

Name: Chaitanya Pidugu.

Banner I'd: 800746834

Data Center Networks with Overlay Networks and Microsegmentation

In this part of the lab, we will create a few tenant VMs to illustrate how communication through the overlay/underlay network functions. We will also show how a set of security rules can be generated to map different rules to different VMs, based on their roles.

Step A - Install VirtualBox & Configure DC Gateway

1. Stop FRR from running so you don't potentially advertise a shorter route to the 192.168.42.0/24 network than your neighbor:
`eid@whateverphobia$ sudo systemctl disable --now frr.service`
2. On your lab machines (or home machine, as long as you have 8 cores, >=32GB of RAM and 500+GB of solid-state disk space) install Virtual Box using apt:
`eid@whateverphobia$ sudo apt install virtualbox`

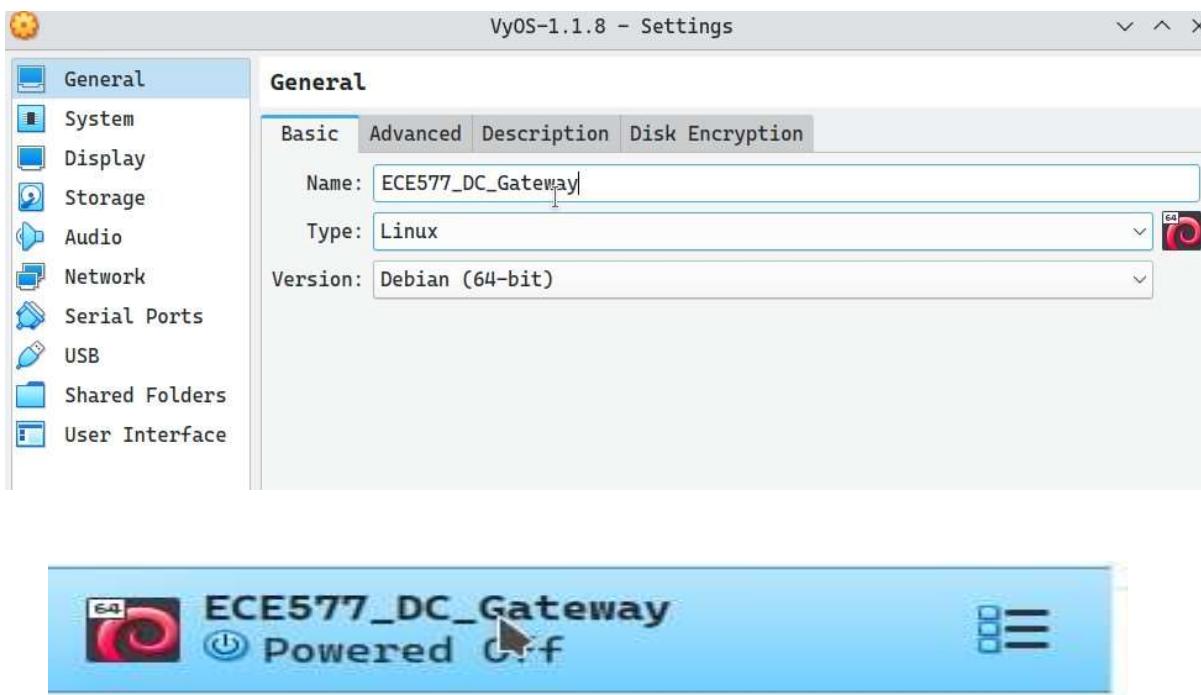
```
cpidugu@pyrophobia:~$ sudo systemctl disable --now frr.service
[sudo] password for cpidugu:
Synchronizing state of frr.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install disable frr
cpidugu@pyrophobia:~$ sudo apt install virtualbox
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
virtualbox is already the newest version (6.1.38-dfsg-3~ubuntu1.22.04.1).
0 upgraded, 0 newly installed, 0 to remove and 35 not upgraded.
cpidugu@pyrophobia:~$
```

3. You might want to reboot after install (I think it dynamically builds/loads some modules)
4. Download the VyOS VirtualBox appliance from the lab directory (vyos-1.1.8-amd64.ova)



vyos-1.1.8-amd64.ova
231 MB – googleusercontent.com – 13:56

5. Start VirtualBox and go to File → Import Appliance and select the VyOS .ova file (v 1.1.8 as of this writing). If you get a warning about a certificate, you can ignore it. On the Appliance Settings window, you should change the name from VyOS-1.1.8 to ECE577_DC_Gateway



6. Wait for it to import and it should appear as a VM to run, so press the green Start arrow.
7. The image has two NICs, one that is bridged to your eno1 interface and the second on an internal network named “intnet”.
8. Set the IP address of the first interface (eth1) to be 10.0.1.1XX/24. Since this is bridged to your eno1 interface, it should be in the same subnet, and should be accessible from the same machine (or any other machine in EB3012 on the 10.0.1.0/24 subnet).
 - a. Login using the username vyos and password vyos
 - b. VyOS has a similar interface to FRR and our Baystack switches that use the command prompt, with a little difference. Like FRR/Baystack, typing “configure” is needed to modify the configuration of the device. (The prompt should change from vyos@vyos:~\$ to vyos@vyos#). Most configuration is done using the “set” command, and you can always type ? to see the next options and press Tab to complete. For example, to do this step, you would type in config mode: vyos@vyos# set interfaces ethernet eth0 address 10.0.1.1XX/24
 - c. Unlike FRR/Baystack, a successful set command doesn’t take effect until you also issue a “commit” command to actually make this change. This allows you to group commands together, instead of always having to make changes one at a time

- i. Typing “discard” undoes all uncommitted changes
 - ii. Typing “delete” followed by a command undoes that command (kind of like the “no” keyword in FRR/Baystack)
 - d. To save the current config, you just have to type the “save” command, which will write the current config to the file VyOS reads on startup.
9. Change the second NIC to have the address 192.168.42.1/24

```

ECE577_DC_Gateway [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
interfaces {
    ethernet eth0 {
        address 10.0.1.136/24
        duplex auto
        hw-id 08:00:27:17:88:ae
        smp_affinity auto
        speed auto
    }
    ethernet eth1 {
        address 192.168.42.1/24
        duplex auto
        hw-id 08:00:27:ca:d5:2c
        smp_affinity auto
        speed auto
    }
    loopback lo {
    }
}
protocols {
    static {
        route 0.0.0.0/0 {
            next-hop 10.0.1.254 {
            }
        }
    }
}

```

10. Set the VyOS router’s default gateway to the 10.0.1.254 address
 11. Figure out how to have the ECE577_DC_Gateway NAT all outbound traffic to the 10.0.1.1XX address

```

[edit]
vyos@vyos# set system gateway-address 10.0.1.254
[edit]
vyos@vyos# set nat source rule 100 outbound-interface eth0
[edit]
vyos@vyos# set nat source rule 100 source address 192.168.42.0/24
[edit]
vyos@vyos# set nat source rule 100 translation address masquerade
[edit]
vyos@vyos#

```

12. Set the DNS resolver of the ECE577_DC_Gateway to the 146.163.252.126 address

13. Configure the ECE577_DC_Gateway to forward DNS requests on the eth1 interface
 - a. OPTIONAL: Turn on SSH on the ECE577_DC_Gateway so you can access it using a standard terminal instead of the VirtualBox window by typing:
eid@whateverphobia\$ ssh -oHostKeyAlgorithms=+ssh-rsa
vyos@10.0.1.1XX
14. Once all of this is set up, your gateway should now allow anything on the intnet subnet (192.168.42.1) to access the internet by pointing to it (192.168.42.1) as their gateway and DNS resolver.

Step B - Getting the DC Running

1. Download the ECE577_SU23_Controller.ova image from here: [ECE577_SU23_DC_Controller.ova](#). The shell login is ubuntu with password ubuntu.
2. In a few minutes, your new OpenStack controller should be ready. The DC Controller has 2 network interfaces, the one on the 192.168.42.0/24 network, which is used for communication to all OpenStack nodes, and the 2nd one (the “public” network) which is nominally used for direct communication with internal VMs. The “public” subnet in the 172.24.4.0/24 address space (Yes, I know this is slightly contradictory here, as a public network should be accessible from the outside with a real IP address. For the purposes of this lab, we’ll pretend the “public” network would really be a publicly routable, IP network subnet assigned by your data center provider, and that this IP space is the 172.24.4.0/24 subnet, which we will also pretend is publicly routable). It creates a bridge interface called “br-ex” and should assign that bridge the 172.24.4.1 address to make it the gateway for the 172.24.4.0/24 subnet. As we will see in a few steps, this is the range of “publicly” accessible IPs that can be assigned to internal VMs.

NOTE: IF YOU EVER REBOOT YOUR CONTROLLER, IT DOESN'T PROPERLY REASSIGN THE IP TO br-ex ON STARTUP!!! YOU MUST DO THIS:

```
$ sudo ip addr add 172.24.4.1/24 dev br-ex  
$ sudo ip link set br-ex up
```

(You might be able to get netplan to attempt this on reboots, but since netplan starts before OpenStack, it's probably not feasible without lots of additional scripting).

```

        valid_lft forever preferred_lft forever
        inet6 ::1/128 scope host
            valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
0
    link/ether 08:00:27:d0:0b:8e brd ff:ff:ff:ff:ff:ff
    inet 192.168.42.11/24 brd 192.168.42.255 scope global enp0s3
        valid_lft forever preferred_lft forever
        inet6 fe80::a00:27ff:fed0:b8e/64 scope link
            valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel master ovs-system state UP group
p default qlen 1000
    link/ether 08:00:27:9e:ab:fc brd ff:ff:ff:ff:ff:ff
    inet6 fe80::a00:27ff:fe9e:abfc/64 scope link
        valid_lft forever preferred_lft forever
4: ovs-system: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default qlen 1000
    link/ether 86:3d:23:40:d0:db brd ff:ff:ff:ff:ff:ff
5: br-ex: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UNKNOWN group default qlen
1000
    link/ether 08:00:27:9e:ab:fc brd ff:ff:ff:ff:ff:ff
    inet 172.24.4.1/24 scope global br-ex
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe9e:abfc/64 scope link
        valid_lft forever preferred_lft forever
6: br-int: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default qlen 1000
    link/ether fe:5d:e9:c3:37:93 brd ff:ff:ff:ff:ff:ff
7: geneve_sys_6081: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 65000 qdisc noqueue master ovs-system stat
UNKNOWN group default qlen 1000
    link/ether 9a:ee:de:c7:c5:c7 brd ff:ff:ff:ff:ff:ff
    inet6 fe80::54f3:79ff:fea3:53c9/64 scope link
        valid_lft forever preferred_lft forever
8: virbr0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default qlen
1000
    link/ether 52:54:00:08:93:27 brd ff:ff:ff:ff:ff:ff
    inet 192.168.122.1/24 brd 192.168.122.255 scope global virbr0
        valid_lft forever preferred_lft forever
ubuntu@controller:~$ _

