (RIP), Intermediate System-to-Intermediate System (IS-IS), IPv6, GRE, VRF-Lite, NTP
b) Security: ZBFW, IPsec VPN, Easy VPN, DMVPN, FlexVPN
c) Management: Cisco IOS XE CLI, SSH, Flexible NetFlow, SNMP, EEM, and NETCONF

1 year annual cost of CSR 1000V (AX stack) on aws = $ 3,723.00 (Software cost)

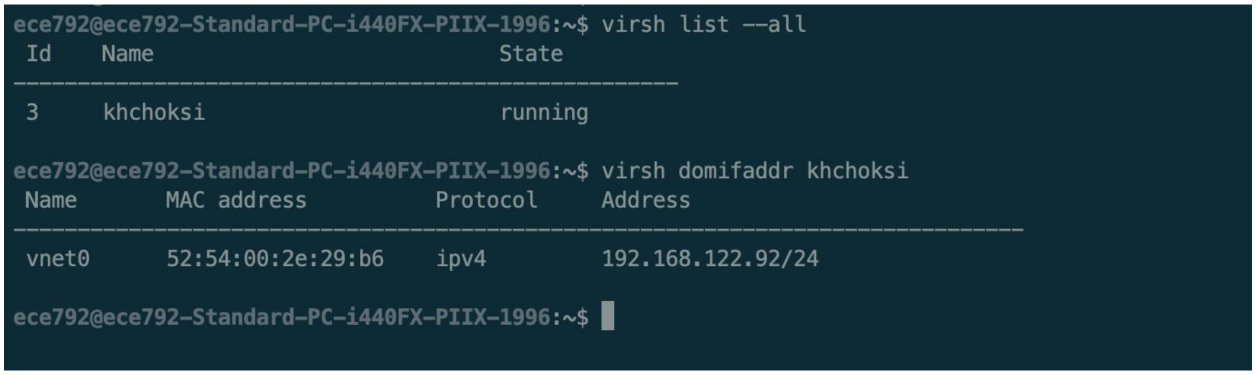
1 year of c4.large EC2 instance cost = $515.00 (C4.large, yearly cost)

Cost of running two 1000 Mbps CSR in AWS = (3,723 + 515.00) * 2 = $8,476 USD

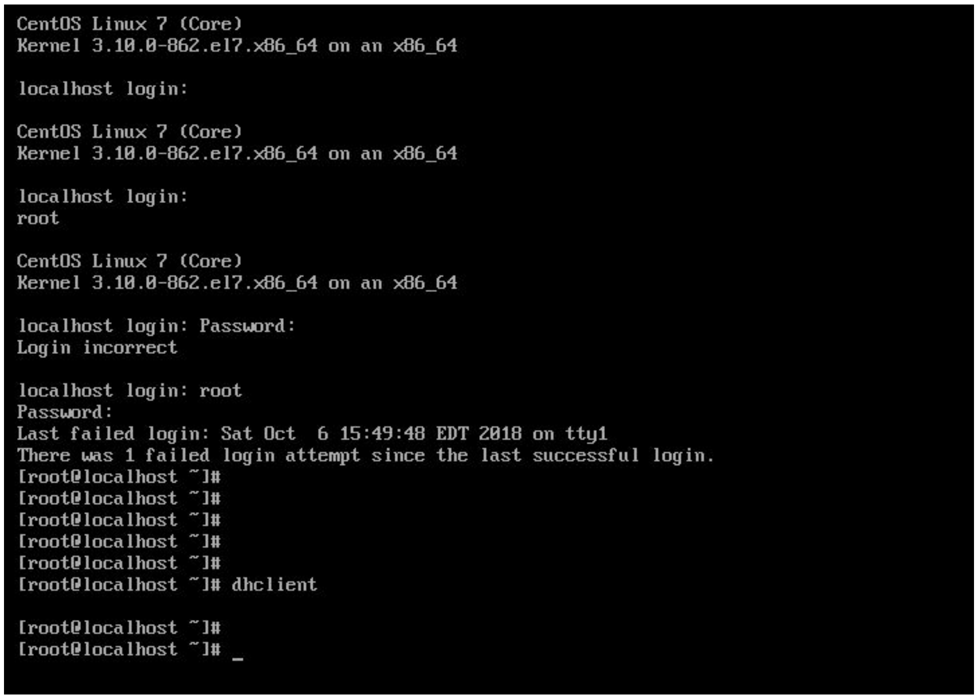
https://aws.amazon.com/marketplace/pp/B00OCG4OAA?qid=1539727240910&sr=0-1&ref_=srh_res_product_title&cl_spe=T1
https://aws.amazon.com/ec2/pricing/reserved-instances/pricing/

PROBLEM 2:

```
sudo virt-install -n khchoksi -r 2048 --vcpu=4 --cpu host --disk
path=/var/lib/libvirt/images/khchoksi.img,size=10 --network network=khchoksivm -c
/home/ece792/iso/CentOS-7-x86_64-Minimal-1804.iso -v
```



```
ece792@ece792-Standard-PC-i440FX-PIIX-1996:~$ virsh list --all
 Id    Name                           State
----------------------------------------------------
 3     khchoksi                       running

ece792@ece792-Standard-PC-i440FX-PIIX-1996:~$ virsh domifaddr khchoksi
 Name       MAC address          Protocol     Address
-------------------------------------------------------------------------------
 vnet0      52:54:00:2e:29:b6    ipv4         192.168.122.92/24

ece792@ece792-Standard-PC-i440FX-PIIX-1996:~$
```



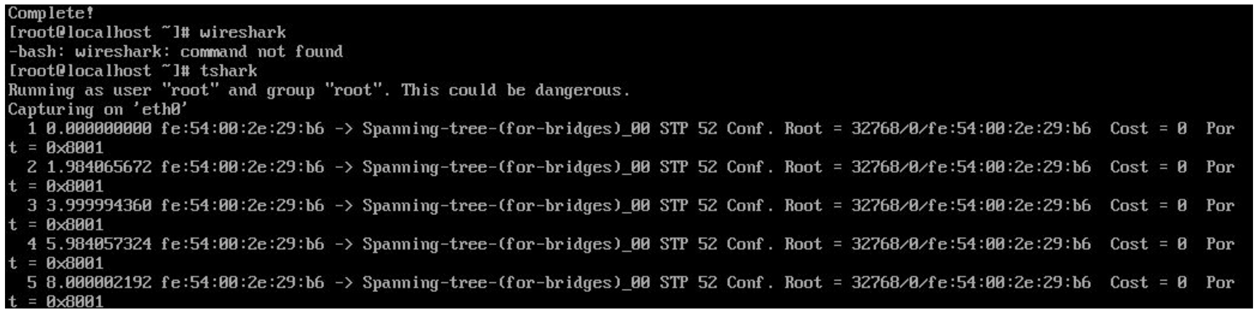
```
CentOS Linux 7 (Core)
Kernel 3.10.0-862.el7.x86_64 on an x86_64

localhost login:

CentOS Linux 7 (Core)
Kernel 3.10.0-862.el7.x86_64 on an x86_64

localhost login:
root

CentOS Linux 7 (Core)
Kernel 3.10.0-862.el7.x86_64 on an x86_64

localhost login: Password:
Login incorrect

localhost login: root
Password:
Last failed login: Sat Oct  6 15:49:48 EDT 2018 on tty1
There was 1 failed login attempt since the last successful login.
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]# dhclient

[root@localhost ~]#
[root@localhost ~]# _
```

To install required applications:

```
$ yum install iperf3

$ yum install wireshark
```

(i)

	VM's NIC	hypervisor NIC
IP Address	192.168.122.92/24	192.168.124.15
MAC Address	52:54:00:2e:29:b6	52:54:00:2f:dd:ba



```
Complete!
[root@localhost ~]# wireshark
-bash: wireshark: command not found
[root@localhost ~]# tshark
Running as user "root" and group "root". This could be dangerous.
Capturing on 'eth0'
  1 0.000000000 fe:54:00:2e:29:b6 -> Spanning-tree-(for-bridges)_00 STP 52 Conf. Root = 32768/0/fe:54:00:2e:29:b6  Cost = 0  Por
t = 0x8001
  2 1.984065672 fe:54:00:2e:29:b6 -> Spanning-tree-(for-bridges)_00 STP 52 Conf. Root = 32768/0/fe:54:00:2e:29:b6  Cost = 0  Por
t = 0x8001
  3 3.999994360 fe:54:00:2e:29:b6 -> Spanning-tree-(for-bridges)_00 STP 52 Conf. Root = 32768/0/fe:54:00:2e:29:b6  Cost = 0  Por
t = 0x8001
  4 5.984057324 fe:54:00:2e:29:b6 -> Spanning-tree-(for-bridges)_00 STP 52 Conf. Root = 32768/0/fe:54:00:2e:29:b6  Cost = 0  Por
t = 0x8001
  5 8.000002192 fe:54:00:2e:29:b6 -> Spanning-tree-(for-bridges)_00 STP 52 Conf. Root = 32768/0/fe:54:00:2e:29:b6  Cost = 0  Por
t = 0x8001
```

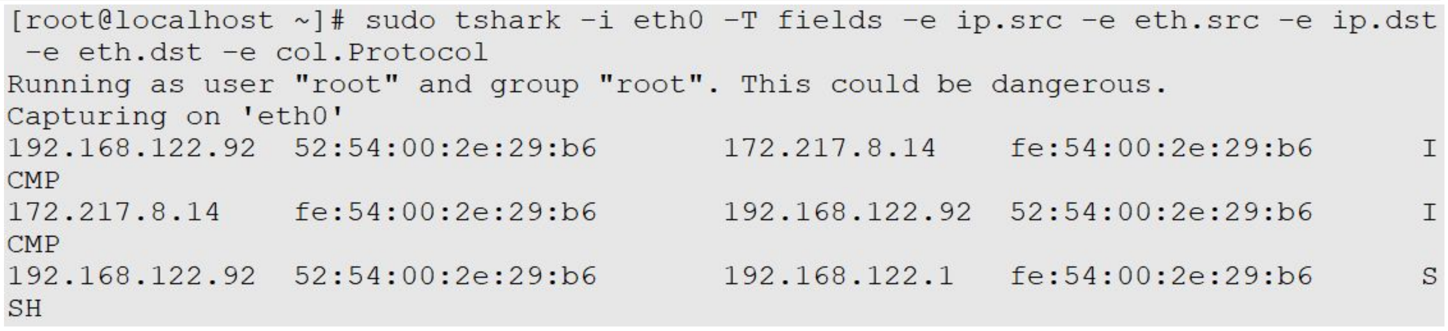
(ii)

	output interface of VM
srcIP	192.168.122.92
destIP	172.217.8.14
srcMAC	52:54:00:2e:29:b6
destMAC	fe:54:00:2e:29:b6

	output interface of hypervisor
srcIP	192.168.122.15
destIP	172.217.8.14
srcMAC	52:54:00:2f:dd:aa
destMAC	52:54:00:3b:8a:fe

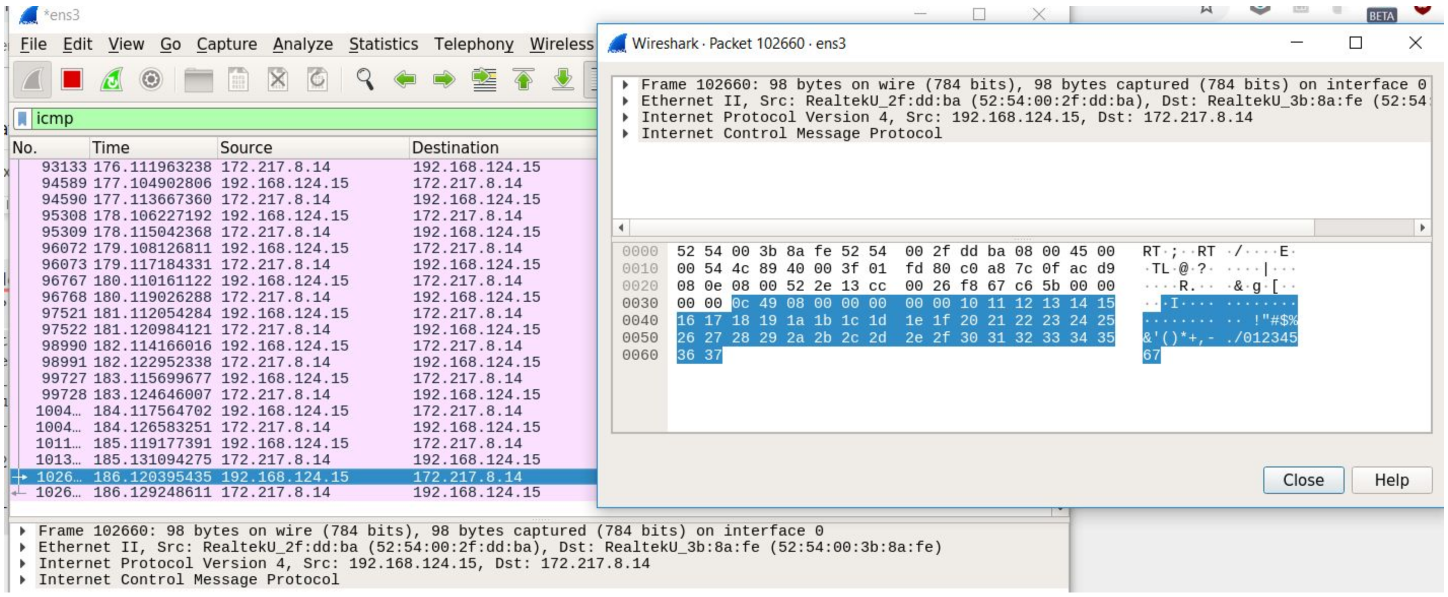
- Packet going out of the VM

```
$ sudo tshark -i eth0 -T fields -e ip.src -e eth.src -e ip.dst -e eth.dst -e
col.Protocol
```



```
[root@localhost ~]# sudo tshark -i eth0 -T fields -e ip.src -e eth.src -e ip.dst
 -e eth.dst -e col.Protocol
Running as user "root" and group "root". This could be dangerous.
Capturing on 'eth0'
192.168.122.92  52:54:00:2e:29:b6       172.217.8.14    fe:54:00:2e:29:b6       I
CMP
172.217.8.14    fe:54:00:2e:29:b6       192.168.122.92  52:54:00:2e:29:b6       I
CMP
192.168.122.92  52:54:00:2e:29:b6       192.168.122.1   fe:54:00:2e:29:b6       S
SH
```

- Packet going out of the hypervisor using wireshark


*ens3
File Edit View Go Capture Analyze Statistics Telephony Wireless
icmp
Wireshark · Packet 102660 · ens3
Close
Help