```

3. The Controller node is at IP address 192.168.42.11, but your lab machine doesn't know that network so, at this point, would forward that traffic to it's gateway (and be blocked!) So, tell your lab machine you know how to get to the 192.168.42.0 subnet via the ECE577_DC_Gateway at 10.0.1.1XX:
`eid@whateverphobia$ sudo ip route add 192.168.42.0/24 via 10.0.1.1XX`

```

cidup@pyrophobia:~$ sudo ip route add 192.168.42.0/24 via 10.0.1.136
[sudo] password for cidup:
cidup@pyrophobia:~$ ip address
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 1500 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 brd 127.255.255.255 scope host
        valid_lft forever preferred_lft forever
    inet6 ::1/128 brd :: scope host
        valid_lft forever preferred_lft forever
2: eno1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 8c:ec:4b:d2:7b:4c brd ff:ff:ff:ff:ff:ff
    altname enp0s3l1f6
    inet 10.0.1.16 brd 10.0.1.255 scope global eno1
        valid_lft forever preferred_lft forever
        inet6 fd01:8639:35c1:9:8ec:4bff:fed2:7bic/64 scope global mngtmpaddr noprefixroute
            valid_lft forever preferred_lft forever
            inet6 fd01:4616:3133:ffff:36/64 scope global
                valid_lft forever preferred_lft forever
                inet6 fd01:4616:3133:ffff:36/64 scope link
                    valid_lft forever preferred_lft forever
                    valid_lft forever preferred_lft forever
3: enp3s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state DOWN group default qlen 1000
    link/ether 00:01:21:68:61:6a brd ff:ff:ff:ff:ff:ff
4: enx00e04cc57dbf: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:00:00:00:00:00 brd ff:ff:ff:ff:ff:ff
    inet 10.0.1.100 brd 10.0.1.255 scope global enx00e04cc57dbf
        valid_lft forever preferred_lft forever
        inet6 fe80::2e0:4cff:fecc:7dbf/64 scope link
            valid_lft forever preferred_lft forever
5: wlan3@enx00e04cc57dbf: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default qlen 1000
    link/ether 00:00:00:c5:7d:bf brd ff:ff:ff:ff:ff:ff
    inet 10.0.1.101 brd 10.0.1.255 scope global wlan3
        valid_lft forever preferred_lft forever
        inet6 fd01:4616:3133:34:34::/64 scope global
            valid_lft forever preferred_lft forever
            inet6 fe80::2e0:4cff:fecc:7dbf/64 scope link
                valid_lft forever preferred_lft forever
6: wlan3@enx00e04cc57dbf: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default qlen 1000
    link/ether 00:00:00:c5:7d:bf brd ff:ff:ff:ff:ff:ff
    inet 10.0.1.102 brd 10.0.1.255 scope global wlan3
        valid_lft forever preferred_lft forever
        inet6 fd01:4616:3133:36:36::/64 scope link
            valid_lft forever preferred_lft forever
            inet6 fe80::2e0:4cff:fecc:7dbf/64 scope link
                valid_lft forever preferred_lft forever
7: wlan13@enx00e04cc57dbf: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default qlen 1000
    link/ether 00:00:00:c5:7d:bf brd ff:ff:ff:ff:ff:ff

```

4. You should now be able to login to the admin console by pointing your lab machine's browser to 192.168.42.11, login is admin password is ece577

Compute	Value	Description
Instances	Used 0 of 10	Instances
vCPUs	Used 0 of 20	vCPUs
RAM	Used 0B of 500GB	RAM
Volumes	Used 0 of 10	Volumes
Volume Snapshots	Used 0 of 10	Volume Snapshots
Volume Storage	Used 0B of 1000GB	Volume Storage
Network		
Floating IPs	Allocated 0 of 50	Floating IPs
Security Groups	Used 1 of 10	Security Groups
Security Group Rules	Used 4 of 100	Security Group Rules
Ports	Used 1 of 100	Ports
Routers	Used 1 of 10	Routers

5. Download a Compute node image from here: [ECE577_SU23_Compute0.ova](#)

6. As before with the Controller, import it into VirtualBox and start it. The shell login is ubuntu and password is ubuntu.

7. Verify that the compute0 node is recognized by going to Admin → Compute → Hypervisors in the OpenStack admin console:

The screenshot shows the 'Hypervisor Summary' page under the 'Compute' tab. At the top, there are four pie charts representing VCPU Usage (Used 0 of 8), Memory Usage (Used 10B of 5.0GB), Local Storage Usage (Used 0Bytes of 820B), and Local Disk Usage (Used 0Bytes of 31GB). Below these are two tabs: 'Hypervisor' (selected) and 'Compute Host'. A table lists two hosts:

Hostname	Type	VCPUs (used)	VCPUs (total)	RAM (used)	RAM (total)	Local Storage (used)	Local Storage (total)	Instances
compute0	QEMU	0	4	512MB	1.9GB	0Bytes	31GB	0
controller	QEMU	0	4	512MB	3.8GB	0Bytes	31GB	0

This screenshot shows the same 'Hypervisor Summary' page after a configuration change. The memory usage statistics have updated:

- compute0: Used 10B of 11.6GB
- controller: Used 0B of 124GB

The rest of the interface and data remains the same as the first screenshot.

Step C - Creating the Virtual Networks

Now that you have added a Compute Node to the Data Center, it's time to start adding Virtual Machines. First, you will have to create a few new networks for these Virtual Machines.

Network Creation:

1. Log back in to the OpenStack console. Click on Admin, then click on Network, then click on Networks. Click the +Create Network button
2. Name the network ece577_web. The project should be demo, the Provider network type should be Geneve, with Geneve Segment ID 7780. Enable Admin

State, Shared, and Create Subnet should also be checked. Click Next to create the Subnet.

Create Network

Network * [Subnet](#) [Subnet Details](#)

Name
ece577_web

Project *
demo

Provider Network Type * ?
Geneve

Segmentation ID * ?
5577

Enable Admin State ?

Shared

External Network

Create Subnet

Availability Zone Hints ?

MTU ?
 ▲ ▼

[Cancel](#) [« Back](#) [Next »](#)

3. Name the subnet ece577_web_subnet. The address range should be

Create Network

Network * Subnet Subnet Details

Name: ece577_web

Project *: demo

Provider Network Type *: Geneve

Segmentation ID *: 7780

Enable Admin State ?

Shared

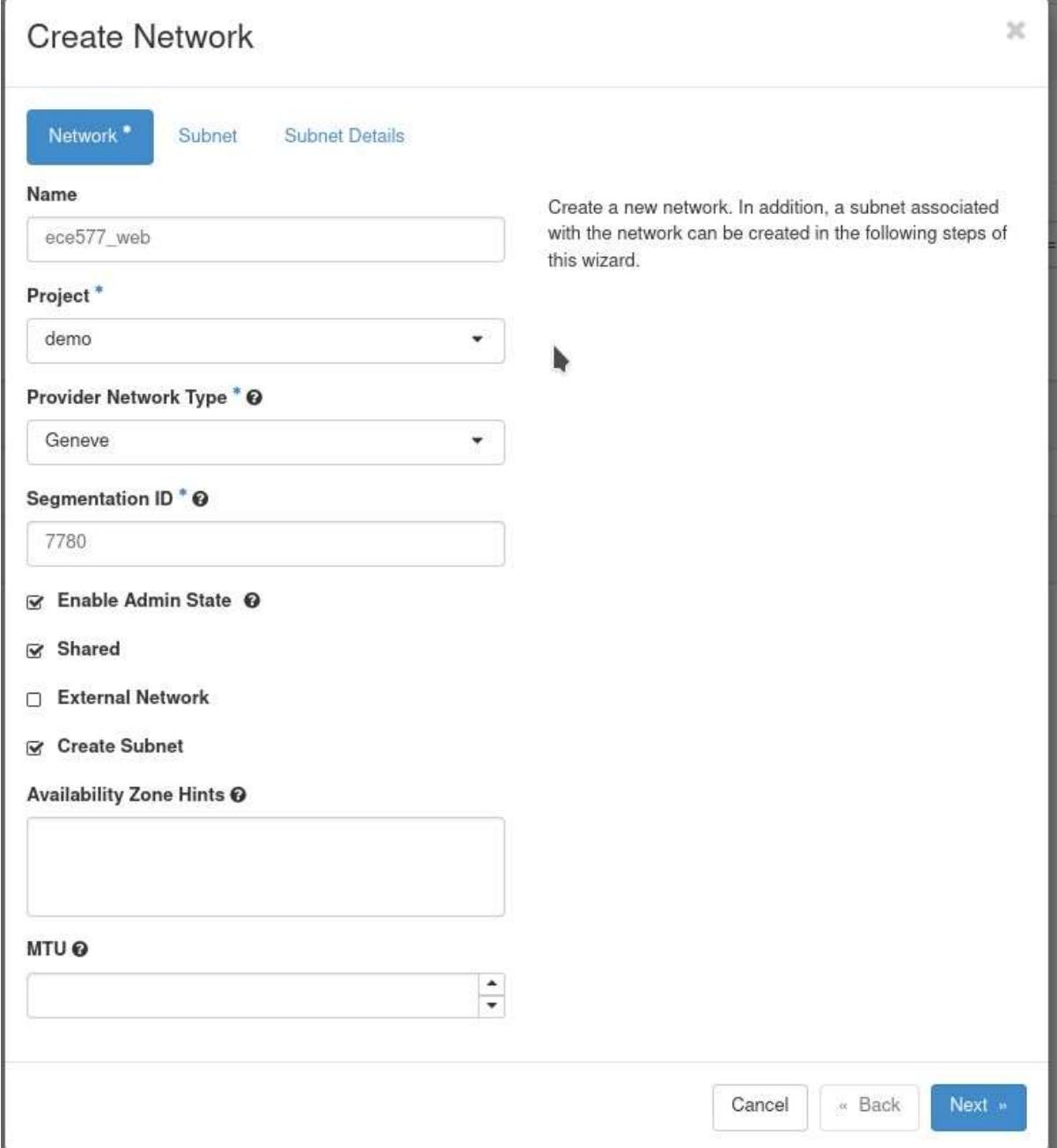
External Network

Create Subnet

Availability Zone Hints:

MTU ?

Cancel « Back Next »



10.77.80.0/24. The Gateway IP (we'll get there in a minute) will be 10.77.80.1.
Click Next.

Create Network

Network *

Subnet

Subnet Details

Subnet Name

ece577_web_subnet

Network Address Source

Enter Network Address manually

Network Address ?

10.77.80.0/24

IP Version

IPv4

Gateway IP ?

10.77.80.1

Disable Gateway

Creates a subnet associated with the network. You need to enter a valid "Network Address" and "Gateway IP". If you did not enter the "Gateway IP", the first value of a network will be assigned by default. If you do not want gateway please check the "Disable Gateway" checkbox. Advanced configuration is available by clicking on the "Subnet Details" tab.

[Cancel](#)

[« Back](#)

[Next »](#)

Create Network

Network *

Subnet

Subnet Details

Subnet Name

ece577_web_subnet

Network Address Source

Enter Network Address manually

Creates a subnet associated with the network. You need to enter a valid "Network Address" and "Gateway IP". If you did not enter the "Gateway IP", the first value of a network will be assigned by default. If you do not want gateway please check the "Disable Gateway" checkbox. Advanced configuration is available by clicking on the "Subnet Details" tab.

Network Address ⓘ

10.77.80.0/24

IP Version

IPv4

Gateway IP ⓘ

10.77.80.1

Disable Gateway

Cancel

« Back

Next »

4. Under Subnet Details, add the subnet range 10.77.80.30,10.77.80.250 and name servers (one per line): 10.77.53.53, 10.77.53.153, 1.1.1.1. Click Create.

Create Network



Network * Subnet

Subnet Details

Enable DHCP

Specify additional attributes for the subnet.

Allocation Pools ?

10.77.80.30,10.77.80.250

DNS Name Servers ?

10.77.53.53

10.77.53.153

1.1.1.1

Host Routes ?

Cancel

« Back

Create

Create Network

Network * Subnet

Subnet Details

Enable DHCP

Specify additional attributes for the subnet.

Allocation Pools ?

10.77.80.30,10.77.80.250

DNS Name Servers ?

10.77.53.53

|10.77.53.153

1.1.1.1

Host Routes ?

Cancel

« Back

Create

5. Repeat step 2, creating another network called ece577_dns. Provider network type should be Geneve, with Segment ID 7753. Project should be demo, Enable Admin State, Shared, and Create Subnet should also be checked. Click Next to create the Subnet.

Create Network



Network *

Subnet

Subnet Details

Name

ece577_dns

Create a new network. In addition, a subnet associated with the network can be created in the following steps of this wizard.

Project *

demo

Provider Network Type * ?

Geneve

Segmentation ID * ?

7753

Enable Admin State ?

Shared

External Network

Create Subnet

Availability Zone Hints ?

MTU ?

Cancel

« Back

Next »

6. Repeat step 3, Name the subnet ece577_dns_subnet. The address range should be 10.77.53.0/24. The Gateway IP (we'll get there in a minute) will be 10.77.53.1. Click Next.

Create Network



Network *

Subnet

Subnet Details

Subnet Name

ece577_dns_subnet

Network Address Source

Enter Network Address manually

Creates a subnet associated with the network. You need to enter a valid "Network Address" and "Gateway IP". If you did not enter the "Gateway IP", the first value of a network will be assigned by default. If you do not want gateway please check the "Disable Gateway" checkbox. Advanced configuration is available by clicking on the "Subnet Details" tab.

Network Address ?

10.77.53.0/24

IP Version

IPv4

Gateway IP ?

10.77.53.1

Disable Gateway

Cancel

« Back

Next »

7. Configure the same subnet details as above, but with range 10.77.53.53,10.77.53.53 for the first line in the Allocation Pools box and 10.77.53.153,10.77.53.153 as the second line. This will ensure the generated VM will only have one of those two addresses.

Create Network



Network *

Subnet

Subnet Details

Enable DHCP

Specify additional attributes for the subnet.

Allocation Pools

10.77.53.53,10.77.53.53
10.77.53.153,10.77.53.153

DNS Name Servers

Host Routes

[Cancel](#)

[« Back](#)

[Create](#)

8. Now, you should now have 2 internal networks. We next need to create the gateways for these networks. Under Project → Networks → Routers, click router1. Then click the Interfaces tab. Click +AddInterface.

Project	Network Name	Subnets Associated	Shared	External	Status	Admin State	Availability Zones	Actions
demo	ece577_web	ece577_web_subnet 10.77.80.0/24	Yes	No	Active	UP	-	Edit Network
demo	ece577_dns	ece577_dns_subnet 10.77.53.0/24	Yes	No	Active	UP	-	Edit Network

router1

[Overview](#)
[Interfaces](#)
[Static Routes](#)

Displaying 2 items

Name	Fixed IPs	Status	Type	Admin State	Actions
(15bd7528-0a53)	• 10.0.0.1	Active	Internal Interface	UP	Delete Interface
(a4329c69-eaac)	• 172.24.4.54	Active	External Gateway	UP	Delete Interface

Displaying 2 items

Project / Network / Routers / router1

Compute > router1

Volumes >

Network > Overview Interfaces Static Routes

+ Add Interface Delete Interfaces

Name	Fixed IPs	Status	Type	Admin State	Actions
(10939c18-1eea)	• fd7d7af4-90ce-1	Active	Internal Interface	UP	<button>Delete Interface</button>
(20d699f-f3e3)	• 192.168.42.183 • 2001:db8:1	Active	External Gateway	UP	<button>Delete Interface</button>
(b6096af7-d3e4)	• 10.0.0.1	Active	Internal Interface	UP	<button>Delete Interface</button>

- In the drop down, select your web subnet, and add the 10.77.80.1 address. Click Submit.

Add Interface

Subnet *

ece577_web: 10.77.80.0/24 (ece577_web_su...)

IP Address (optional) •

10.77.80.1

Description:

You can connect a specified subnet to the router. If you don't specify an IP address here, the gateway's IP address of the selected subnet will be used as the IP address of the new created interface of the router. If the gateway's IP address is in use, you must use a different address which belongs to the selected subnet.

Cancel Submit

Add Interface

Subnet *

ece577_web: 10.77.80.0/24 (ece577_web_su...)

IP Address (optional) ?

10.77.80.1

Description:

You can connect a specified subnet to the router.

If you don't specify an IP address here, the gateway's IP address of the selected subnet will be used as the IP address of the newly created interface of the router. If the gateway's IP address is in use, you must use a different address which belongs to the selected subnet.

Cancel Submit

10. Repeat for your dns subnet with the 10.77.53.1 address. At this point, we now have two software defined networks on a software defined router. Next, we will need to add some Vms.

Name	IP Address	Status	Type	Admin State	Actions
(15bd7528-0a50)			Internal Interface	UP	<button>Delete Interface</button>
(3ef668a0-b44c)	10.77.80.1	Active	Internal Interface	UP	<button>Delete Interface</button>
(e4f29fc9-eaae)	172.24.4.54	Active	External Gateway	UP	<button>Delete Interface</button>

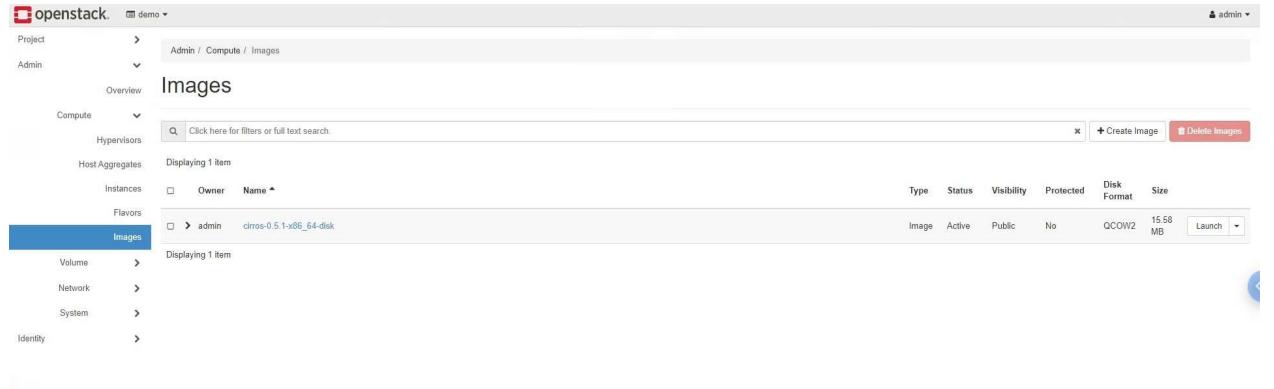
Step D - Creating VMs

Downloading a Base Image

To create VMs, we first need to have a base OS image to build the VMs from. We will

fetch the latest Ubuntu LTS cloud image.

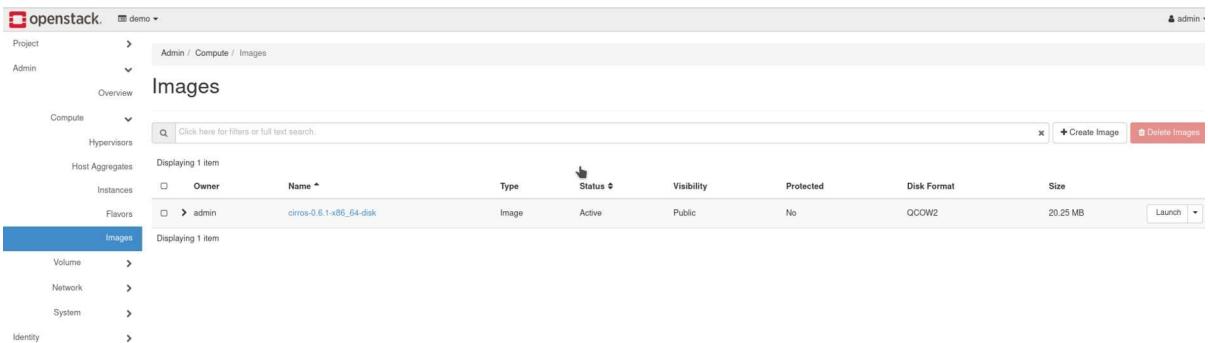
1. In your host (or wherever you are accessing the web console from) go to:
<https://cloud-images.ubuntu.com/jammy/current/jammy-server-cloudimg-amd64.img>
2. Once the image file is downloaded, go to Admin → Compute → Images in the OpenStack web console:



The screenshot shows the OpenStack Compute Images interface. The left sidebar has 'Compute' selected under 'Images'. The main area displays a table with one item:

Type	Status	Visibility	Protected	Disk Format	Size
Image	Active	Public	No	QCOW2	15.58 MB

The image name is 'cirros-0.5.1-x86_64-disk'. There are 'Create Image' and 'Delete Images' buttons at the top right.



This screenshot shows the same OpenStack Compute Images interface as the previous one, but with a different table header. The columns are: Owner, Name, Type, Status, Visibility, Protected, Disk Format, and Size. The data remains the same as in the first screenshot.

3. Click on Create Image. The Name should be Ubuntu_2204, the Image Source should be the jammy-server-cloudimg-amd64.img file you just downloaded, and the format is QCOW2 - QEMU Emulator in the dropdown.

Create Image

Image Details

Specify an image to upload to the Image Service.

Image Name

Image Description

Image Source

File*

Format*

Image Requirements

Kernel

Ramdisk

Architecture

Minimum Disk (GB) **Minimum RAM (MB)**

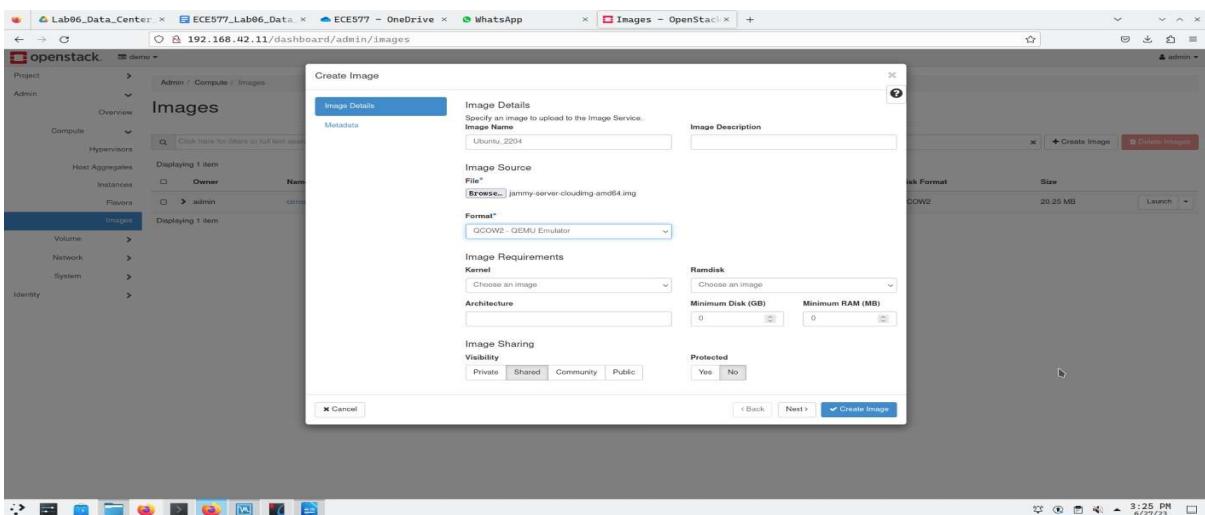
Image Sharing

Visibility Private Shared Community Public

Protected Yes No

Cancel **Back** **Next >** **Create Image**

Click “Create Image”