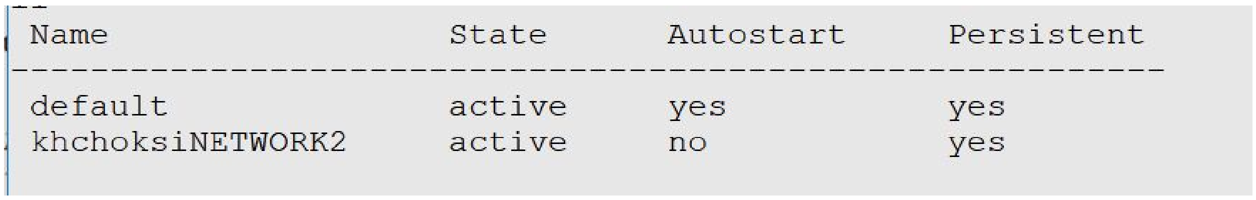
```
No.    Time          Source          Destination
 93133 176.111963238 172.217.8.14    192.168.124.15
 94589 177.104902806 192.168.124.15  172.217.8.14
 94590 177.113667360 172.217.8.14    192.168.124.15
 95308 178.106227192 192.168.124.15  172.217.8.14
 95309 178.115042368 172.217.8.14    192.168.124.15
 96072 179.108126811 192.168.124.15  172.217.8.14
 96073 179.117184331 172.217.8.14    192.168.124.15
 96767 180.110161122 192.168.124.15  172.217.8.14
 96768 180.119026288 172.217.8.14    192.168.124.15
 97521 181.112054284 192.168.124.15  172.217.8.14
 97522 181.120984121 172.217.8.14    192.168.124.15
 98990 182.114166016 192.168.124.15  172.217.8.14
 98991 182.122952338 172.217.8.14    192.168.124.15
 99727 183.115699677 192.168.124.15  172.217.8.14
 99728 183.124646007 172.217.8.14    192.168.124.15
1004…  184.117564702 192.168.124.15  172.217.8.14
1004…  184.126583251 172.217.8.14    192.168.124.15
1011…  185.119177391 192.168.124.15  172.217.8.14
1013…  185.131094275 172.217.8.14    192.168.124.15
1026…  186.120395435 192.168.124.15  172.217.8.14
1026…  186.129248611 172.217.8.14    192.168.124.15

▸ Frame 102660: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface 0
▸ Ethernet II, Src: RealtekU_2f:dd:ba (52:54:00:2f:dd:ba), Dst: RealtekU_3b:8a:fe (52:54:00:3b:8a:fe)
▸ Internet Protocol Version 4, Src: 192.168.124.15, Dst: 172.217.8.14
▸ Internet Control Message Protocol
```

```
▸ Frame 102660: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface 0
▸ Ethernet II, Src: RealtekU_2f:dd:ba (52:54:00:2f:dd:ba), Dst: RealtekU_3b:8a:fe (52:54
▸ Internet Protocol Version 4, Src: 192.168.124.15, Dst: 172.217.8.14
▸ Internet Control Message Protocol

0000  52 54 00 3b 8a fe 52 54  00 2f dd ba 08 00 45 00   RT·;··RT ·/····E·
0010  00 54 4c 89 40 00 3f 01  fd 80 c0 a8 7c 0f ac d9   ·TL·@·?· ····|···
0020  08 0e 08 00 52 2e 13 cc  00 26 f8 67 c6 5b 00 00   ····R.·· ·&·g·[··
0030  00 00 0c 49 08 00 00 00  00 00 10 11 12 13 14 15   ···I···· ········
0040  16 17 18 19 1a 1b 1c 1d  1e 1f 20 21 22 23 24 25   ········ ·· !"#$%
0050  26 27 28 29 2a 2b 2c 2d  2e 2f 30 31 32 33 34 35   &'()*+,- ./012345
0060  36 37                                              67
```

These tuples are in different network. According to datapath, the interface ens3 will do NAT and encap, decap operation and changes the src IP to ens3 IP. And it will be sent out the request over ens3.

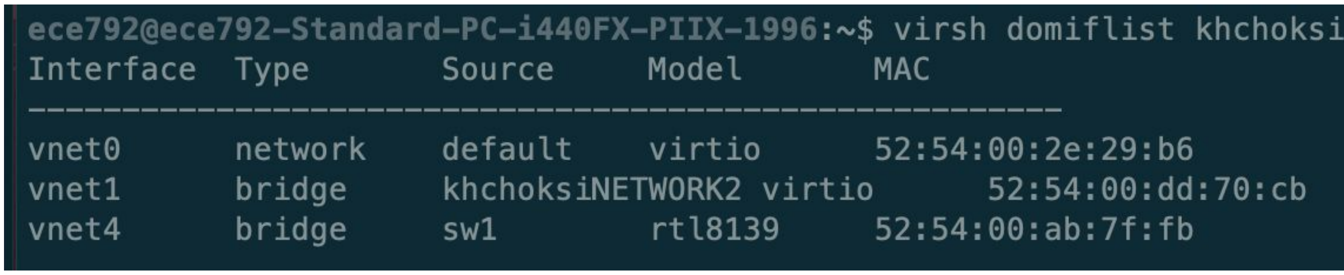
PROBLEM 3:

1. khchoksiNETWORK2.xml
    a. $ virsh net-define khchoksiNETWORK2.xml
    b. $ brctl addbr sw1
    c. virsh net- start khchoksiNETWORK2



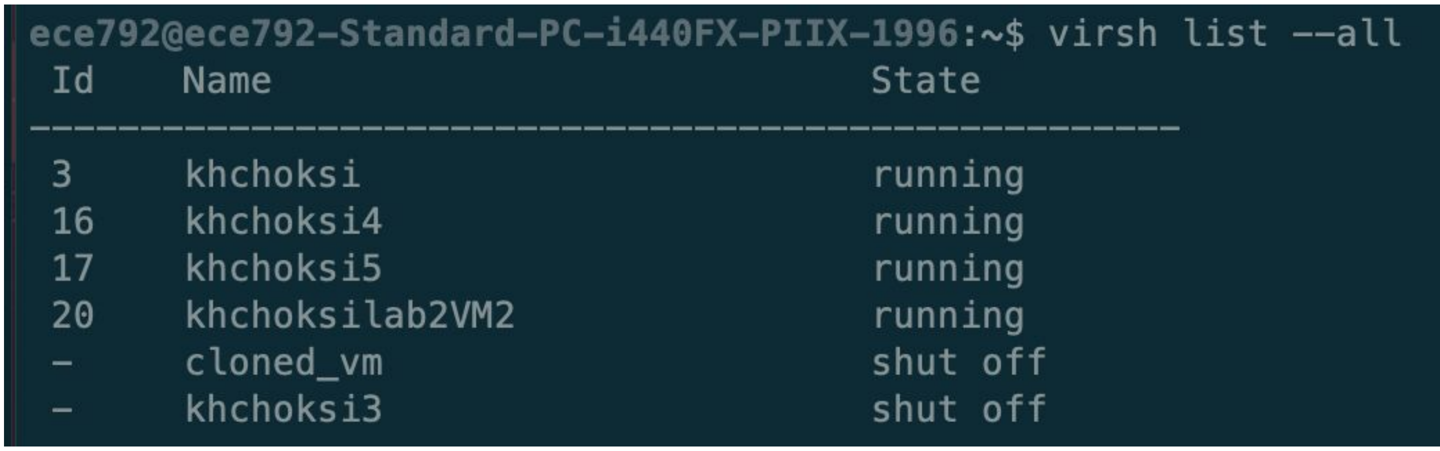
```
 Name                 State      Autostart     Persistent
----------------------------------------------------------
 default              active     yes           yes
 khchoksiNETWORK2     active     no            yes
```

2.
    a. virsh attach-interface --domain khchoksi --type bridge --source sw1
    b. deactivate VM: $ virsh shutdown khchoksi
    c. Add interface to xml file
    d. $ virsh define /etc/libvirt/qemu/khchoksi.xml
    e. $ virsh start khchoksi (restart VM)



```
ece792@ece792-Standard-PC-i440FX-PIIX-1996:~$ virsh domiflist khchoksi
Interface  Type       Source     Model       MAC
-------------------------------------------------------
vnet0      network    default    virtio      52:54:00:2e:29:b6
vnet1      bridge     khchoksiNETWORK2 virtio      52:54:00:dd:70:cb
vnet4      bridge     sw1        rtl8139     52:54:00:ab:7f:fb
```

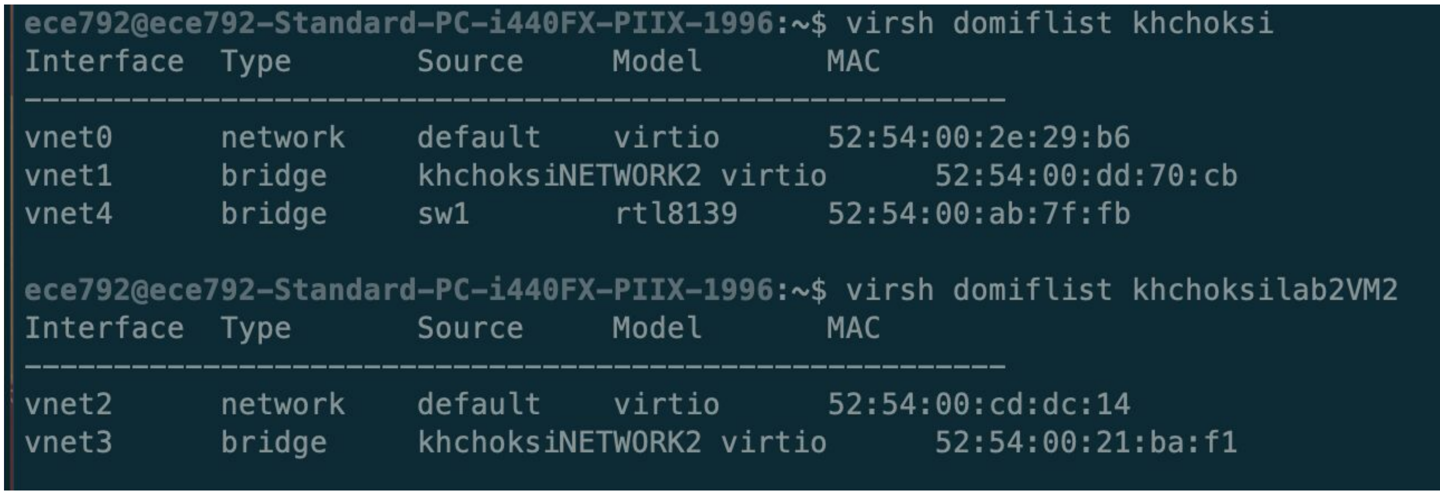
3. $ virsh suspend khchoksi
   $ virt clone --original khchoksi --name khchoksilab2VM2 --auto-clone
   $ virsh resume khchoksi
   $ virsh start khchoksilab2VM2



```
ece792@ece792-Standard-PC-i440FX-PIIX-1996:~$ virsh list --all
 Id    Name                           State
----------------------------------------------------
 3     khchoksi                       running
 16    khchoksi4                      running
 17    khchoksi5                      running
 20    khchoksilab2VM2                running
 -     cloned_vm                      shut off
 -     khchoksi3                      shut off
```

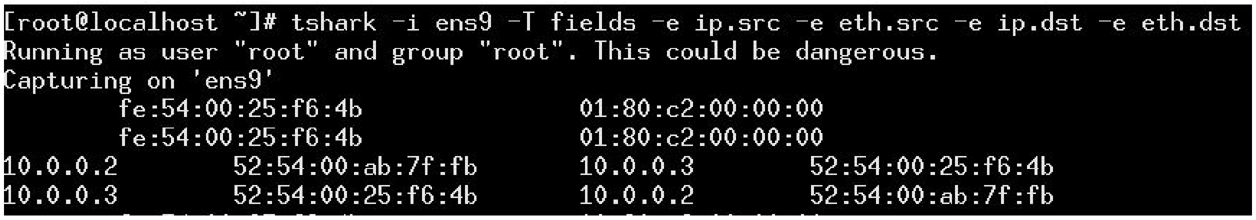
4. Assign ip to eth1 : ifconfig ens9 10.0.0.1/24

VM	interface	network	MAC Address	IP Addresses
khchoksi	vnet0	default	52:54:00:2e:29:b6	192.168.122.92
	ens9	khchoksiNETWORK2	52:54:00:ab:7f:fb	10.0.0.2
khchoksilab2VM2	vnet2	default	52:54:00:cd:dc:14	192.168.122.51
	ens9	khchoksiNETWORK2	52:54:00:25:f6:4b	10.0.0.3



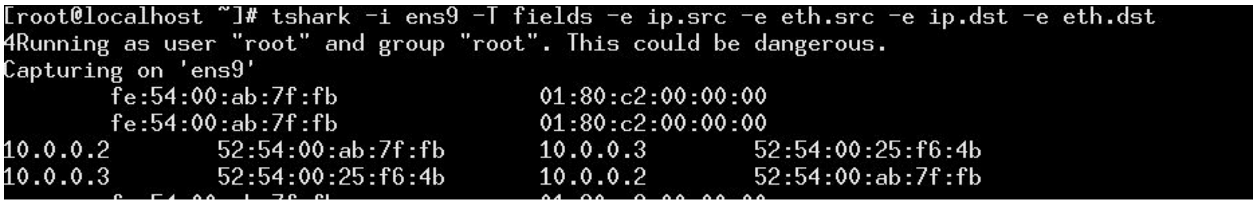
```
ece792@ece792-Standard-PC-i440FX-PIIX-1996:~$ virsh domiflist khchoksi
Interface  Type       Source     Model       MAC
-------------------------------------------------------
vnet0      network    default    virtio      52:54:00:2e:29:b6
vnet1      bridge     khchoksiNETWORK2 virtio      52:54:00:dd:70:cb
vnet4      bridge     sw1        rtl8139     52:54:00:ab:7f:fb

ece792@ece792-Standard-PC-i440FX-PIIX-1996:~$ virsh domiflist khchoksilab2VM2
Interface  Type       Source     Model       MAC
-------------------------------------------------------
vnet2      network    default    virtio      52:54:00:cd:dc:14
vnet3      bridge     khchoksiNETWORK2 virtio      52:54:00:21:ba:f1
```

5. Tuple fields on 10.0.0.2 are:



```
[root@localhost ~]# tshark -i ens9 -T fields -e ip.src -e eth.src -e ip.dst -e eth.dst
Running as user "root" and group "root". This could be dangerous.
Capturing on 'ens9'
        fe:54:00:25:f6:4b               01:80:c2:00:00:00
        fe:54:00:25:f6:4b               01:80:c2:00:00:00
10.0.0.2        52:54:00:ab:7f:fb       10.0.0.3        52:54:00:25:f6:4b
10.0.0.3        52:54:00:25:f6:4b       10.0.0.2        52:54:00:ab:7f:fb
```

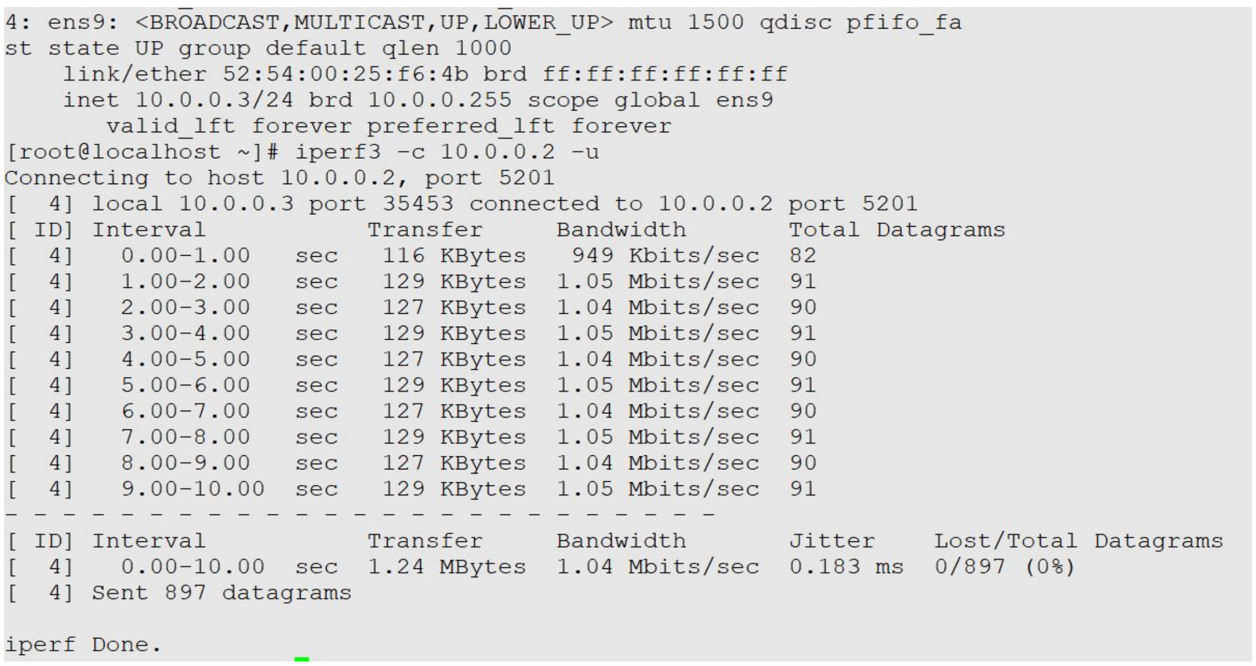
Tuple fields on 10.0.0.3 are:



```
[root@localhost ~]# tshark -i ens9 -T fields -e ip.src -e eth.src -e ip.dst -e eth.dst
4Running as user "root" and group "root". This could be dangerous.
Capturing on 'ens9'
        fe:54:00:ab:7f:fb               01:80:c2:00:00:00
        fe:54:00:ab:7f:fb               01:80:c2:00:00:00
10.0.0.2        52:54:00:ab:7f:fb       10.0.0.3        52:54:00:25:f6:4b
10.0.0.3        52:54:00:25:f6:4b       10.0.0.2        52:54:00:ab:7f:fb
```

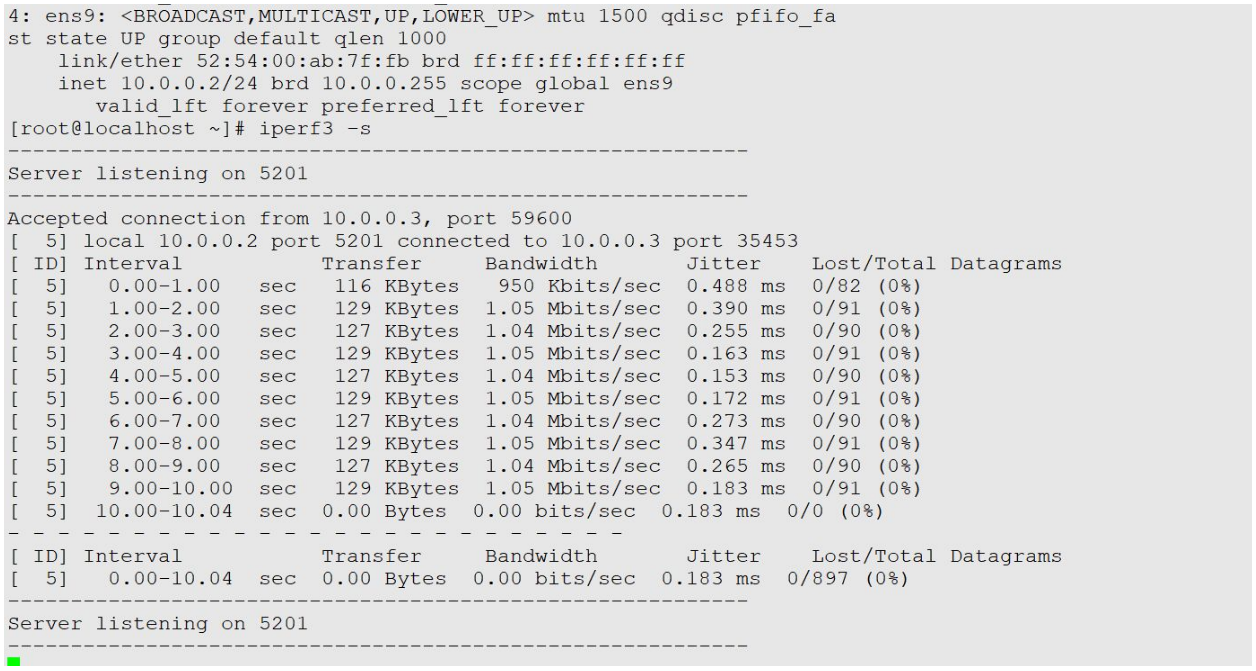
The tuple fields do not change as the packets are forwarded over L2 and no encap/decap takes place.

6. On khchoksilab2VM2



```
4: ens9: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fa
st state UP group default qlen 1000
    link/ether 52:54:00:25:f6:4b brd ff:ff:ff:ff:ff:ff
    inet 10.0.0.3/24 brd 10.0.0.255 scope global ens9
       valid_lft forever preferred_lft forever
[root@localhost ~]# iperf3 -c 10.0.0.2 -u
Connecting to host 10.0.0.2, port 5201
[  4] local 10.0.0.3 port 35453 connected to 10.0.0.2 port 5201
[ ID] Interval           Transfer     Bandwidth       Total Datagrams
[  4]   0.00-1.00   sec   116 KBytes   949 Kbits/sec  82
[  4]   1.00-2.00   sec   129 KBytes  1.05 Mbits/sec  91
[  4]   2.00-3.00   sec   127 KBytes  1.04 Mbits/sec  90
[  4]   3.00-4.00   sec   129 KBytes  1.05 Mbits/sec  91
[  4]   4.00-5.00   sec   127 KBytes  1.04 Mbits/sec  90
[  4]   5.00-6.00   sec   129 KBytes  1.05 Mbits/sec  91
[  4]   6.00-7.00   sec   127 KBytes  1.04 Mbits/sec  90
[  4]   7.00-8.00   sec   129 KBytes  1.05 Mbits/sec  91
[  4]   8.00-9.00   sec   127 KBytes  1.04 Mbits/sec  90
[  4]   9.00-10.00  sec   129 KBytes  1.05 Mbits/sec  91
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Jitter    Lost/Total Datagrams
[  4]   0.00-10.00  sec  1.24 MBytes  1.04 Mbits/sec  0.183 ms  0/897 (0%)
[  4] Sent 897 datagrams

iperf Done.
```

On khchoksi



```
4: ens9: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fa
st state UP group default qlen 1000
    link/ether 52:54:00:ab:7f:fb brd ff:ff:ff:ff:ff:ff
    inet 10.0.0.2/24 brd 10.0.0.255 scope global ens9
       valid_lft forever preferred_lft forever
[root@localhost ~]# iperf3 -s
-----------------------------------------------------------
Server listening on 5201
-----------------------------------------------------------
Accepted connection from 10.0.0.3, port 59600
[  5] local 10.0.0.2 port 5201 connected to 10.0.0.3 port 35453
[ ID] Interval           Transfer     Bandwidth       Jitter    Lost/Total Datagrams
[  5]   0.00-1.00   sec   116 KBytes   950 Kbits/sec  0.488 ms  0/82 (0%)
[  5]   1.00-2.00   sec   129 KBytes  1.05 Mbits/sec  0.390 ms  0/91 (0%)
[  5]   2.00-3.00   sec   127 KBytes  1.04 Mbits/sec  0.255 ms  0/90 (0%)
[  5]   3.00-4.00   sec   129 KBytes  1.05 Mbits/sec  0.163 ms  0/91 (0%)
[  5]   4.00-5.00   sec   127 KBytes  1.04 Mbits/sec  0.153 ms  0/90 (0%)
[  5]   5.00-6.00   sec   129 KBytes  1.05 Mbits/sec  0.172 ms  0/91 (0%)
[  5]   6.00-7.00   sec   127 KBytes  1.04 Mbits/sec  0.273 ms  0/90 (0%)
[  5]   7.00-8.00   sec   129 KBytes  1.05 Mbits/sec  0.347 ms  0/91 (0%)
[  5]   8.00-9.00   sec   127 KBytes  1.04 Mbits/sec  0.265 ms  0/90 (0%)
[  5]   9.00-10.00  sec   129 KBytes  1.05 Mbits/sec  0.183 ms  0/91 (0%)
[  5]  10.00-10.04  sec  0.00 Bytes  0.00 bits/sec  0.183 ms  0/0 (0%)
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Jitter    Lost/Total Datagrams
[  5]   0.00-10.04  sec  0.00 Bytes  0.00 bits/sec  0.183 ms  0/897 (0%)
-----------------------------------------------------------
Server listening on 5201
-----------------------------------------------------------
```

The maximum throughput achieved is 1.04 Mbits/sec.
From the top command we noticed that CPU usage is not consumed and CPU is idle and I/O is also being affected. So by elimination, we can think that it is the memory.
We can also deduce that, both the VMs will have virtual memory from host and will have memory(RAM) crunch at the time of transferring packets. So, we think that that memory could be the bottleneck.

PROBLEM 4:

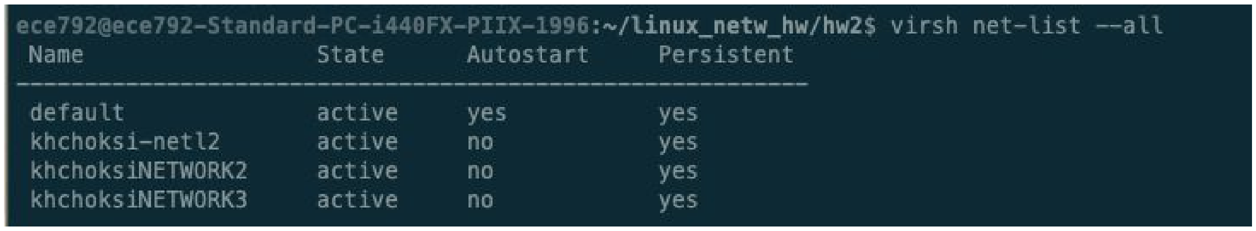
(Note: All the code and output is store in q4 folder submitted in zip)

README

Prerequisite: Install ansible on host machine

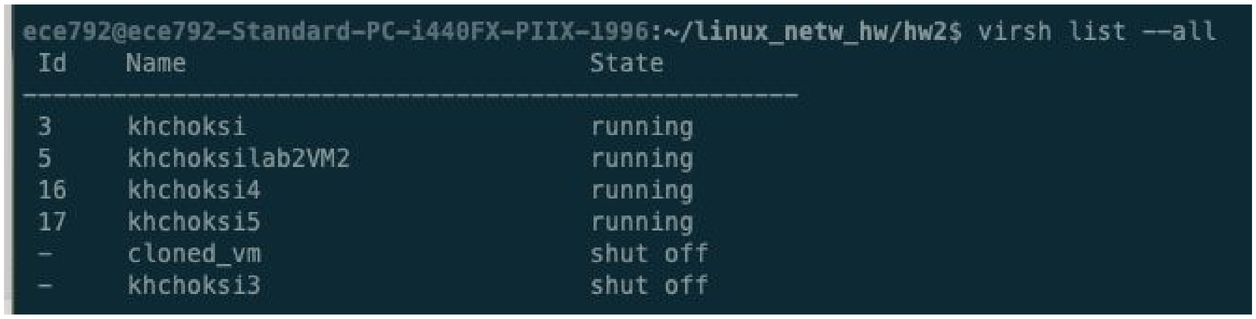
```
$ sudo apt-add-repository ppa:ansible/ansible
$ sudo apt-get update
$ sudo apt-get install ansible
$ ansible --version
     ansible 2.7.0
```

1. **Ansible Playbook: q4_1.yml**
    a. Create pure L2 network and named khchoksi-netl2