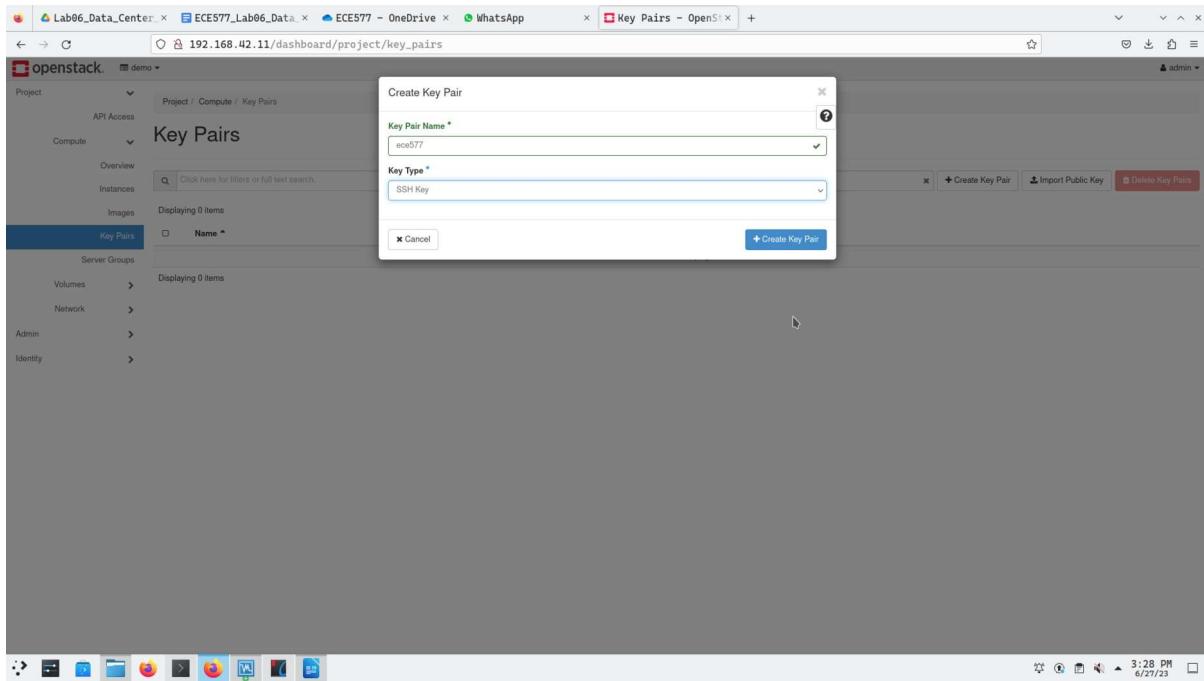
Creating a Key File

4. Before we create VMs, we need to create an encryption key pair for SSH into the machines (by default there is no login password for security purposes!). Log back in to the OpenStack console. Click on Project → Compute → Key Pairs

The screenshot shows the OpenStack dashboard with the URL `192.168.42.11/dashboard/project/key_pairs`. The left sidebar is under the 'Compute' tab, with 'Key Pairs' selected. The main area displays a table with columns 'Name', 'Type', and 'Fingerprint'. A search bar at the top says 'Click here for filters or full text search.' and a button '+ Create Key Pair' is visible.

5. Click +Create Key Pair. For the Key Pair name, use ece577. For Key Pair type, make is SSH. Click Create Key Pair.

The screenshot shows the 'Create Key Pair' dialog box overlaid on the OpenStack dashboard. The dialog has fields for 'Key Pair Name' (set to 'ece577') and 'Key Type' (set to 'SSH Key'). At the bottom right of the dialog is a blue button labeled '+ Create Key Pair'.



6. This should generate a ece577.pem file for you to download. It will likely end up in your Downloads folder. DON'T LOSE THIS FILE!!! IT'S THE ONLY WAY TO LOG IN!!!



Create VM Flavor

A VM flavor is a predefined configuration of vCPUs, RAM, and disk space. Our Ubuntu_2204 image needs 2G of RAM and 4G of disk space, however there isn't a pre-defined flavor that has this so we will make one.

7. Go to Admin → Compute → Flavors and click “Create Flavor”. Name it tiny1G, set vCPUs to 1, RAM to 1024, and Root Disk to 4.

Flavors											
	Flavor Name	VCPUs	RAM	Root Disk	Ephemeral Disk	Swap Disk	RX/TX factor	ID	Public	Metadata	Actions
	cirros256	1	256MB	1GB	0GB	0MB	1.0	c1	Yes	Yes	<button>Update Metadata</button>
	ds1G	1	1GB	10GB	0GB	0MB	1.0	d2	Yes	Yes	<button>Update Metadata</button>
	ds2G	2	2GB	10GB	0GB	0MB	1.0	d3	Yes	Yes	<button>Update Metadata</button>
	ds4G	4	4GB	20GB	0GB	0MB	1.0	d4	Yes	Yes	<button>Update Metadata</button>
	ds512M	1	512MB	5GB	0GB	0MB	1.0	d1	Yes	Yes	<button>Update Metadata</button>
	m1.large	4	8GB	80GB	0GB	0MB	1.0	4	Yes	Yes	<button>Update Metadata</button>
	m1.medium	2	4GB	40GB	0GB	0MB	1.0	3	Yes	Yes	<button>Update Metadata</button>
	m1.micro	1	192MB	1GB	0GB	0MB	1.0	84	Yes	Yes	<button>Update Metadata</button>
	m1.nano	1	128MB	1GB	0GB	0MB	1.0	42	Yes	Yes	<button>Update Metadata</button>
	m1.small	1	2GB	20GB	0GB	0MB	1.0	2	Yes	Yes	<button>Update Metadata</button>
	m1.tiny	1	512MB	1GB	0GB	0MB	1.0	1	Yes	Yes	<button>Update Metadata</button>
	m1.xlarge	8	16GB	160GB	0GB	0MB	1.0	5	Yes	Yes	<button>Update Metadata</button>

8.

Create Flavor

Flavor Information

Name *	<input type="text" value="tiny1G"/>
ID *	<input type="text" value="auto"/>
VCPUs *	<input type="text" value="1"/>
RAM (MB) *	<input type="text" value="1024"/>
Root Disk (GB) *	<input type="text" value="4"/>
Ephemeral Disk (GB)	<input type="text" value="0"/>
Swap Disk (MB)	<input type="text" value="0"/>
RX/TX Factor	<input type="text" value="1"/>

Create Flavor

Flavor Information *

Name *

ID ?

VCPUs *

RAM (MB) *

Root Disk (GB) *

Ephemeral Disk (GB)

Swap Disk (MB)

RX/TX Factor

Flavors define the sizes for RAM, disk, number of cores, and other resources and can be selected when users deploy instances.

Cancel

Create Flavor

Click Create Flavor.

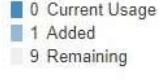
Creating VMs Through Web Console

9. Go to Project → Compute → Instances. Click “Launch Instance”. Create an instance named “web0” and click Next

Launch Instance

Details

Please provide the initial hostname for the instance, the availability zone where it will be deployed, and the instance count. Increase the Count to create multiple instances with the same settings.

Source	Instance Name *	Total Instances (10 Max)
Flavor *	web0	 10%
Networks *	Description	
Network Ports	Availability Zone	0 Current Usage 1 Added 9 Remaining
Security Groups	nova	
Key Pair	Count *	
Configuration	1	
Server Groups		
Scheduler Hints		
Metadata		

Cancel **Next >** **Launch Instance**

Launch Instance

Details

Please provide the initial hostname for the instance, the availability zone where it will be deployed, and the instance count. Increase the Count to create multiple instances with the same settings.

Source *	Project Name	Total Instances (10 Max)
Flavor *	demo	 10%
Networks *	Instance Name *	
Network Ports	web0	0 Current Usage 1 Added 9 Remaining
Security Groups	Description	
Key Pair	Availability Zone	
Configuration	nova	
Server Groups	Count *	
Scheduler Hints	1	
Metadata		

Cancel **Next >** **Launch Instance**

10. Select “Image” as Boot Source from dropdown, make sure Create a New Volume is checked, set Volume Size to 4 and make sure Delete Volume on Instance

Delete is checked. Select the Ubuntu_2204 image and click Next.

Launch Instance

Details

Source

Flavor *

Networks *

Network Ports

Security Groups

Key Pair

Configuration

Server Groups

Scheduler Hints

Metadata

Select Boot Source

Image

Create New Volume

Yes No

Volume Size (GB) *

4

Delete Volume on Instance Delete

Yes No

Allocated

Displaying 1 item

Name	Updated	Size	Format	Visibility
Ubuntu 22.04	6/19/23 5:14 PM	655.13 MB	QCOW2	Shared

Available (2)

Select one

Click here for filters or full text search.

Displaying 2 items

Name	Updated	Size	Format	Visibility
cirros-0.6.1-x86_64-disk	6/15/23 7:50 PM	20.25 MB	QCOW2	Public
Debian12	6/19/23 4:46 PM	335.23 MB	QCOW2	Shared

Displaying 2 items

Cancel **Back** **Next >** **Launch Instance**

Lab06_Data_Center > ECE577_Lab06_Data > ECE577 - OneDrive > (2) WhatsApp > Instances - OpenS | +

openstack demo

Project API Access Project / Compute / Instances

Compute Instances

Instances

Images

Key Pairs

Server Groups

Volumes

Network

Admin

Identity

Launch Instance

Details

Select Boot Source

Image

Create New Volume

Yes No

Volume Size (GB) *

4

Delete Volume on Instance Delete

Yes No

Allocated

Displaying 1 item

Name	Updated	Size	Format	Visibility
Ubuntu_2204	6/27/23 8:26 PM	655.13 MB	QCOW2	Shared

Available (2)

Select one

Click here for filters or full text search.

Displaying 1 item

Name	Updated	Size	Format	Visibility
cirros-0.6.1-x86_64-disk	6/15/23 7:50 PM	20.25 MB	QCOW2	Public

Displaying 1 item

Cancel **Back** **Next >** **Launch Instance**

11. select tiny1G as the Flavor and click Next.

Launch Instance

Details

Source

Allocated

Flavor

Networks *

Name	VCPUS	RAM	Total Disk	Root Disk	Ephemeral Disk	Public
tiny1G	1	1 GB	4 GB	4 GB	0 GB	Yes

Network Ports

Displaying 1 item

Security Groups

Available (13)