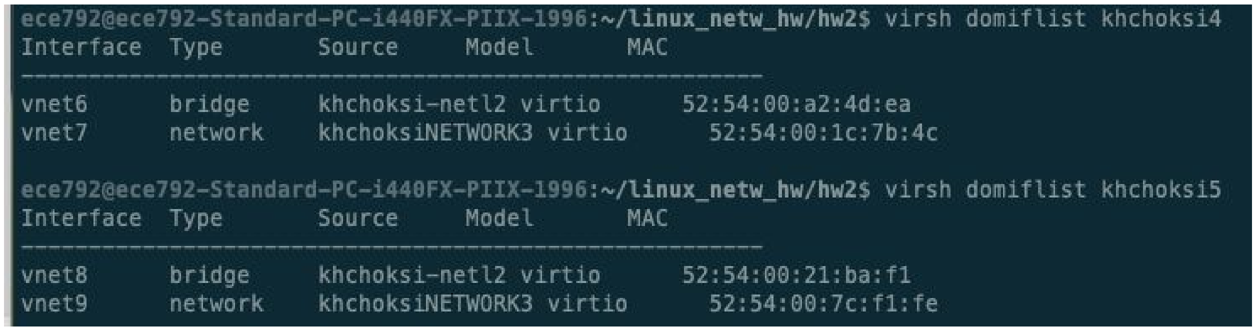


```
ece792@ece792-Standard-PC-i440FX-PIIX-1996:~/linux_netw_hw/hw2$ virsh net-list --all
 Name                 State      Autostart     Persistent
----------------------------------------------------------
 default              active     yes           yes
 khchoksi-netl2       active     no            yes
 khchoksiNETWORK2     active     no            yes
 khchoksiNETWORK3     active     no            yes
```

    b. Create two VMs (khchoksi4, khchoksi5) and connect with this L2 Network
    Please make sure to run the ansible script with X11 forwarding.
    - The script will wait for configuration of VM using GUI (which can't be automated)



```
ece792@ece792-Standard-PC-i440FX-PIIX-1996:~/linux_netw_hw/hw2$ virsh list --all
 Id    Name                           State
----------------------------------------------------
 3     khchoksi                       running
 5     khchoksilab2VM2                running
 16    khchoksi4                      running
 17    khchoksi5                      running
 -     cloned_vm                      shut off
 -     khchoksi3                      shut off
```



```
ece792@ece792-Standard-PC-i440FX-PIIX-1996:~/linux_netw_hw/hw2$ virsh domiflist khchoksi4
Interface  Type       Source     Model       MAC
-------------------------------------------------------
vnet6      bridge     khchoksi-netl2 virtio      52:54:00:a2:4d:ea
vnet7      network    khchoksiNETWORK3 virtio      52:54:00:1c:7b:4c

ece792@ece792-Standard-PC-i440FX-PIIX-1996:~/linux_netw_hw/hw2$ virsh domiflist khchoksi5
Interface  Type       Source     Model       MAC
-------------------------------------------------------
vnet8      bridge     khchoksi-netl2 virtio      52:54:00:21:ba:f1
vnet9      network    khchoksiNETWORK3 virtio      52:54:00:7c:f1:fe
```

Run playbook using following command

```
$ sudo ansible-playbook q4_1.yml
--extra-vars="/home/ece792/iso/CentOS-7-x86_64-Minimal-1804.iso"
```

Playbook: q4_1.yml:

```yaml
---
- hosts: localhost
 gather_facts: no
 vars:
   network_name: khchoksi-netl2
   bridge_name: sw2
   packages:
      - python-libvirt
      - python-lxml

   guests:
      - name: khchoksi4
        mem: 512
        vcpu: 1
        network: "{{ network_name }}"
      - name: khchoksi5
        mem: 1024
        vcpu: 2
        network: "{{ network_name }}"

   vm_disk_location: /var/lib/libvirt/images/
   iso_file_path: {{ iso_file | default('/home/ece792/iso/CentOS-7-x86_64-Minimal-1804.iso') }}

 tasks:
   # Install required packages
   - name: Install required packages for libvirt, lxml
     apt:
       name: "{{packages}}"
     become: yes

   # Define a new network
   - name: Define Virtual Network
     virt_net:
       command: define
       name: '{{ network_name }}'
       xml: '{{ lookup("template", "templates/bridge_template.xml.j2") }}'

   # Create and start a network
   - name: Create Virtual Network if not created
     virt_net:
      command: create
      name: "{{ network_name }}"
     ignore_errors: true

   # Stop a network
   # - name: Stop Virtual Network if running
   #   virt_net:
   #    command: stop
   #    name: "{{ network_name }}"
   #   ignore_errors: true # To make task idempotent

   # List available networks
   - name: List available networks
     virt_net:
       command: list_nets
```

```
  # Create New VM and will pop up UI
  - name: Create VM instance
    command: >
               virt-install -n {{ item.name }} -r {{ item.mem }} --vcpu={{ item.vcpu }} --cpu host --disk
path={{ vm_disk_location }}{{ item.name }}.img,size=5 --network network={{ network_name }} -c {{
iso_file_path }} -v
    become: yes
    with_items: "{{ guests }}"
```

Intermediate steps to setup ssh:

I. Create a new NAT bridge virbr1 so that both the newly created vms can have ips.

II. Create new network: khchoksiNETWORK3.xml

```
<network>
  <name>khchoksiNETWORK3</name>
  <uuid>eadcd6b7-c89a-43b5-9fe0-407eb0034038</uuid>
  <forward mode='nat'/>
  <bridge name='virbr1' stp='on' delay='0'/>
  <mac address='52:54:00:9f:f8:b6'/>
  <ip address='192.168.119.1' netmask='255.255.255.0'>
    <dhcp>
      <range start='192.168.119.2' end='192.168.119.254'/>
    </dhcp>
  </ip>
</network>
```

III. $ brctl addbr virbr1
$ virsh net- start khchoksiNETWORK3
Add this interface to both the VMs and restart them.

```
<interface type='network'>
      <source network='khchoksiNETWORK3'/>
      <model type='virtio'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x0a' function='0x0'/>
</interface>
```

IV. Get the ips of both the vms (if not assigned, do dhcpclient).

V. Create SSH key pairs on host machine
    $ ssh-keygen
    and then follow the command

VI. Copy public keys to both guest machines
    $ ssh-copy-id -i ./keys/vm_rsa.pub root@192.168.119.58
    $ ssh-copy-id -i ./keys/vm_rsa.pub root@192.168.119.252

VII. Create new inventory file as follows:

```
[vms]
localhost ansible_connection=local
192.168.119.58 ansible_ssh_user=root ansible_ssh_private_key_file=./keys/vm_rsa
192.168.119.252 ansible_ssh_user=root ansible_ssh_private_key_file=./keys/vm_rsa
```

2. **Ansible playbook to collect logs:**
    a. Make sure inventory file and keys are created as mentioned above.
    b. Run the playbook with total time(in minutes) as parameter given below. If time not defined, it will take default 5 minutes. Granularity is set as 1 minutes as mentioned in the description.

```
$ sudo ansible-playbook q4_2.yml -i ./inventory --extra-vars "time=7"
```

Logs will be generated at: /var/customlogs/logs.csv (Attached within zip:-> q4_2_logs.csv)
Note: Ansible playbook will create 'customlogs' directory if not present.

Playbook: q4_2.yml

```
---
- hosts: vms
 gather_facts: no
 vars:
    total_time: "{{ time | default(5) }}"    # Defined total time if not passed from command line
    granularity: 60 #in seconds
    log_file_directory: /var/customlogs
    log_file_path: "{{ log_file_directory }}/logs.csv"
  tasks:
   - name: Create logs directory if not present
     file:
       path: "{{ log_file_directory }}"
       state: directory
       mode: 0777
       owner: ece792
       group: ece792
     become: yes
     delegate_to: localhost
     run_once: true

   - name: Generate Log CSV File Header
     shell: echo "hostname, timestamp, cpu1min, cpu5min, cpu15min" >> "{{ log_file_path }}"
     delegate_to: localhost
     run_once: true
     become: yes

   - name: Generate loop sequence based on input total time parameter
     set_fact:
        loop_sequence: "{{ loop_sequence | default([]) + [item | int] }}"
     with_sequence: start=1 end={{ total_time }}

#    - name: debug_list
#      debug:
```

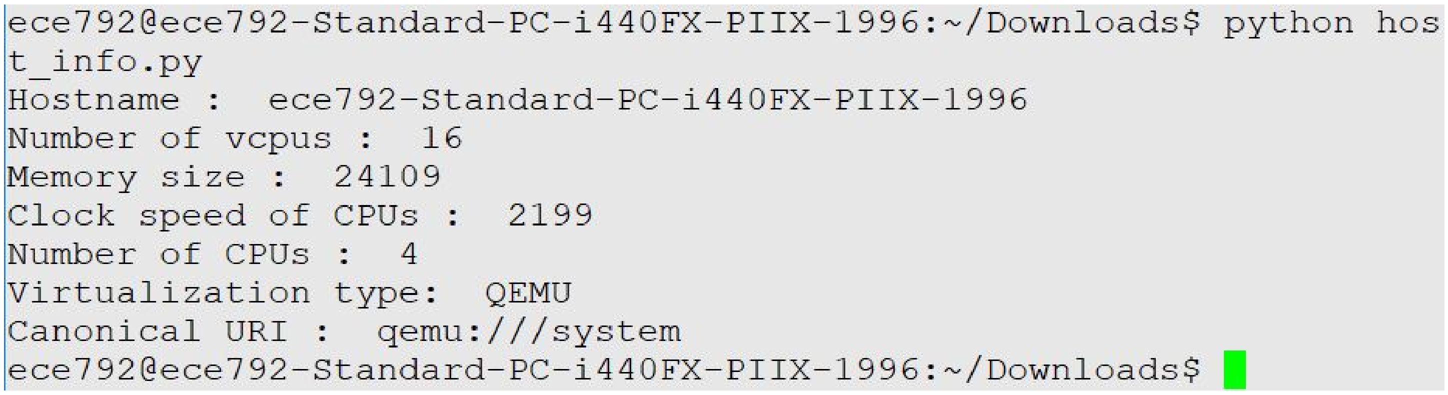
```
#         msg: "{{ loop_sequence }} "

   - name: Fetch cpu usages from host and guests, store it in output variable
     shell: "echo -n '{{hostvars[inventory_hostname]['inventory_hostname']}},' && date +%X | awk -F,
'{printf \" %s, \", $1}' && uptime | sed 's/.*load average: //' | awk -F\\, '{ printf \"%s, %s, %s\", $1,
$2, $3}'"
     register: output
     loop: "{{ loop_sequence }}"
     loop_control:
        pause: "{{ granularity }}"

#     - name: debugging
#       debug:
#         msg: "{{ item.stdout }}"
#       with_items: "{{ output.results }}"

   - name: Writing logs to csv file
     shell: |
            echo "{{ item.stdout }}" >> "{{ log_file_path }}"
     with_items: "{{ output.results }}"
     delegate_to: localhost
     become: yes
```

Problem 5:

(Note: All the code and output is store in q5 folder submitted in zip)

1) Obtaining host information:

```python
import sys
import libvirt
import random

conn = libvirt.open('qemu:///system')
if not conn:
        print "Connection failed"
        exit(1)

domainIDs = conn.listDomainsID()
if len(domainIDs) == 0:
        print "No active domains"
randomid = random.sample(domainIDs, 1)

rdom = conn.lookupByID(randomid[0])

state, maxmem, mem, cpus, cput = rdom.info()
print "UUID of the guest vm : ", rdom.UUIDString()
print "OS type of the guest vm : ", rdom.OSType()
print "Max vcpus of the guest vm : ", str(rdom.maxVcpus())
print "State of the guest vm : ", str(state)
print "Name of the guest vm : ", rdom.name()
print "Max memory of the guest vm : ", str(maxmem)
print "Number of cpus in the guest vm : ", str(cpus)

conn.close()
exit(1)
```



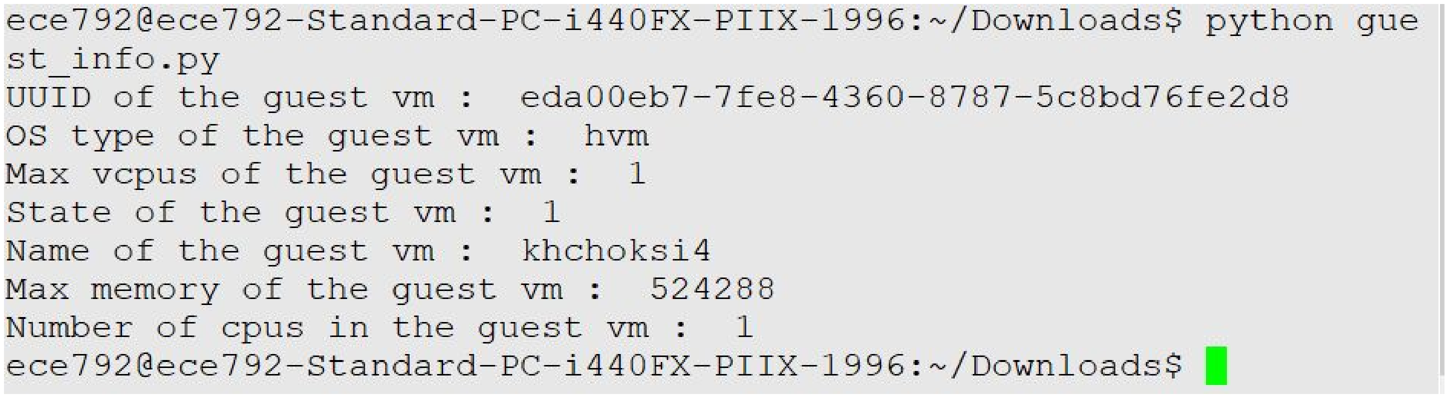
```
ece792@ece792-Standard-PC-i440FX-PIIX-1996:~/Downloads$ python hos
t_info.py
Hostname :  ece792-Standard-PC-i440FX-PIIX-1996
Number of vcpus :  16
Memory size :  24109
Clock speed of CPUs :  2199
Number of CPUs :  4
Virtualization type:  QEMU
Canonical URI :  qemu:///system
ece792@ece792-Standard-PC-i440FX-PIIX-1996:~/Downloads$
```

(README) To run this code, simply run following:

```
$ python host_info.py
```

2) Obtaining Guest Information

```
import sys
import libvirt

conn = libvirt.open('qemu:///system')

if conn == None:
        print('Connection failed')
        exit(1)

node_info = conn.getInfo()

print "Hostname : ", conn.getHostname()
print "Number of vcpus : ", conn.getMaxVcpus(None)
print "Memory size : ", node_info[1]
print "Clock speed of CPUs : ", node_info[3]
print "Number of CPUs : ", node_info[2]
print "Virtualization type: ", conn.getType()
print "Canonical URI : ", conn.getURI()

conn.close()
exit(1)
```



```
ece792@ece792-Standard-PC-i440FX-PIIX-1996:~/Downloads$ python gue
st_info.py
UUID of the guest vm :  eda00eb7-7fe8-4360-8787-5c8bd76fe2d8
OS type of the guest vm :  hvm
Max vcpus of the guest vm :  1
State of the guest vm :  1
Name of the guest vm :  khchoksi4
Max memory of the guest vm :  524288
Number of cpus in the guest vm :  1
ece792@ece792-Standard-PC-i440FX-PIIX-1996:~/Downloads$ 
```

(README) To run this code, simply run following:

```
$ python guest_info.py
```

3) Performance Monitoring:

We calculate cpu utilization using : 100*(cpu_time$_{t2}$ - cpu_time$_{t1}$)/10^9*(t2 - t1)

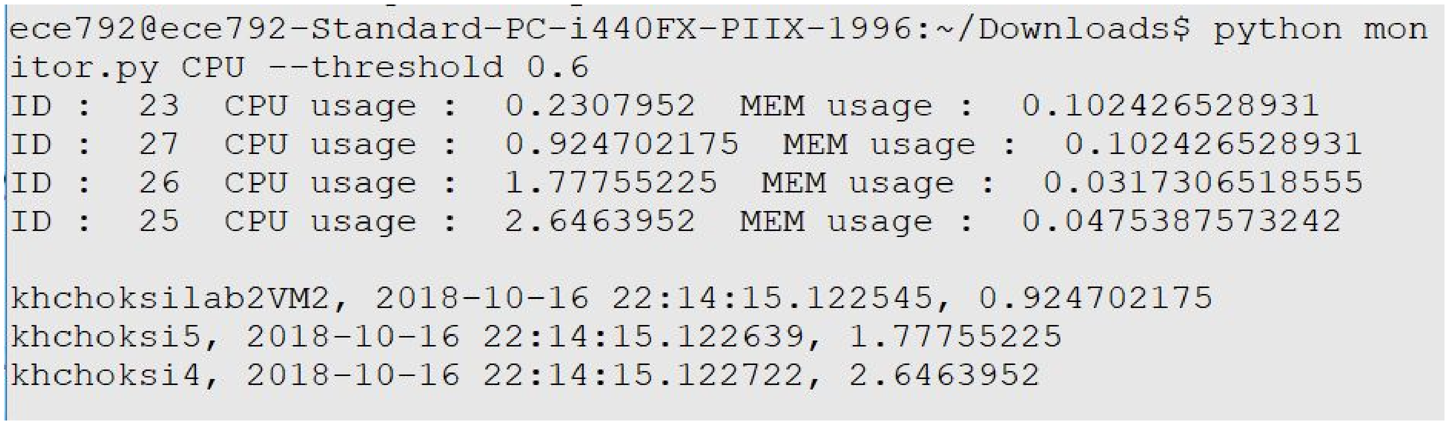
Where t1 and t2 are in seconds, and cpu_time is in nanoseconds

In cases where we have more than 1 vcpu, we divide the utilization by the number of vcpus to obtain the aggregate CPU utilization.