Select one

Key Pair

Configuration

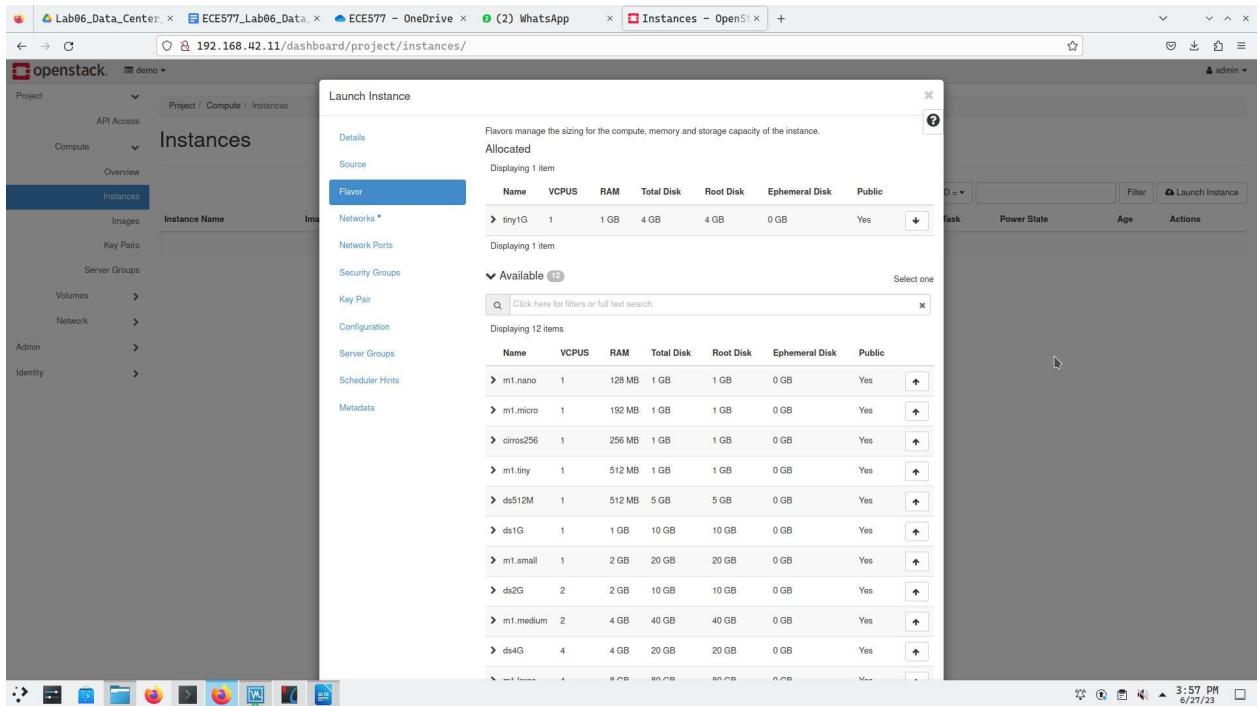
Displaying 13 items

Server Groups

Name	VCPUS	RAM	Total Disk	Root Disk	Ephemeral Disk	Public
m1.nano	1	128 MB	1 GB	1 GB	0 GB	Yes
m1.micro	1	192 MB	1 GB	1 GB	0 GB	Yes
cirros256	1	256 MB	1 GB	1 GB	0 GB	Yes
m1.tiny	1	512 MB	1 GB	1 GB	0 GB	Yes
ds512M	1	512 MB	5 GB	5 GB	0 GB	Yes
ds1G	1	1 GB	10 GB	10 GB	0 GB	Yes
m1.small	1	2 GB	20 GB	20 GB	0 GB	Yes
tiny2G	1	2 GB	4 GB	4 GB	0 GB	Yes
ds2G	2	2 GB	10 GB	10 GB	0 GB	Yes
m1.medium	2	4 GB	40 GB	40 GB	0 GB	Yes

Scheduler Hints

Metadata



Step B - Live Migrating The Machine

Make sure the web0 machine is assigned a floating IP and that you can ping that floating IP from the host.

Start a continuous ping on the floating IP (\$ping -t in Windows). Keep an eye on this. Go to Admin → Compute → Instances and select “Live Migration” from the drop-down next to web0:

You will be prompted to select the new host node, so select compute0

Verify that web0 is now running on compute0

Go back to your host Terminal and see how many (if any) pings were lost during the migration

11. Select the ece577_web Network and click Next.

Launch Instance

Details

Networks provide the communication channels for instances in the cloud.



Source

Flavor

Networks

Network Ports

Security Groups

Key Pair

Configuration

Server Groups

Scheduler Hints

Metadata

Allocated (1)

Select networks from those listed below.

Network	Subnets Associated	Shared	Admin State	Status	Actions
1 > ece577_web	ece577_web_subnet	Yes	Up	Active	

Available (3)

Select at least one network

Network	Subnets Associated	Shared	Admin State	Status	Actions
> private	private-subnet	No	Up	Active	
> ece577_dns	ece577_dns_subnet	Yes	Up	Active	
> shared	shared-subnet	Yes	Up	Active	

Click here for filters or full text search.

Cancel

< Back

Next >

Launch Instance

Launch Instance

Details Networks provide the communication channels for instances in the cloud. You can select ports instead of networks or a mix of both.

Source Allocated 1

Displaying 1 item

Network	Subnets Associated	Shared	Admin State	Status
ece577_web	ece577_web_subnet	Yes	Up	Active

Network Ports Displaying 1 item

Security Groups Available 3 Select one or more

Key Pair Configuration

Server Groups Scheduler Hints

Metadata

Displaying 3 items

Network	Subnets Associated	Shared	Admin State	Status
ece577_dns	ece577_dns_subnet	Yes	Up	Active
shared	shared-subnet	Yes	Up	Active
private	private-subnet	No	Up	Active

Displaying 3 items

[Cancel](#) [Back](#) [Next >](#) [Launch Instance](#)

12. Click Next on Network Ports, Next on Security Groups, and make sure the ece577 Key Pair is selected. Click Launch Instance

Launch Instance

Details A key pair allows you to SSH into your newly created instance. You may select an existing key pair, import a key pair, or generate a new key pair.

Source [Create Key Pair](#) [Import Key Pair](#)

Flavor Allocated

Displaying 1 item

Name	Type	Fingerprint
ece577	ssh	df:94:92:a7:12:2e:8a:41:43:19:3b:e2:0f:63:b1:ab

Network Ports Displaying 1 item

Security Groups Available 0 Select one

Key Pair Configuration

Server Groups

Scheduler Hints

Metadata

Displaying 0 items

Name	Type	Fingerprint
------	------	-------------

No items to display.

Displaying 0 items

[Cancel](#) [Back](#) [Next >](#) [Launch Instance](#)

Launch Instance

Details Networks provide the communication channels for instances in the cloud. You can select ports instead of networks or a mix of both.

Source Allocated 1

Displaying 1 item

Network	Subnets Associated	Shared	Admin State	Status
ece577_web	ece577_web_subnet	Yes	Up	Active

Flavor

Network Ports

Security Groups

Key Pair

Select one or more

Configuration

Server Groups

Scheduler Hints

Metadata

Displaying 3 items

Network	Subnets Associated	Shared	Admin State	Status
ece577_dns	ece577_dns_subnet	Yes	Up	Active
shared	shared-subnet	Yes	Up	Active
private	private-subnet	No	Up	Active

Displaying 3 items

< Cancel < Back Next > Launch Instance

13. It may take a few minutes (it is, after all, a nested VM, running on a loopback disk image mounted on a virtual disk, so we probably couldn't be less efficient if we tried!) but if everything is successful, it should be running!

openstack demo admin

Project API Access Compute Instances

Instances

Displaying 1 item

Instance Name	Image Name	IP Address	Flavor	Key Pair	Status	Availability Zone	Task	Power State	Age	Actions
web0	-	10.77.80.64	m1.tiny2G	ece577	Active	nova	None	Running	1 minute	Create Snapshot

Displaying 1 item

Volumes Network Admin Identity

openstack demo admin

Project API Access Compute Instances

Instances

Displaying 1 item

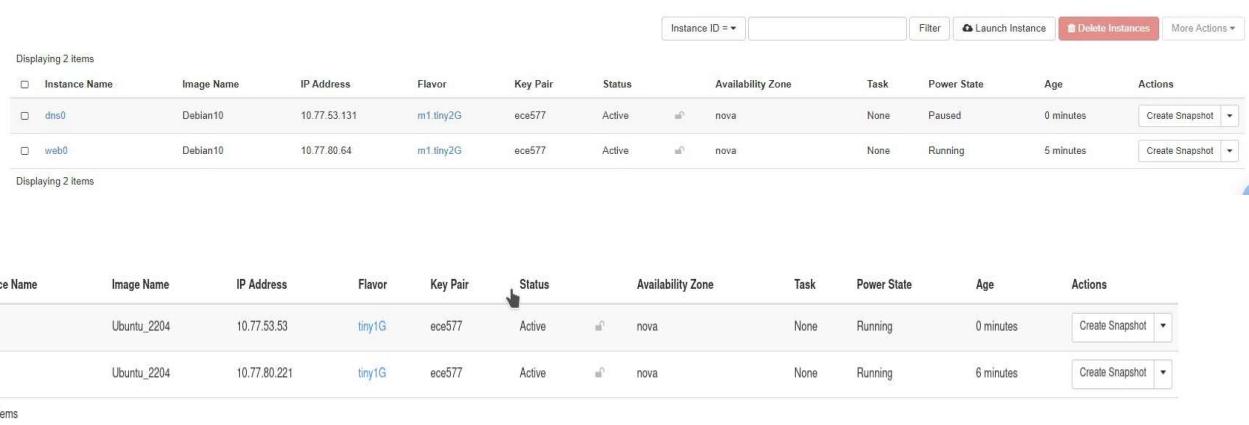
Instance Name	Image Name	IP Address	Flavor	Key Pair	Status	Availability Zone	Task	Power State	Age	Actions
web0	-	10.77.80.221	tiny1G	ece577	Active	nova	None	Running	0 minutes	Create Snapshot

Displaying 1 item

Volumes Network Admin Identity

14. Repeat the steps to create another VM, but this time name it dns0 and place it on the ece577_dns Network. (Your IPs will vary as they are assigned by the OpenStack DHCP server)

Instances



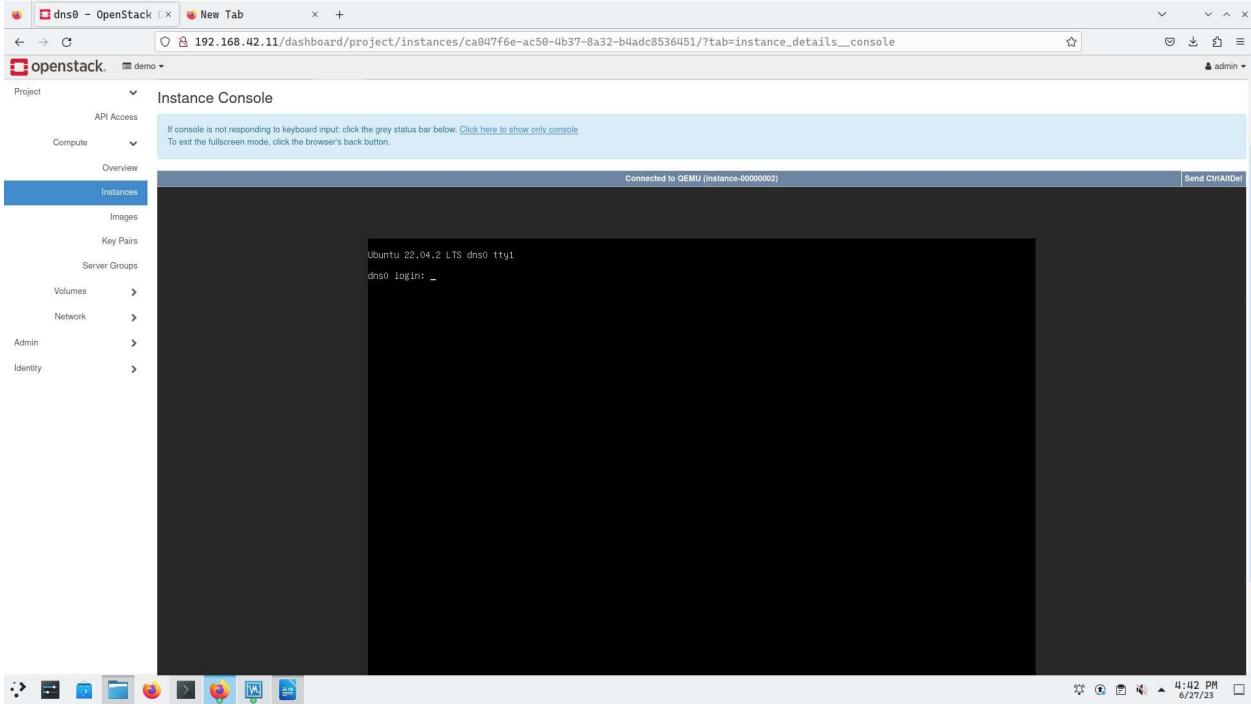
The screenshot shows the OpenStack Instances page with two VMs listed:

Instance Name	Image Name	IP Address	Flavor	Key Pair	Status	Availability Zone	Task	Power State	Age	Actions
dns0	Debian10	10.77.53.131	m1.tiny2G	ece577	Active	nova	None	Paused	0 minutes	<button>Create Snapshot</button>
web0	Debian10	10.77.80.64	m1.tiny2G	ece577	Active	nova	None	Running	5 minutes	<button>Create Snapshot</button>

Instance Name	Image Name	IP Address	Flavor	Key Pair	Status	Availability Zone	Task	Power State	Age	Actions
dns0	Ubuntu_2204	10.77.53.53	tiny1G	ece577	Active	nova	None	Running	0 minutes	<button>Create Snapshot</button>
web0	Ubuntu_2204	10.77.80.221	tiny1G	ece577	Active	nova	None	Running	6 minutes	<button>Create Snapshot</button>

Accessing the VMs

There isn't much to do at this point, as the machine, while running, is not practically accessible due to the fact that the root/user accounts are locked, and there is no default password! (There is also NO route from your compute machine to the internal network yet either!) You can click on the VM name under Instances and click on the Console tab to see the VM's console, but at this point, it won't do you much good. The only way to log in to your machine is to access it via SSH. However this requires three things, the SSH Key (which you should have in your ece577.pem file), a route to the machine (which you currently don't as the ece577_web and ece577_dns networks are internal, tenant networks), and a security rule that allows SSH (The default rule blocks all inbound traffic!)