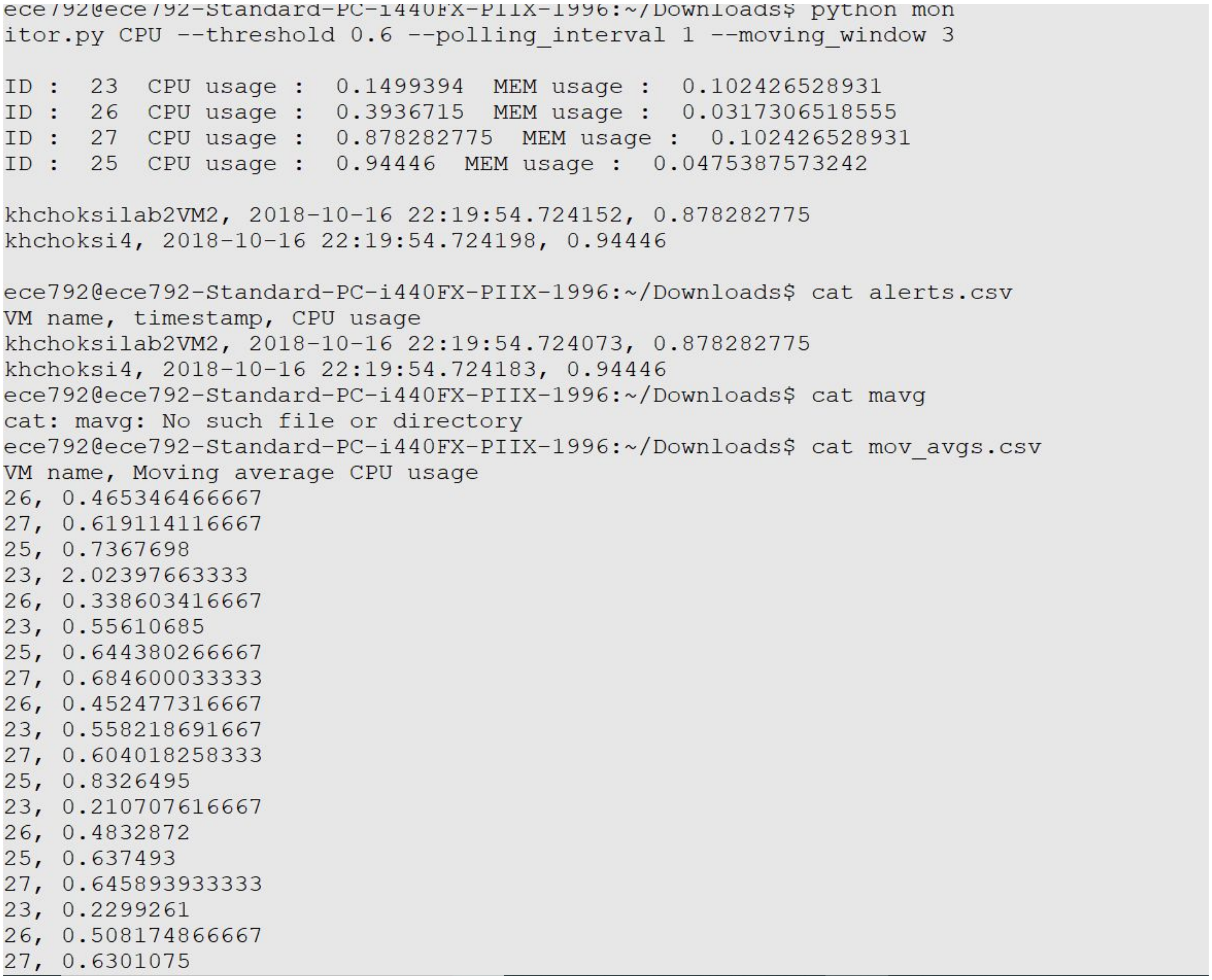
For memory, we calculate the utilization as : 1 - memory_available/memory_actual
This gives us the amount of memory used, which we believe is a measure of the memory utilization

```
import sys
import libvirt
import random
import os
import datetime
import argparse
import collections
from time import sleep

py_parser = argparse.ArgumentParser(description='Monitor script')

py_parser.add_argument('order', nargs=1, choices = ["CPU","MEM"], help="order to sort
by")
py_parser.add_argument('--threshold', nargs=1, type = float, help="threshold CPU
value")
py_parser.add_argument('--polling_interval', nargs=1, type = float, help="polling
interval value")
py_parser.add_argument('--moving_window', nargs=1, type = int, help="moving window
value")

py_args = py_parser.parse_args()

order = py_args.order[0]

if not py_args.threshold:
        threshold = 0
else:
        threshold = py_args.threshold[0]

conn = libvirt.open('qemu:///system')
if not conn:
        print "Connection failed"
        exit(1)

domainIDs = conn.listDomainsID()
if len(domainIDs) == 0:
        print "No active domains"

vm_id_list = [] + domainIDs
stats = []
for vm_id in vm_id_list:
        vm = conn.lookupByID(vm_id)
        cpu_stats = vm.getCPUStats(True)[0]
        mem_stats = vm.memoryStats()
        vcpus = vm.maxVcpus()
        stats.append([vm_id, cpu_stats['cpu_time']*1.0,\
        1 - mem_stats['available']*1.0/mem_stats['actual'], vcpus, vm])

sleep(1)

for indx, stat in enumerate(stats):
        vm = stat[-1]
        cpu_stats = vm.getCPUStats(True)[0]
        stats[indx][1] = (cpu_stats['cpu_time'] - stats[indx][1])/10**9
        stats[indx][1] = (stats[indx][1]*100)/stats[indx][3]
        if stats[indx][1] > 100:
                stats[indx][1] = 100

#Sort by CPU or MEM
if order == "CPU":
        stats.sort(key=lambda x: x[1])
else:
        stats.sort(key=lambda x: x[2])

#Logging
if not os.path.isfile("alerts.csv"):
        log_file = open("alerts.csv",'w')
        log_file.write("VM name, timestamp, CPU usage\n")
else:
        log_file = open("alerts.csv",'a')

log_op = ""
print_op = ""
for vm_stat in stats:
        #Printing sorted list
        print "ID : ", vm_stat[0], " CPU usage : ", vm_stat[1]\
        , " MEM usage : ", vm_stat[2]
        #If cpu > threshold, log and print
        if vm_stat[1] > threshold:
                log_op += vm_stat[-1].name()+", "+str(datetime.datetime.now())+",
"+str(vm_stat[1])+"\n"
                print_op += vm_stat[-1].name()+", "+str(datetime.datetime.now())+",
"+str(vm_stat[1])+"\n"

log_file.write(log_op)
print "\n",print_op

#Bonus part
```

```
poll_int = py_args.polling_interval[0]
mov_wind = py_args.moving_window[0]

if poll_int==None or mov_wind==None:
        exit(1)
poll_int = py_args.polling_interval[0]
mov_wind = py_args.moving_window[0]

prev_poll_time = {}
curr_poll_time = {}
polled_values = collections.defaultdict(list)

for indx, stat in enumerate(stats):
        vm = stat[-1]
        cpu_stats = vm.getCPUStats(True)[0]
        prev_poll_time[stat[0]] = cpu_stats['cpu_time']

if not os.path.isfile("mov_avgs.csv"):
        mavgs = open("mov_avgs.csv",'w')
        mavgs.write("VM ID, Moving average CPU usage\n")
else:
        mavgs = open("mov_avgs.csv",'a')

poll_timer = 0
try:
        while True:
                sleep(poll_int)

                if poll_timer >= mov_wind*poll_int:
                        #Log the moving window averages

                        unsorted_list = []
                        for v in polled_values:
                                unsorted_list.append([v,
sum(polled_values[v])/len(polled_values[v])])
                                polled_values[v].pop(0)
                        unsorted_list.sort(key = lambda x: x[1])
                        for v in unsorted_list:
                                mavgs.write(str(v[0])+", "+str(v[1])+"\n")
                        #print unsorted_list

                for indx, stat in enumerate(stats):
                        vm = stat[-1]
                        cpu_stats = vm.getCPUStats(True)[0]
                        curr_poll_time[stat[0]] = cpu_stats['cpu_time']
                        polled_values[stat[0]].append(100*(curr_poll_time[stat[0]] - \
                        prev_poll_time[stat[0]])/(10**9 * poll_int * stats[indx][3]))
                        prev_poll_time[stat[0]] = curr_poll_time[stat[0]]
                        #print polled_values
                poll_timer += poll_int
except:
        mavgs.close()
        exit(0)
```

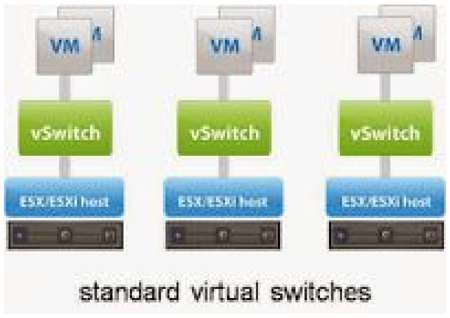


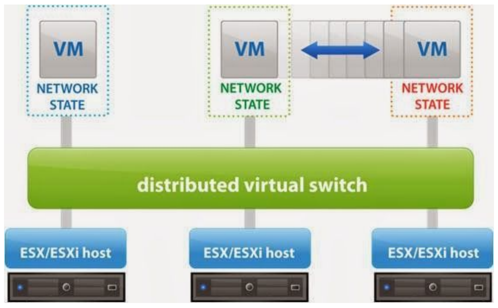
```
ece792@ece792-Standard-PC-i440FX-PIIX-1996:~/Downloads$ python mon
itor.py CPU --threshold 0.6
ID :  23  CPU usage :  0.2307952  MEM usage :  0.102426528931
ID :  27  CPU usage :  0.924702175  MEM usage :  0.102426528931
ID :  26  CPU usage :  1.77755225  MEM usage :  0.0317306518555
ID :  25  CPU usage :  2.6463952  MEM usage :  0.0475387573242

khchoksilab2VM2, 2018-10-16 22:14:15.122545, 0.924702175
khchoksi5, 2018-10-16 22:14:15.122639, 1.77755225
khchoksi4, 2018-10-16 22:14:15.122722, 2.6463952
```



```
ece792@ece792-Standard-PC-i440FX-PIIX-1996:~/Downloads$ python mon
itor.py CPU --threshold 0.6 --polling_interval 1 --moving_window 3

ID :  23  CPU usage :  0.1499394  MEM usage :  0.102426528931
ID :  26  CPU usage :  0.3936715  MEM usage :  0.0317306518555
ID :  27  CPU usage :  0.878282775  MEM usage :  0.102426528931
ID :  25  CPU usage :  0.94446  MEM usage :  0.0475387573242

khchoksilab2VM2, 2018-10-16 22:19:54.724152, 0.878282775
khchoksi4, 2018-10-16 22:19:54.724198, 0.94446

ece792@ece792-Standard-PC-i440FX-PIIX-1996:~/Downloads$ cat alerts.csv
VM name, timestamp, CPU usage
khchoksilab2VM2, 2018-10-16 22:19:54.724073, 0.878282775
khchoksi4, 2018-10-16 22:19:54.724183, 0.94446
ece792@ece792-Standard-PC-i440FX-PIIX-1996:~/Downloads$ cat mavg
cat: mavg: No such file or directory
ece792@ece792-Standard-PC-i440FX-PIIX-1996:~/Downloads$ cat mov_avgs.csv
VM name, Moving average CPU usage
26, 0.465346466667
27, 0.619114116667
25, 0.7367698
23, 2.02397663333
26, 0.338603416667
23, 0.55610685
25, 0.644380266667
27, 0.684600033333
26, 0.452477316667
23, 0.558218691667
27, 0.604018258333
25, 0.8326495
23, 0.210707616667
26, 0.4832872
25, 0.637493
27, 0.645893933333
23, 0.2299261
26, 0.508174866667
27, 0.6301075
```

PROBLEM 6:





Distributed vSwitch: Distributed vSwitches allow different hosts to use the switch as long as they exist within the same host cluster. A distributed vSwitch extends its ports and management across all the servers in a cluster, supporting up to 500 hosts per distributed switch. Instead of making virtual networks more complicated with its additional options, the distributed vSwitch simplifies operations and helps catch configuration errors and increase network visibility.