Assigning a Public Floating IP

15. Under Project → Compute → Instances there is a drop-down menu with options for your VMs. Pick the web0 VM first and select the “Associate Floating IP” option

Project / Compute / Instances

Instances

Instance Name	Image Name	IP Address	Flavor	Key Pair	Status	Availability Zone	Task	Power State	Age	Actions
dns0	Debian10	10.77.53.131	m1.tiny2G	ece577	Active	nova	None	Paused	9 minutes	<button>Create Snapshot</button>
web0	Debian10	10.77.80.64	m1.tiny2G	ece577	Active	nova	None	Running	14 minutes	<button>Create Snapshot</button>

Displaying 2 items

Instance ID = Filter More Actions

Associate Floating IP
Attach Interface
Detach Interface
Edit Instance
Attach Volume
Detach Volume
Update Metadata
Edit Security Groups
Edit Port Security Groups
Console
View Log
Rescue Instance
Pause Instance
Suspend Instance
Shelve Instance
Reize Instance
Lock Instance
Soft Reboot Instance
Hard Reboot Instance
Shut Off Instance

16. Click the + Next to IP Address and when the IP selection screen pops up, click Allocate IP

Allocate Floating IP

Pool *

public

Description

|

Description:

Allocate a floating IP from a given floating IP pool.

Project Quotas

Floating IP 0 of 50 Used

Cancel Allocate IP

The screenshot shows the OpenStack dashboard for the 'demo' project. The left sidebar has 'Compute' selected under 'Project'. The main area shows two instances: 'dns0' and 'web0'. A modal dialog box titled 'Allocate Floating IP' is open in the center. It contains fields for 'Pool' (set to 'public'), 'Description' (empty), 'DNS Domain' (empty), and 'DNS Name' (empty). Below these fields are 'Cancel' and 'Allocate IP' buttons. The background shows the list of instances with their task, power state, age, and actions.

17. That will bring you back to the original screen and you should then see an IP in the 172.24.4.0/24 network. Click Associate.

Manage Floating IP Associations

IP Address *

Select the IP address you wish to associate with the selected instance or port.

Port to be associated *

18.

Your machine is now externally accessible on the 172.24.4.0/24 network! Now, let's fix the SSH security rule

Network Security Groups

Security Groups are a set of firewall rules that can be applied selectively to individual VM interfaces. The default group allows all outbound traffic but no inbound traffic. While good for a default policy, we would like to access our newly created VMs. Let's create a rule that allows us to do so

19. Click In Project → Network → Security Groups. Click “Create Security Group”
20. Name it “EXT_ACCESS” and click “Create Security Group”

Create Security Group

Name *

Description

Description:

Security groups are sets of IP filter rules that are applied to network interfaces of a VM. After the security group is created, you can add rules to the security group.

Create Security Group

The screenshot shows the OpenStack dashboard with the URL 192.168.42.11/dashboard/project/security_groups/. The left sidebar shows the navigation menu with 'Security Groups' selected. The main content area displays the 'Security Groups' table, which currently shows one item: 'default' with ID '69d0a444-3997-44b4-a15'. A modal window titled 'Create Security Group' is open in the center. It contains fields for 'Name *' (set to 'EXT_ACCESS') and 'Description' (empty). Below the fields is a descriptive text about security groups. At the bottom right of the modal is a blue 'Create Security Group' button. The status bar at the bottom right of the screen shows the time as 4:51 PM and the date as 6/27/23.

20. Click “Add Rule”, select “SSH” under the “Rule” drop down and click “Add”

Add Rule

Rule *

Description ?

Remote * ?

CIDR* ?

Description:

Rules define which traffic is allowed to instances assigned to the security group. A security group rule consists of three main parts:

Rule: You can specify the desired rule template or use custom rules, the options are Custom TCP Rule, Custom UDP Rule, or Custom ICMP Rule.

Open Port/Port Range: For TCP and UDP rules you may choose to open either a single port or a range of ports. Selecting the "Port Range" option will provide you with space to provide both the starting and ending ports for the range. For ICMP rules you instead specify an ICMP type and code in the spaces provided.

Remote: You must specify the source of the traffic to be allowed via this rule. You may do so either in the form of an IP address block (CIDR) or via a source group (Security Group). Selecting a security group as the source will allow any other instance in that security group access to any other instance via this rule.

[Cancel](#)

[Add](#)

22. Click “Add Rule”, select “All ICMP” under the “Rule” drop down and click “Add”:

Displaying 4 items								
<input type="checkbox"/>	Direction	Ether Type	IP Protocol	Port Range	Remote IP Prefix	Remote Security Group	Description	Actions
<input type="checkbox"/>	Egress	IPv4	Any	Any	0.0.0.0/0	-	-	Delete Rule
<input type="checkbox"/>	Egress	IPv6	Any	Any	::/0	-	-	Delete Rule
<input type="checkbox"/>	Ingress	IPv4	ICMP	Any	0.0.0.0/0	-	-	Delete Rule
<input type="checkbox"/>	Ingress	IPv4	TCP	22 (SSH)	0.0.0.0/0	-	-	Delete Rule

Displaying 4 items

Add Rule

Rule *
All ICMP

Description ?

Direction
Ingress

Remote * ?
CIDR

CIDR* ?
0.0.0.0/0

Description:
Rules define which traffic is allowed to instances assigned to the security group. A security group rule consists of three main parts:
Rule: You can specify the desired rule template or use custom rules, the options are Custom TCP Rule, Custom UDP Rule, or Custom ICMP Rule.
Open Port/Port Range: For TCP and UDP rules you may choose to open either a single port or a range of ports. Selecting the "Port Range" option will provide you with space to provide both the starting and ending ports for the range. For ICMP rules you instead specify an ICMP type and code in the spaces provided.
Remote: You must specify the source of the traffic to be allowed via this rule. You may do so either in the form of an IP address block (CIDR) or via a source group (Security Group). Selecting a security group as the source will allow any other instance in that security group access to any other instance via this rule.

Cancel **Add**

23. Go back to Project → Compute → Instances and click on the web0 VM. Click on the Interfaces tab and click the “Edit Security Groups” option under Actions:

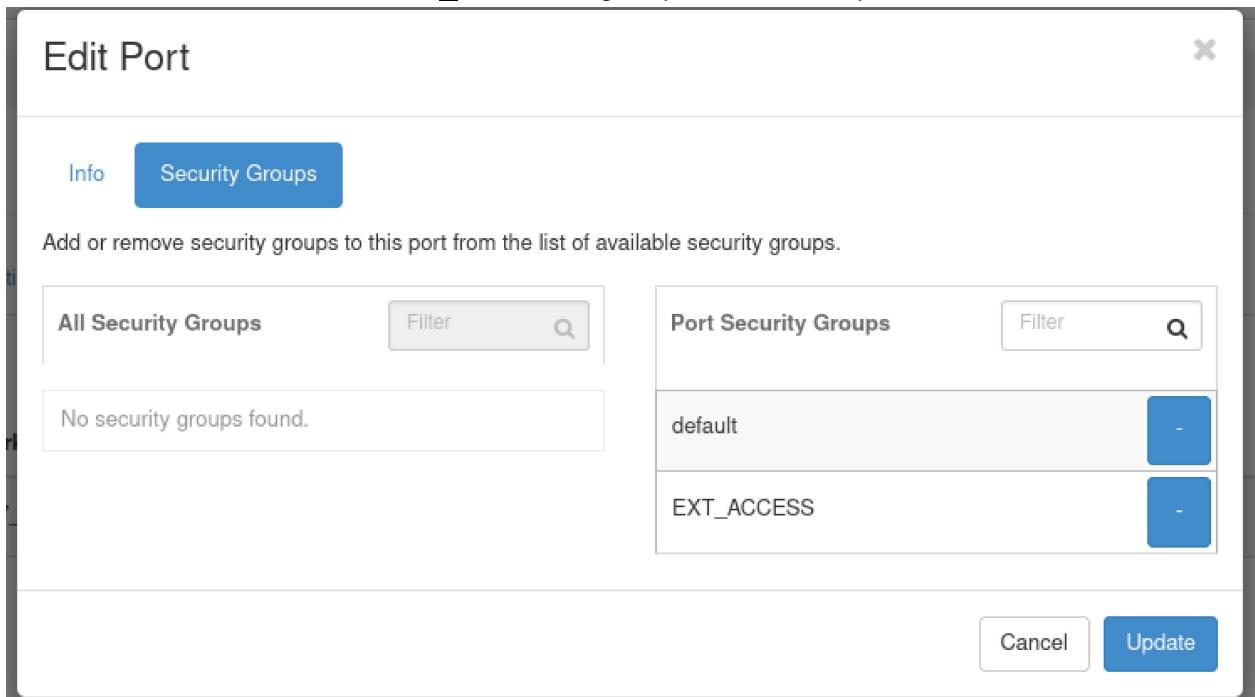
Project / Compute / Instances / web0

web0

Create Snapshot ▾

Overview	Interfaces	Log	Console	Action Log		
Displaying 1 item						
Name	Network	Fixed IPs	MAC Address	Status	Admin State	Actions
(023ef555-2fa2)	ece577_web	• 10.77.80.64	fa:16:3e:75:ec:ff	Active	UP	<input type="button" value="Edit Security Groups"/>
Displaying 1 item						

12.24. Click the + next to the EXT_ACCESS group and click “Update”



Route To Public Subnet

We can now at least SSH into our VM, BUT at this point, only the controller node has access to the 172.24.4.0/24 network. So, to fix this and to allow external devices to SSH into it (like what would happen in a real data center), we need to add routes to this network.

25. Go to your VyOS VM and add a static route to the 172.24.4.0/24 subnet through your Controller's 192.168.42.11 interface (YES, I KNOW THIS SHOULD BE ACCESSIBLE DIRECTLY ON THE PUBLIC NETWORK, BUT BEAR WITH ME HERE)

ECE577_DC_Router [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

```
[edit]
vyos@vyos# set protocols static
Possible completions:
+> arp           Static ARP translation
+> interface-route
                  Interface based static route
+> interface-route6
                  Interface based IPv6 static route
+> route          Static route
+> route6         Static IPv6 route
+> table          Policy route table number

[edit]
vyos@vyos# set protocols static r
route   route6
[edit]
vyos@vyos# set protocols static route
Possible completions:
> <x.x.x.x/x>  Static route
> 0.0.0.0/0

[edit]
vyos@vyos# set protocols static route 172.24.4.0/24 next-hop 192.168.42.11
```

26. Now that your VyOS machine knows how to get to 172.24.4.0/24, inform your host

- a. eid@whateverphobia\$ sudo ip route add 172.24.4.0/24 via 10.0.1.1XX
27. Your host should now have access to the machine. You should be able to ping it at this point, which, if it were in a real datacenter, would have a real, public facing IP address and global reachability!

```
cpidugu@pyrophobia:~$ ping 172.24.4.76
PING 172.24.4.76 (172.24.4.76) 56(84) bytes of data.
64 bytes from 172.24.4.76: icmp_seq=1 ttl=61 time=12.0 ms
64 bytes from 172.24.4.76: icmp_seq=2 ttl=61 time=1.68 ms
64 bytes from 172.24.4.76: icmp_seq=3 ttl=61 time=1.00 ms
64 bytes from 172.24.4.76: icmp_seq=4 ttl=61 time=4.17 ms
^C
--- 172.24.4.76 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3003ms
rtt min/avg/max/mdev = 1.003/4.718/12.019/4.376 ms
cpidugu@pyrophobia:~$ ping 172.24.4.100
PING 172.24.4.100 (172.24.4.100) 56(84) bytes of data.
64 bytes from 172.24.4.100: icmp_seq=1 ttl=61 time=10.7 ms
64 bytes from 172.24.4.100: icmp_seq=2 ttl=61 time=5.57 ms
64 bytes from 172.24.4.100: icmp_seq=3 ttl=61 time=4.13 ms
64 bytes from 172.24.4.100: icmp_seq=4 ttl=61 time=6.86 ms
64 bytes from 172.24.4.100: icmp_seq=5 ttl=61 time=4.12 ms
^C
--- 172.24.4.100 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4006ms
rtt min/avg/max/mdev = 4.116/6.270/10.678/2.428 ms
```

SSH Login

28. You can only log in with your key.
 - b. You should just be able to eid@whateverphobia\$ ssh -i ece577.pem ubuntu@172.24.4.ZZZ (whatever floating IP was assigned) from a terminal to get access.

```
cpidugu@pyrophobia:~/Downloads$ sudo chmod 600 ece577.pem
[sudo] password for cpidugu:
cpidugu@pyrophobia:~/Downloads$ ssh -i ece577.pem ubuntu@172.24.4.76
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.0-75-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

 System information as of Wed Jun 28 17:09:43 UTC 2023

 System load:  0.15380859375   Processes:          85      █
 Usage of /:   38.3% of 3.70GB   Users logged in:     0
 Memory usage: 18%              IPv4 address for ens3: 10.77.80.50
 Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@web0:~$ █
```

29. If you don't want to have to do all this again in the future, you can give your ubuntu user an actual password for logging in through the web console, so go ahead and do that now (\$ sudo passwd ubuntu), but this is actually pretty uncommon to do in a real data center environment, as you want only tenants with the key to have direct access to machines for security reasons.

```
ubuntu@web0:~$ sudo passwd ubuntu
New password:
Retype new password:
passwd: password updated successfully
ubuntu@web0:~$ █
```

30. Repeat this process for the dns0 VM.

```
cpidugu@pyrophobia:~/Downloads$ ssh -i ece577.pem ubuntu@172.24.4.100
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.0-75-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage
[...]
System information as of Wed Jun 28 17:16:01 UTC 2023

System load: 0.26025390625   Processes:          86
Usage of /: 38.3% of 3.70GB  Users logged in:      0
Memory usage: 18%           IPv4 address for ens3: 10.77.53.53
Swap usage:  0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@dns0:~$ sudo passwd ubuntu
New password:
Retype new password:
passwd: password updated successfully
ubuntu@dns0:~$
```

ASSIGNMENT PART 1:

Your VMs are fully operational, you can assign them reachable IPs, but at this point they can't actually access the Internet!

- 1) Your job here in the assignment is to figure out how to get your VMs to be able to access the Internet and it should work whether they have a floating IP assigned or not. Document and explain this in your report. It might work already through

NAT on the ECE577_DC_Gateway, but verify it is.

- 2) Once the Internet is working, install a Web Server on the web0 VM (any one, I don't really care, you could just do \$ python -m SimpleHTTPServer in a directory with an index.html file). Make a security rule that allows only HTTP & HTTPS into your web server

```
ubuntu@web0:~$ sudo python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
10.0.1.36 - - [28/Jun/2023 17:50:15] "GET / HTTP/1.1" 200 -
10.0.1.36 - - [28/Jun/2023 17:50:15] code 404, message File not found
10.0.1.36 - - [28/Jun/2023 17:50:15] "GET /favicon.ico HTTP/1.1" 404 -
10.0.1.36 - - [28/Jun/2023 17:52:13] "GET /index.html HTTP/1.1" 200 -
```



- 3) Install BIND on your dns0 VM. Listen for DNS requests through this VM like back in Lab 2. Don't forget to allow DNS with a security rule here. Reboot any VMs for this take effect if needed.

```
(Ubuntu) 172.24.4.100
options {
    directory "/var/cache/bind";

    // If there is a firewall between you and nameservers you want
    // to talk to, you may need to fix the firewall to allow multiple
    // ports to talk. See http://www.kb.cert.org/vuls/id/800113

    // If your ISP provided one or more IP addresses for stable
    // nameservers, you probably want to use them as forwarders.
    // Uncomment the following block, and insert the addresses replacing
    // the all-0's placeholder.

    [ forwarders {
        8.8.8.8;
        8.8.4.4; ];
    };

    // If BIND logs error messages about the root key being expired,
    // you will need to update your keys. See https://www.isc.org/bind-keys
    //
    dnssec-validation auto;

    listen-on-v6 { any; };

};

~
```

```
ubuntu@dns0:~$ sudo vi /etc/bind/named.conf.options
ubuntu@dns0:~$ sudo systemctl restart bind9
ubuntu@dns0:~$ sudo systemctl restart status.bind9
Failed to restart status.service: Unit status.service not found.
ubuntu@dns0:~$ sudo systemctl status bind9
● named.service - BIND Domain Name Server
   Loaded: loaded (/lib/systemd/system/named.service; enabled; vendor preset: enabled)
     Active: active (running) since Wed 2023-06-28 18:31:35 UTC; 18s ago
       Docs: man:named(8)
    Process: 7292 ExecStart=/usr/sbin/named $OPTIONS (code=exited, status=0/SUCCESS)
   Main PID: 7293 (named)
      Tasks: 3 (limit: 1101)
     Memory: 5.2M
        CPU: 704ms
      CGroup: /system.slice/named.service
              └─7293 /usr/sbin/named -u bind

Jun 28 18:31:36 dns0 named[7293]: network unreachable resolving './NS/IN': 2001:500:1::53#53
Jun 28 18:31:36 dns0 named[7293]: network unreachable resolving './NS/IN': 2001:dc3::35#53
Jun 28 18:31:36 dns0 named[7293]: network unreachable resolving './NS/IN': 2001:500:2d::d#53
Jun 28 18:31:36 dns0 named[7293]: network unreachable resolving './NS/IN': 2001:7fd::1#53
Jun 28 18:31:36 dns0 named[7293]: network unreachable resolving './NS/IN': 2001:7fe::53#53
Jun 28 18:31:36 dns0 named[7293]: network unreachable resolving './NS/IN': 2001:503:ba3e::2:30#53
Jun 28 18:31:36 dns0 named[7293]: network unreachable resolving './NS/IN': 2001:500:a8::e#53
Jun 28 18:31:36 dns0 named[7293]: network unreachable resolving './NS/IN': 2001:500:2f::f#53
Jun 28 18:31:36 dns0 named[7293]: network unreachable resolving './NS/IN': 2001:500:12::d0d#53
Jun 28 18:31:36 dns0 named[7293]: resolver priming query complete: success
ubuntu@dns0:~$
```

STEP D - VMs On ANY Host

The VMs created in the previous part are most likely running on the Controller node since that is where you launched them (and also it's the first node in the pool of

hypervisors). You can check which nodes are running them by going to Admin → Compute→ Hypervisors and looking. We are going to launch two new VMs explicitly on the compute0 node now and investigate how the overlay network is functioning to both.

1. Go to the terminal of your controller VM. As the stack user, you also have access to command line equivalents of all of the web interface stuff we just did. We are going to use this to create two new VMs.
2. Log in to your Controller node as ubuntu. Change into the stack user using the following command:
`ubuntu@controller$ sudo su - stack`
3. Type the following command as stack:
`stack@controller:~/devstack$. openrc admin demo`
4. This give you admin access to all openstack commands and places you in the demo project
5. Type the following to launch a new instance:
`stack@controller:~/devstack$ openstack server create --image Ubuntu_2204 --flavor tiny2G --key-name ece577 --availability-zone nova:compute0:compute0 --network ece577_web web1`
6. Verify that it is running on compute0 by looking at the Hypervisors again
7. Create a Security Rule that allows ICMP if you haven't already. Apply that rule to both web0 and web1.
8. Attempt to ping from web0 to web1. It should be successful

ECE577_SU23_DC_Controller [Running] - Oracle VM VirtualBox

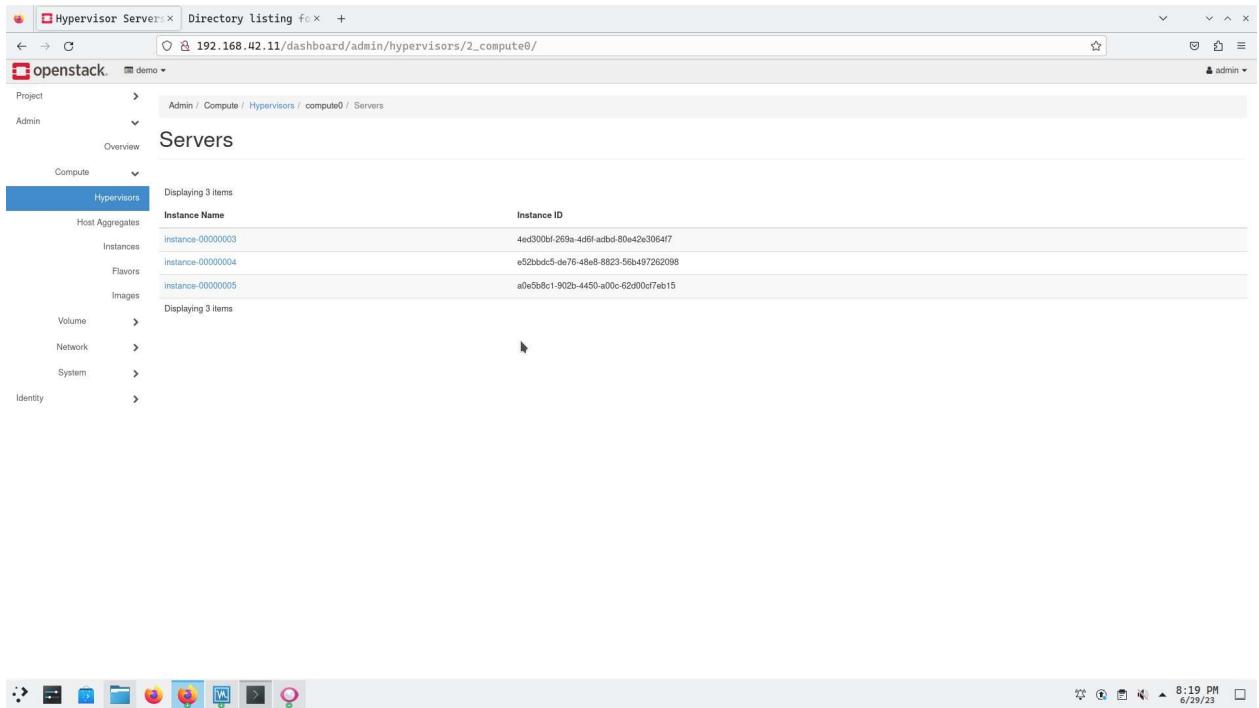
File Machine View Input Devices Help

```
stack@controller:~/devstack$ openstack server create --image Ubuntu_2204 --flavor tiny1G --key-name ece577 --availability-zone nova:compute0:compute0 --network ece577_web web1
```