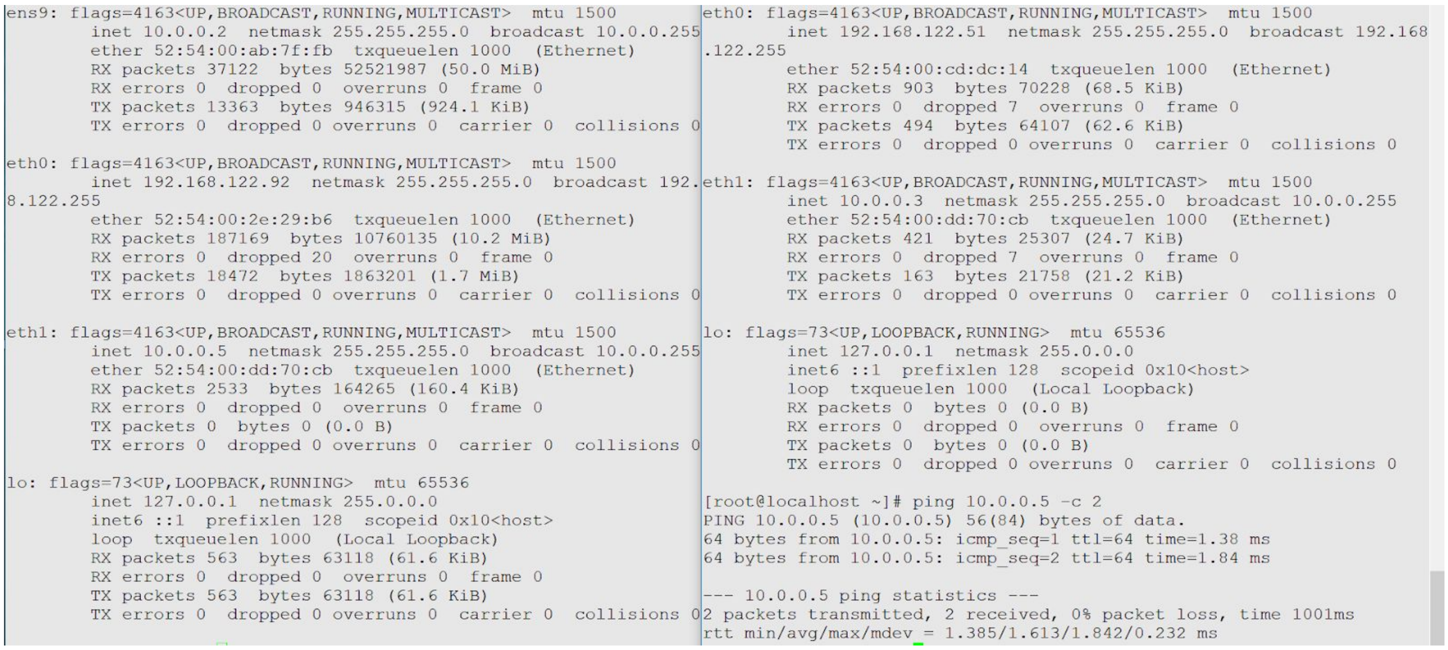
Standard vSwitch: A standard vSwitch works within one ESX/ESXi host only. Standard switch is created in host level i.e. we can create and manage vSphere standard switch independently on an ESXi host. Inbound traffic shaping is not available as a part in the standard switch.

In cases where we want to extend the L2 layer across VMs in different hosts, instead of creating GRE/VXLAN tunnels between each host pair, we could use the distributed vSwitch.

Ref:
https://searchvmware.techtarget.com/photostory/2240185944/Getting-VMware-terminology-straight/9/How-do-switches-vSwitches-and-distributed-vSwitches-differ

PROBLEM 7:

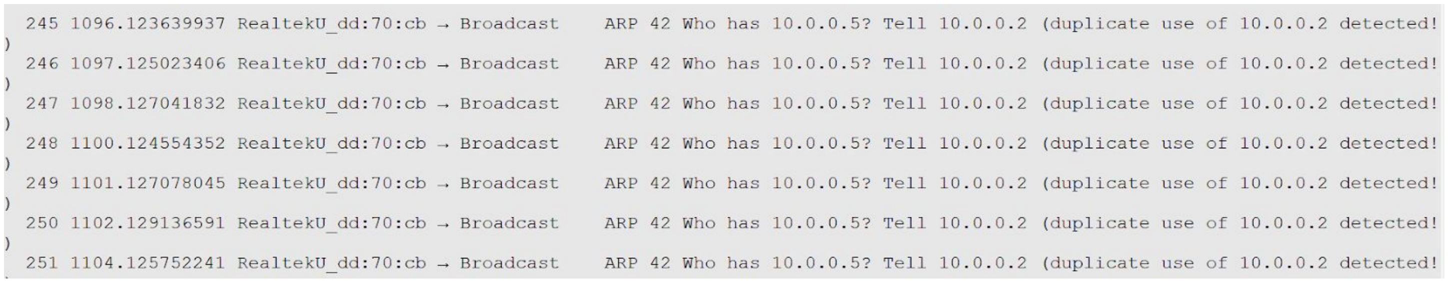
1. If 2 VMs connected to same bridge in bridge mode:
    a. **Same MAC address**:
    It will overwrite the entries for same MAC address when ARP will be done. The 2VMs should be able to ping each other.
    As they have different IP address, the ARP will be resolved and ping will be successful between two VMs as well as bridge will work fine.



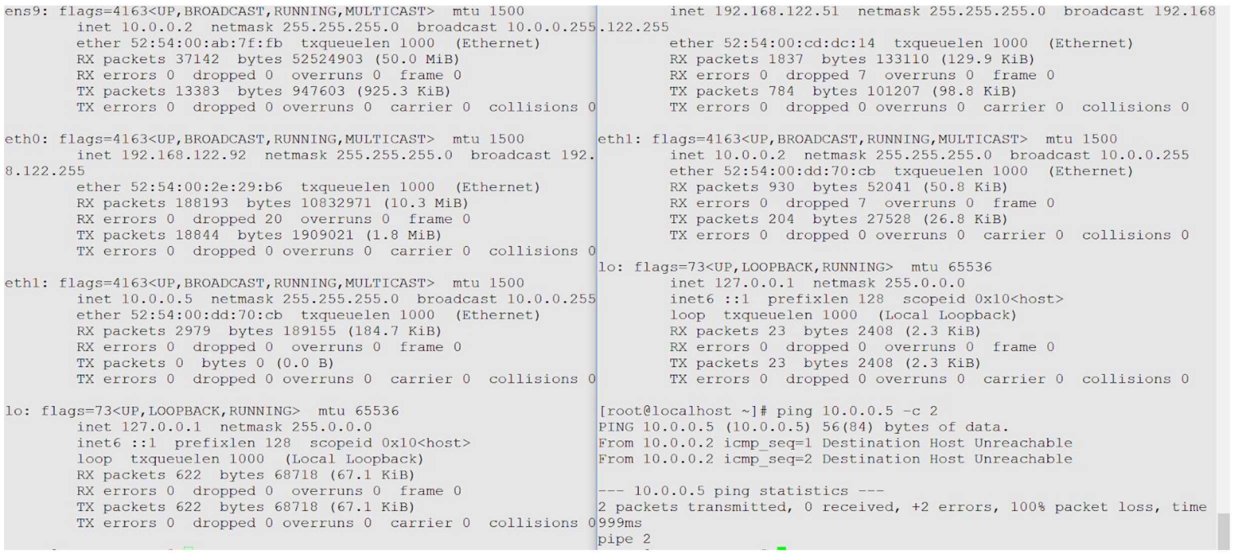
```
ens9: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.0.2  netmask 255.255.255.0  broadcast 10.0.0.255
        ether 52:54:00:ab:7f:fb  txqueuelen 1000  (Ethernet)
        RX packets 37122  bytes 52521987 (50.0 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 13363  bytes 946315 (924.1 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.122.92  netmask 255.255.255.0  broadcast 192.
8.122.255
        ether 52:54:00:2e:29:b6  txqueuelen 1000  (Ethernet)
        RX packets 187169  bytes 10760135 (10.2 MiB)
        RX errors 0  dropped 20  overruns 0  frame 0
        TX packets 18472  bytes 1863201 (1.7 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.0.5  netmask 255.255.255.0  broadcast 10.0.0.255
        ether 52:54:00:dd:70:cb  txqueuelen 1000  (Ethernet)
        RX packets 2533  bytes 164265 (160.4 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 563  bytes 63118 (61.6 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 563  bytes 63118 (61.6 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

```
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.122.51  netmask 255.255.255.0  broadcast 192.168
.122.255
        ether 52:54:00:cd:dc:14  txqueuelen 1000  (Ethernet)
        RX packets 903  bytes 70228 (68.5 KiB)
        RX errors 0  dropped 7  overruns 0  frame 0
        TX packets 494  bytes 64107 (62.6 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.0.3  netmask 255.255.255.0  broadcast 10.0.0.255
        ether 52:54:00:dd:70:cb  txqueuelen 1000  (Ethernet)
        RX packets 421  bytes 25307 (24.7 KiB)
        RX errors 0  dropped 7  overruns 0  frame 0
        TX packets 163  bytes 21758 (21.2 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

[root@localhost ~]# ping 10.0.0.5 -c 2
PING 10.0.0.5 (10.0.0.5) 56(84) bytes of data.
64 bytes from 10.0.0.5: icmp_seq=1 ttl=64 time=1.38 ms
64 bytes from 10.0.0.5: icmp_seq=2 ttl=64 time=1.84 ms

--- 10.0.0.5 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 1.385/1.613/1.842/0.232 ms
```

    b. **Same IP address:**
    Switch will throw an error that, duplicate use of ip detected.



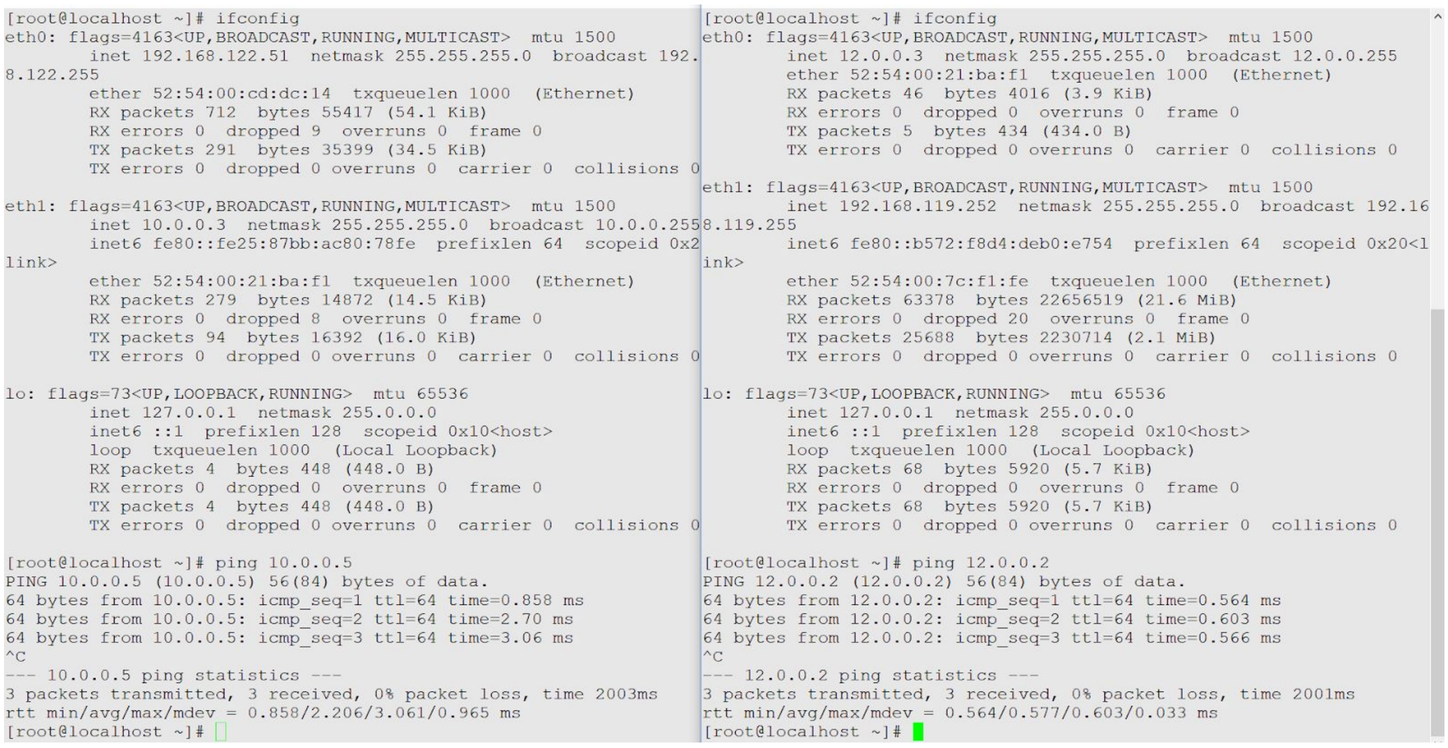
```
  245 1096.123639937 RealtekU_dd:70:cb → Broadcast    ARP 42 Who has 10.0.0.5? Tell 10.0.0.2 (duplicate use of 10.0.0.2 detected!
)
  246 1097.125023406 RealtekU_dd:70:cb → Broadcast    ARP 42 Who has 10.0.0.5? Tell 10.0.0.2 (duplicate use of 10.0.0.2 detected!
)
  247 1098.127041832 RealtekU_dd:70:cb → Broadcast    ARP 42 Who has 10.0.0.5? Tell 10.0.0.2 (duplicate use of 10.0.0.2 detected!
)
  248 1100.124554352 RealtekU_dd:70:cb → Broadcast    ARP 42 Who has 10.0.0.5? Tell 10.0.0.2 (duplicate use of 10.0.0.2 detected!
)
  249 1101.127078045 RealtekU_dd:70:cb → Broadcast    ARP 42 Who has 10.0.0.5? Tell 10.0.0.2 (duplicate use of 10.0.0.2 detected!
)
  250 1102.129136591 RealtekU_dd:70:cb → Broadcast    ARP 42 Who has 10.0.0.5? Tell 10.0.0.2 (duplicate use of 10.0.0.2 detected!
)
  251 1104.125752241 RealtekU_dd:70:cb → Broadcast    ARP 42 Who has 10.0.0.5? Tell 10.0.0.2 (duplicate use of 10.0.0.2 detected!
```



```
ens9: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.0.2  netmask 255.255.255.0  broadcast 10.0.0.255
        ether 52:54:00:ab:7f:fb  txqueuelen 1000  (Ethernet)
        RX packets 37142  bytes 52524903 (50.0 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 13383  bytes 947603 (925.3 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.122.92  netmask 255.255.255.0  broadcast 192.
8.122.255
        ether 52:54:00:2e:29:b6  txqueuelen 1000  (Ethernet)
        RX packets 188193  bytes 10832971 (10.3 MiB)
        RX errors 0  dropped 20  overruns 0  frame 0
        TX packets 18844  bytes 1909021 (1.8 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.0.5  netmask 255.255.255.0  broadcast 10.0.0.255
        ether 52:54:00:dd:70:cb  txqueuelen 1000  (Ethernet)
        RX packets 2979  bytes 189155 (184.7 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 622  bytes 68718 (67.1 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 622  bytes 68718 (67.1 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

```
        inet 192.168.122.51  netmask 255.255.255.0  broadcast 192.168
.122.255
        ether 52:54:00:cd:dc:14  txqueuelen 1000  (Ethernet)
        RX packets 1837  bytes 133110 (129.9 KiB)
        RX errors 0  dropped 7  overruns 0  frame 0
        TX packets 784  bytes 101207 (98.8 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.0.2  netmask 255.255.255.0  broadcast 10.0.0.255
        ether 52:54:00:dd:70:cb  txqueuelen 1000  (Ethernet)
        RX packets 930  bytes 52041 (50.8 KiB)
        RX errors 0  dropped 7  overruns 0  frame 0
        TX packets 204  bytes 27528 (26.8 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 23  bytes 2408 (2.3 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 23  bytes 2408 (2.3 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

[root@localhost ~]# ping 10.0.0.5 -c 2
PING 10.0.0.5 (10.0.0.5) 56(84) bytes of data.
From 10.0.0.2 icmp_seq=1 Destination Host Unreachable
From 10.0.0.2 icmp_seq=2 Destination Host Unreachable

--- 10.0.0.5 ping statistics ---
2 packets transmitted, 0 received, +2 errors, 100% packet loss, time
999ms
pipe 2
```

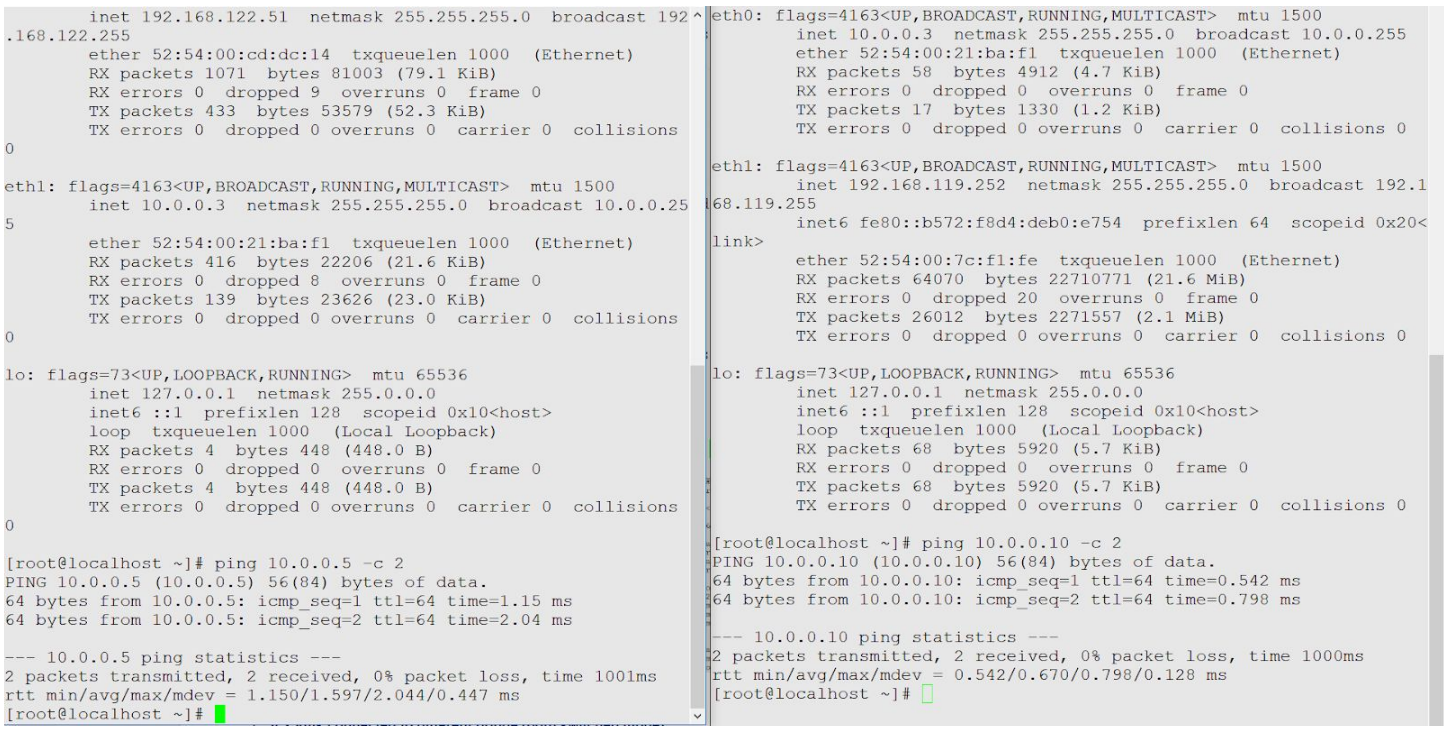
2. **If 2 Vms connected to different bridge (both bridge mode)**
    a. Same MAC address:
    As both VMs are in different networks, it won't affect the bridge.



```
[root@localhost ~]# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.122.51  netmask 255.255.255.0  broadcast 192.1
8.122.255
        ether 52:54:00:cd:dc:14  txqueuelen 1000  (Ethernet)
        RX packets 712  bytes 55417 (54.1 KiB)
        RX errors 0  dropped 9  overruns 0  frame 0
        TX packets 291  bytes 35399 (34.5 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.0.3  netmask 255.255.255.0  broadcast 10.0.0.255
        inet6 fe80::fe25:87bb:ac80:78fe  prefixlen 64  scopeid 0x2
link>
        ether 52:54:00:21:ba:f1  txqueuelen 1000  (Ethernet)
        RX packets 279  bytes 14872 (14.5 KiB)
        RX errors 0  dropped 8  overruns 0  frame 0
        TX packets 94  bytes 16392 (16.0 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 4  bytes 448 (448.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 4  bytes 448 (448.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

[root@localhost ~]# ping 10.0.0.5
PING 10.0.0.5 (10.0.0.5) 56(84) bytes of data.
64 bytes from 10.0.0.5: icmp_seq=1 ttl=64 time=0.858 ms
64 bytes from 10.0.0.5: icmp_seq=2 ttl=64 time=2.70 ms
64 bytes from 10.0.0.5: icmp_seq=3 ttl=64 time=3.06 ms
^C
--- 10.0.0.5 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 0.858/2.206/3.061/0.965 ms
[root@localhost ~]#
```

```
[root@localhost ~]# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 12.0.0.3  netmask 255.255.255.0  broadcast 12.0.0.255
        ether 52:54:00:21:ba:f1  txqueuelen 1000  (Ethernet)
        RX packets 46  bytes 4016 (3.9 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 5  bytes 434 (434.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.119.252  netmask 255.255.255.0  broadcast 192.16
8.119.255
        inet6 fe80::b572:f8d4:deb0:e754  prefixlen 64  scopeid 0x20<l
ink>
        ether 52:54:00:7c:f1:fe  txqueuelen 1000  (Ethernet)
        RX packets 63378  bytes 22656519 (21.6 MiB)
        RX errors 0  dropped 20  overruns 0  frame 0
        TX packets 25688  bytes 2230714 (2.1 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 68  bytes 5920 (5.7 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 68  bytes 5920 (5.7 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

[root@localhost ~]# ping 12.0.0.2
PING 12.0.0.2 (12.0.0.2) 56(84) bytes of data.
64 bytes from 12.0.0.2: icmp_seq=1 ttl=64 time=0.564 ms
64 bytes from 12.0.0.2: icmp_seq=2 ttl=64 time=0.603 ms
64 bytes from 12.0.0.2: icmp_seq=3 ttl=64 time=0.566 ms
^C
--- 12.0.0.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2001ms
rtt min/avg/max/mdev = 0.564/0.577/0.603/0.033 ms
[root@localhost ~]#
```

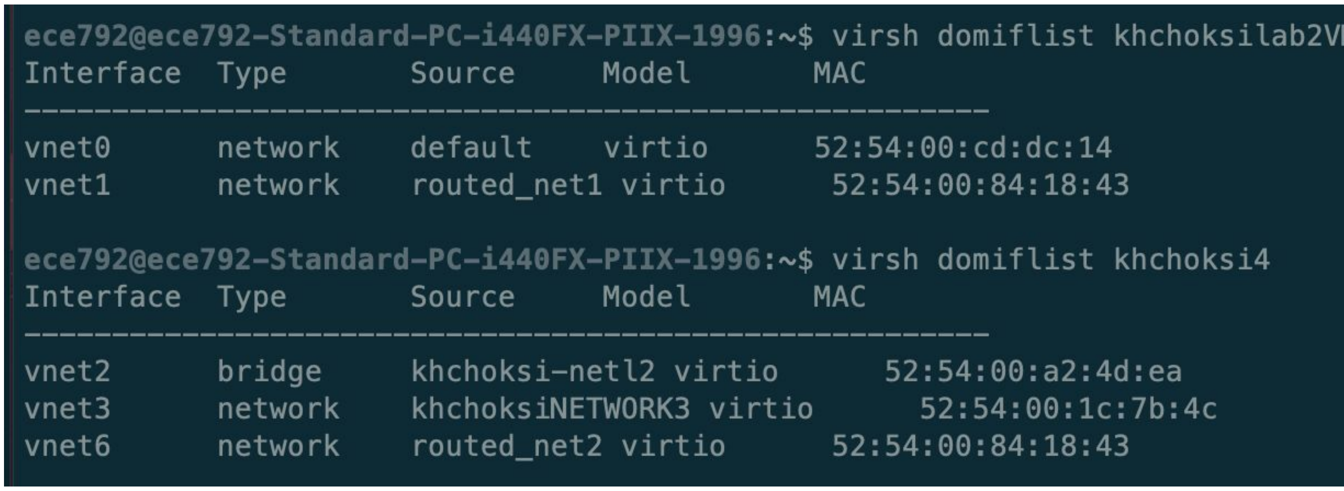
    b. Same IP address:
    As both the VMs are in different network, it won't allow to change the ip of second VM and network won't be affected.



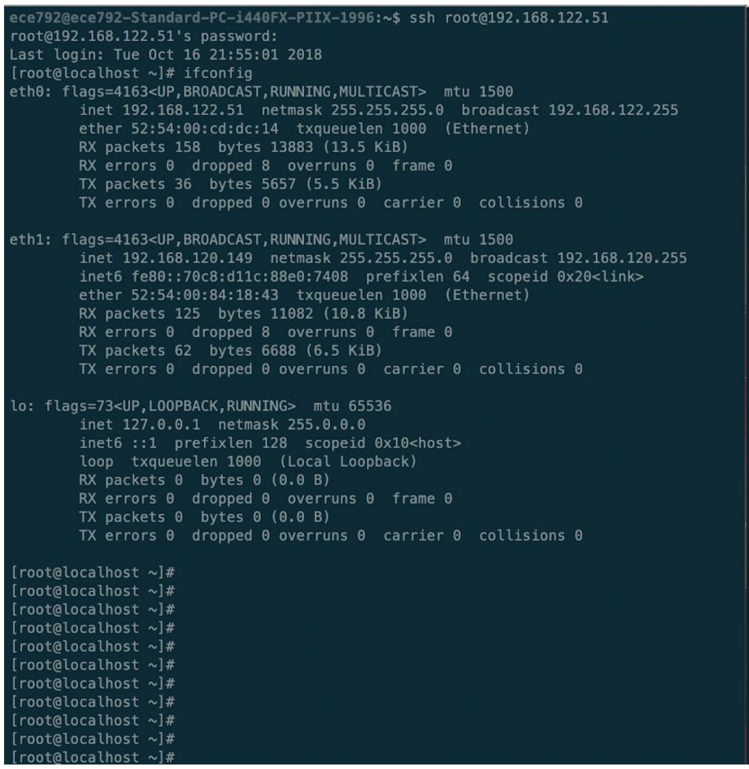
```
        inet 192.168.122.51  netmask 255.255.255.0  broadcast 192
.168.122.255
        ether 52:54:00:cd:dc:14  txqueuelen 1000  (Ethernet)
        RX packets 1071  bytes 81003 (79.1 KiB)
        RX errors 0  dropped 9  overruns 0  frame 0
        TX packets 433  bytes 53579 (52.3 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions
0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.0.3  netmask 255.255.255.0  broadcast 10.0.0.25
5
        ether 52:54:00:21:ba:f1  txqueuelen 1000  (Ethernet)
        RX packets 416  bytes 22206 (21.6 KiB)
        RX errors 0  dropped 8  overruns 0  frame 0
        TX packets 139  bytes 23626 (23.0 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions
0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 4  bytes 448 (448.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 4  bytes 448 (448.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions
0

[root@localhost ~]# ping 10.0.0.5 -c 2
PING 10.0.0.5 (10.0.0.5) 56(84) bytes of data.
64 bytes from 10.0.0.5: icmp_seq=1 ttl=64 time=1.15 ms
64 bytes from 10.0.0.5: icmp_seq=2 ttl=64 time=2.04 ms

--- 10.0.0.5 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 1.150/1.597/2.044/0.447 ms
[root@localhost ~]#
```

```
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.0.3  netmask 255.255.255.0  broadcast 10.0.0.255
        ether 52:54:00:21:ba:f1  txqueuelen 1000  (Ethernet)
        RX packets 58  bytes 4912 (4.7 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 17  bytes 1330 (1.2 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.119.252  netmask 255.255.255.0  broadcast 192.1
68.119.255
        inet6 fe80::b572:f8d4:deb0:e754  prefixlen 64  scopeid 0x20<
link>
        ether 52:54:00:7c:f1:fe  txqueuelen 1000  (Ethernet)
        RX packets 64070  bytes 22710771 (21.6 MiB)
        RX errors 0  dropped 20  overruns 0  frame 0
        TX packets 26012  bytes 2271557 (2.1 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 68  bytes 5920 (5.7 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 68  bytes 5920 (5.7 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

[root@localhost ~]# ping 10.0.0.10 -c 2
PING 10.0.0.10 (10.0.0.10) 56(84) bytes of data.
64 bytes from 10.0.0.10: icmp_seq=1 ttl=64 time=0.542 ms
64 bytes from 10.0.0.10: icmp_seq=2 ttl=64 time=0.798 ms

--- 10.0.0.10 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1000ms
rtt min/avg/max/mdev = 0.542/0.670/0.798/0.128 ms
[root@localhost ~]#
```

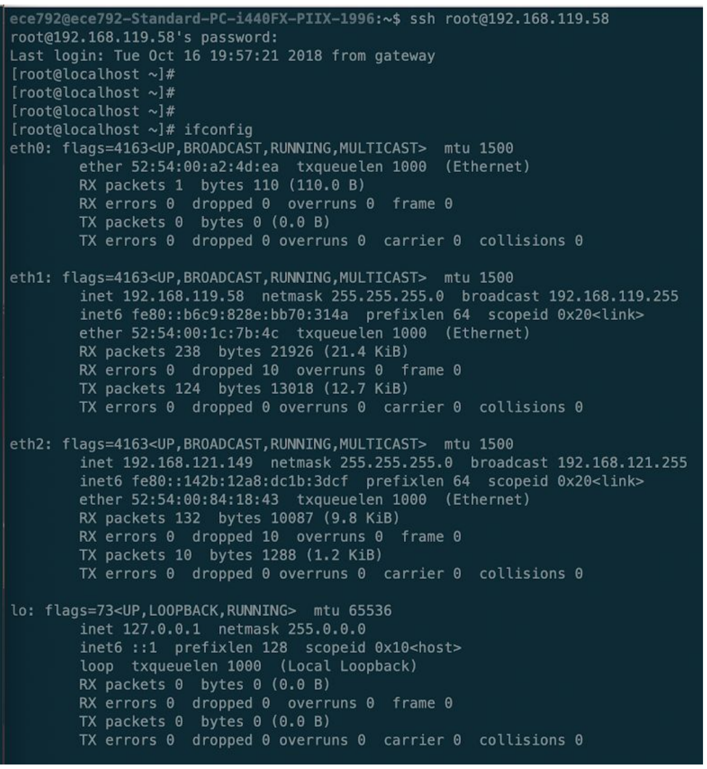
3. **If 2 Vms connected to different bridge (both routed mode)**
   a. **Same MAC address:** If the two VMs are connected to different bridges (over here sw3 and sw4) then everything will work correctly even if we provide same MAC address to both the VMs. As they both are on different networks (routed_subnet1) and (routed_subnet2), it won't hinder each other.



```
ece792@ece792-Standard-PC-i440FX-PIIX-1996:~$ virsh domiflist khchoksilab2V
Interface  Type       Source     Model       MAC
-------------------------------------------------------
vnet0      network    default    virtio      52:54:00:cd:dc:14
vnet1      network    routed_net1 virtio      52:54:00:84:18:43

ece792@ece792-Standard-PC-i440FX-PIIX-1996:~$ virsh domiflist khchoksi4
Interface  Type       Source     Model       MAC
-------------------------------------------------------
vnet2      bridge     khchoksi-netl2 virtio      52:54:00:a2:4d:ea
vnet3      network    khchoksiNETWORK3 virtio      52:54:00:1c:7b:4c
vnet6      network    routed_net2 virtio      52:54:00:84:18:43
```



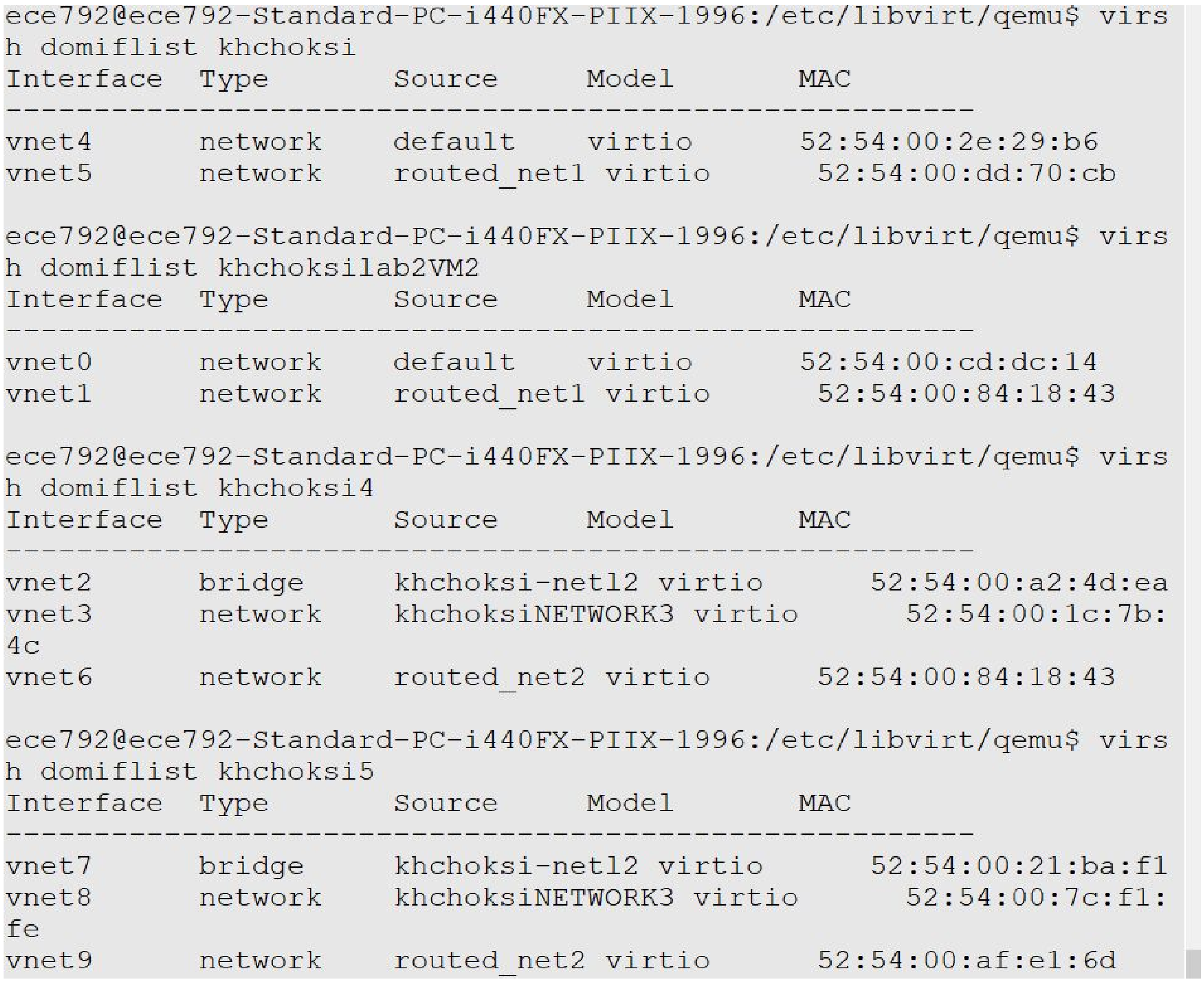
```
ece792@ece792-Standard-PC-i440FX-PIIX-1996:~$ ssh root@192.168.122.51
root@192.168.122.51's password:
Last login: Tue Oct 16 21:55:01 2018
[root@localhost ~]# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.122.51  netmask 255.255.255.0  broadcast 192.168.122.255
        ether 52:54:00:cd:dc:14  txqueuelen 1000  (Ethernet)
        RX packets 158  bytes 13883 (13.5 KiB)
        RX errors 0  dropped 8  overruns 0  frame 0
        TX packets 36  bytes 5657 (5.5 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.120.149  netmask 255.255.255.0  broadcast 192.168.120.255
        inet6 fe80::70c8:d11c:88e0:7408  prefixlen 64  scopeid 0x20<link>
        ether 52:54:00:84:18:43  txqueuelen 1000  (Ethernet)
        RX packets 125  bytes 11082 (10.8 KiB)
        RX errors 0  dropped 8  overruns 0  frame 0
        TX packets 62  bytes 6688 (6.5 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]#
```



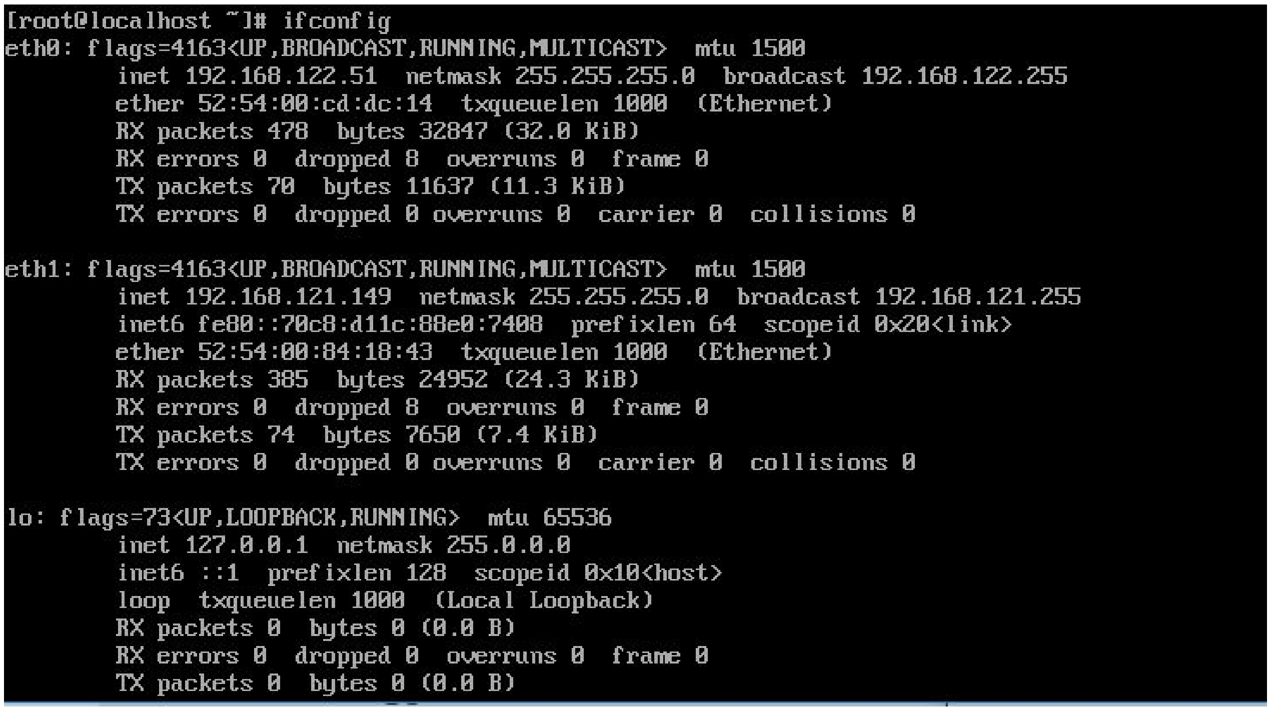
```
ece792@ece792-Standard-PC-i440FX-PIIX-1996:~$ ssh root@192.168.119.58
root@192.168.119.58's password:
Last login: Tue Oct 16 19:57:21 2018 from gateway
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        ether 52:54:00:a2:4d:ea  txqueuelen 1000  (Ethernet)
        RX packets 1  bytes 110 (110.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.119.58  netmask 255.255.255.0  broadcast 192.168.119.255
        inet6 fe80::b6c9:828e:bb70:314a  prefixlen 64  scopeid 0x20<link>
        ether 52:54:00:1c:7b:4c  txqueuelen 1000  (Ethernet)
        RX packets 238  bytes 21926 (21.4 KiB)
        RX errors 0  dropped 10  overruns 0  frame 0
        TX packets 124  bytes 13018 (12.7 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth2: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.121.149  netmask 255.255.255.0  broadcast 192.168.121.255
        inet6 fe80::142b:12a8:dc1b:3dcf  prefixlen 64  scopeid 0x20<link>
        ether 52:54:00:84:18:43  txqueuelen 1000  (Ethernet)
        RX packets 132  bytes 10087 (9.8 KiB)
        RX errors 0  dropped 10  overruns 0  frame 0
        TX packets 10  bytes 1288 (1.2 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

   b. **Same IP address:**

If they have the same IP address, the VM which has its IP address from the other subnet will lose connectivity with the subnet it is in. In our case, the VM khchoksilab2VM2 will lose connectivity from the 192.168.120.0/24 network as it has been moved to the 192.168.121.0/24 network IP address.



```
ece792@ece792-Standard-PC-i440FX-PIIX-1996:/etc/libvirt/qemu$ virs
h domiflist khchoksi
Interface  Type       Source     Model       MAC
-------------------------------------------------------
vnet4      network    default    virtio      52:54:00:2e:29:b6
vnet5      network    routed_net1 virtio      52:54:00:dd:70:cb

ece792@ece792-Standard-PC-i440FX-PIIX-1996:/etc/libvirt/qemu$ virs
h domiflist khchoksilab2VM2
Interface  Type       Source     Model       MAC
-------------------------------------------------------
vnet0      network    default    virtio      52:54:00:cd:dc:14
vnet1      network    routed_net1 virtio      52:54:00:84:18:43

ece792@ece792-Standard-PC-i440FX-PIIX-1996:/etc/libvirt/qemu$ virs
h domiflist khchoksi4
Interface  Type       Source     Model       MAC
-------------------------------------------------------
vnet2      bridge     khchoksi-net12 virtio      52:54:00:a2:4d:ea
vnet3      network    khchoksiNETWORK3 virtio      52:54:00:1c:7b:
4c
vnet6      network    routed_net2 virtio      52:54:00:84:18:43

ece792@ece792-Standard-PC-i440FX-PIIX-1996:/etc/libvirt/qemu$ virs
h domiflist khchoksi5
Interface  Type       Source     Model       MAC
-------------------------------------------------------
vnet7      bridge     khchoksi-net12 virtio      52:54:00:21:ba:f1
vnet8      network    khchoksiNETWORK3 virtio      52:54:00:7c:f1:
fe
vnet9      network    routed_net2 virtio      52:54:00:af:e1:6d
```



```
[root@localhost ~]# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.122.51  netmask 255.255.255.0  broadcast 192.168.122.255
        ether 52:54:00:cd:dc:14  txqueuelen 1000  (Ethernet)
        RX packets 478  bytes 32847 (32.0 KiB)
        RX errors 0  dropped 8  overruns 0  frame 0
        TX packets 70  bytes 11637 (11.3 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.121.149  netmask 255.255.255.0  broadcast 192.168.121.255
        inet6 fe80::70c8:d11c:88e0:7408  prefixlen 64  scopeid 0x20<link>
        ether 52:54:00:84:18:43  txqueuelen 1000  (Ethernet)
        RX packets 385  bytes 24952 (24.3 KiB)
        RX errors 0  dropped 8  overruns 0  frame 0
        TX packets 74  bytes 7650 (7.4 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
```