Field	Value
OS-DCF:diskConfig	MANUAL
OS-EXT-AZ:availability_zone	nova
OS-EXT-SRV-ATTR:host	None
OS-EXT-SRV-ATTR:hypervisor_hostname	None
OS-EXT-SRV-ATTR:instance_name	
OS-EXT-STS:power_state	NOSTATE
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	
accessIPv6	
addresses	
adminPass	sPty7NXer4zs
config_drive	
created	2023-06-30T01:07:27Z
flavor	tiny1G (1315150a-0876-422e-bd60-6df7debef39c)
hostId	
id	a0e5b8c1-902b-4450-a00c-62d00cf7eb15
image	Ubuntu_2204 (8b25ae9f-ee7b-407a-8ecc-57d094142433)
key_name	ece577
name	web1
progress	0
project_id	b10605b9b7ab441dbca06aa35be986fc
properties	
security_groups	name='default'
status	BUILD
updated	2023-06-30T01:07:26Z
user_id	5ad60eb04ca14be0aa920c7eb6376e94
volumes_attached	

```
stack@controller:~/devstack$
```

Right Ctrl



```
ubuntu@web1:~$ ping 172.24.4.76
PING 172.24.4.76 (172.24.4.76) 56(84) bytes of data.
64 bytes from 172.24.4.76: icmp_seq=1 ttl=62 time=5.18 ms
64 bytes from 172.24.4.76: icmp_seq=2 ttl=62 time=7.51 ms
64 bytes from 172.24.4.76: icmp_seq=3 ttl=62 time=7.86 ms
64 bytes from 172.24.4.76: icmp_seq=4 ttl=62 time=5.42 ms
64 bytes from 172.24.4.76: icmp_seq=5 ttl=62 time=5.47 ms
64 bytes from 172.24.4.76: icmp_seq=6 ttl=62 time=5.48 ms
64 bytes from 172.24.4.76: icmp_seq=7 ttl=62 time=5.76 ms
64 bytes from 172.24.4.76: icmp_seq=8 ttl=62 time=5.11 ms
64 bytes from 172.24.4.76: icmp_seq=9 ttl=62 time=6.38 ms
64 bytes from 172.24.4.76: icmp_seq=10 ttl=62 time=5.53 ms
64 bytes from 172.24.4.76: icmp_seq=11 ttl=62 time=5.48 ms
64 bytes from 172.24.4.76: icmp_seq=12 ttl=62 time=5.32 ms
^C
--- 172.24.4.76 ping statistics ---
12 packets transmitted, 12 received, 0% packet loss, time 11059ms
rtt min/avg/max/mdev = 5.112/5.875/7.855/0.867 ms
ubuntu@web1:~$
```

```
ubuntu@web0:~$ ping 172.24.4.100
PING 172.24.4.100 (172.24.4.100) 56(84) bytes of data.
64 bytes from 172.24.4.100: icmp_seq=1 ttl=62 time=8.77 ms
64 bytes from 172.24.4.100: icmp_seq=2 ttl=62 time=10.8 ms
64 bytes from 172.24.4.100: icmp_seq=3 ttl=62 time=6.29 ms
64 bytes from 172.24.4.100: icmp_seq=4 ttl=62 time=5.99 ms
64 bytes from 172.24.4.100: icmp_seq=5 ttl=62 time=5.88 ms
64 bytes from 172.24.4.100: icmp_seq=6 ttl=62 time=5.58 ms
64 bytes from 172.24.4.100: icmp_seq=7 ttl=62 time=5.48 ms
64 bytes from 172.24.4.100: icmp_seq=8 ttl=62 time=6.21 ms

^C--- 172.24.4.100 ping statistics ---
8 packets transmitted, 8 received, 0% packet loss, time 7041ms
rtt min/avg/max/mdev = 5.482/6.869/10.767/1.766 ms
ubuntu@web0:~$ 
```

ASSIGNMENT PART 2

Keep the pings running and attempt to explain the actual path of traffic from web0 to web1. Using tcpdump would be pretty useful here, as well as commands like brctl and ovs-vsctl too, since the software defined networks are implemented using Open vSwitch.

Command : sudo tcpdump -i ens3 -n

```
sudo tcpdump -r web1_to_web0.pcap
sudo tcpdump -r web1_to_web0.pcap
```

```
ubuntu@web0:~$ sudo tcpdump -i ens3 -n
tcpdump: verbose output suppressed, use -v[v] ... for full protocol decode
listening on ens3, link-type EN10MB (Ethernet), snapshot length 262144 bytes
01:34:01.096264 IP 10.77.80.50.22 > 10.0.1.36.50614: Flags [P.], seq 3275167130:3275167238, ack 234636559
, win 502, options [nop,nop,TS val 1674540733 ecr 2359627645], length 108
01:34:01.097870 IP 10.0.1.36.50614 > 10.77.80.50.22: Flags [.], ack 108, win 501, options [nop,nop,TS val
2359628208 ecr 1674540733], length 0
01:34:01.098711 IP 10.77.80.50.22 > 10.0.1.36.50614: Flags [P.], seq 108:144, ack 1, win 502, options [no
p,nop,TS val 1674540735 ecr 2359628208], length 36
01:34:01.099554 IP 10.0.1.36.50614 > 10.77.80.50.22: Flags [.], ack 144, win 501, options [nop,nop,TS val
2359628210 ecr 1674540735], length 0
01:34:01.101713 IP 10.77.80.50.22 > 10.0.1.36.50614: Flags [P.], seq 144:196, ack 1, win 502, options [no
p,nop,TS val 1674540738 ecr 2359628210], length 52
01:34:01.102715 IP 10.0.1.36.50614 > 10.77.80.50.22: Flags [.], ack 196, win 501, options [nop,nop,TS val
2359628213 ecr 1674540738], length 0
01:34:01.103249 IP 10.77.80.50.22 > 10.0.1.36.50614: Flags [P.], seq 196:264, ack 1, win 502, options [no
p,nop,TS val 1674540740 ecr 2359628213], length 68
01:34:01.104378 IP 10.0.1.36.50614 > 10.77.80.50.22: Flags [.], ack 264, win 501, options [nop,nop,TS val
2359628215 ecr 1674540740], length 0
01:34:01.106099 IP 10.77.80.50.22 > 10.0.1.36.50614: Flags [P.], seq 264:332, ack 1, win 502, options [no
p,nop,TS val 1674540743 ecr 2359628215], length 68
01:34:01.106866 IP 10.0.1.36.50614 > 10.77.80.50.22: Flags [.], ack 332, win 501, options [nop,nop,TS val
2359628218 ecr 1674540743], length 0
01:34:01.107868 IP 10.77.80.50.22 > 10.0.1.36.50614: Flags [P.], seq 332:368, ack 1, win 502, options [no
p,nop,TS val 1674540744 ecr 2359628218], length 36
01:34:01.109239 IP 10.0.1.36.50614 > 10.77.80.50.22: Flags [.], ack 368, win 501, options [nop,nop,TS val
2359628219 ecr 1674540744], length 0
01:34:01.130389 IP 10.77.80.50.22 > 10.0.1.36.50614: Flags [P.], seq 368:580, ack 1, win 502, options [no
p,nop,TS val 1674540767 ecr 2359628219], length 212
01:34:01.131497 IP 10.0.1.36.50614 > 10.77.80.50.22: Flags [.], ack 580, win 501, options [nop,nop,TS val
2359628242 ecr 1674540767], length 0
01:34:01.132239 IP 10.77.80.50.22 > 10.0.1.36.50614: Flags [P.], seq 580:616, ack 1, win 502, options [no
p,nop,TS val 1674540769 ecr 2359628242], length 36
01:34:01.133550 IP 10.0.1.36.50614 > 10.77.80.50.22: Flags [.], ack 616, win 501, options [nop,nop,TS val
2359628244 ecr 1674540769], length 0
```

```
ubuntu@web1:~$ sudo tcpdump -i ens3 -n
tcpdump: verbose output suppressed, use -v[v] ... for full protocol decode
listening on ens3, link-type EN10MB (Ethernet), snapshot length 262144 bytes
02:11:19.218000 IP 10.77.80.121.22 > 10.0.1.36.46758: Flags [P.], seq 1865912202:1865912310, ack 193062012, win 502, options [nop,nop,TS val 3592460733 ecr 1218511217], length 108
02:11:19.220072 IP 10.0.1.36.46758 > 10.77.80.121.22: Flags [.], ack 108, win 501, options [nop,nop,TS val 1218511768 ecr 3592460733], length 0
02:11:19.221750 IP 10.77.80.121.22 > 10.0.1.36.46758: Flags [P.], seq 108:144, ack 1, win 502, options [nop,nop,TS val 3592460737 ecr 1218511768], length 36
02:11:19.223068 IP 10.0.1.36.46758 > 10.77.80.121.22: Flags [.], ack 144, win 501, options [nop,nop,TS val 1218511771 ecr 3592460737], length 0
02:11:19.223672 IP 10.77.80.121.22 > 10.0.1.36.46758: Flags [P.], seq 144:196, ack 1, win 502, options [nop,nop,TS val 3592460739 ecr 1218511771], length 52
02:11:19.224715 IP 10.0.1.36.46758 > 10.77.80.121.22: Flags [.], ack 196, win 501, options [nop,nop,TS val 1218511773 ecr 3592460739], length 0
02:11:19.225731 IP 10.77.80.121.22 > 10.0.1.36.46758: Flags [P.], seq 196:264, ack 1, win 502, options [nop,nop,TS val 3592460741 ecr 1218511773], length 68
02:11:19.226817 IP 10.0.1.36.46758 > 10.77.80.121.22: Flags [.], ack 264, win 501, options [nop,nop,TS val 1218511775 ecr 3592460741], length 0
02:11:19.228462 IP 10.77.80.121.22 > 10.0.1.36.46758: Flags [P.], seq 264:332, ack 1, win 502, options [nop,nop,TS val 3592460744 ecr 1218511775], length 68
02:11:19.229654 IP 10.0.1.36.46758 > 10.77.80.121.22: Flags [.], ack 332, win 501, options [nop,nop,TS val 1218511778 ecr 3592460744], length 0
02:11:19.241877 IP 10.77.80.121.22 > 10.0.1.36.46758: Flags [P.], seq 332:552, ack 1, win 502, options [nop,nop,TS val 3592460757 ecr 1218511778], length 220
02:11:19.243039 IP 10.0.1.36.46758 > 10.77.80.121.22: Flags [.], ack 552, win 501, options [nop,nop,TS val 1218511792 ecr 3592460757], length 0
02:11:19.243489 IP 10.77.80.121.22 > 10.0.1.36.46758: Flags [P.], seq 552:588, ack 1, win 502, options [nop,nop,TS val 3592460759 ecr 1218511792], length 36
02:11:19.244430 IP 10.0.1.36.46758 > 10.77.80.121.22: Flags [.], ack 588, win 501, options [nop,nop,TS val 1218511793 ecr 3592460759], length 0
02:11:19.245804 IP 10.77.80.121.22 > 10.0.1.36.46758: Flags [P.], seq 588:768, ack 1, win 502, options [nop,nop,TS val 3592460761 ecr 1218511793], length 180
```

```
ubuntu@web0:~$ sudo tcpdump -r web0_to_web1.pcap
reading from file web0_to_web1.pcap, link-type EN10MB (Ethernet), snapshot length 262144
01:33:12.098162 IP web0.ssh > 10.0.1.36.50614: Flags [P..], seq 3275165534:3275165578, ack 234635659, win 502, options [nop,nop,TS val 1674491735 ecr 2359578475], length 44
01:33:12.099984 IP 10.0.1.36.50614 > web0.ssh: Flags [.], ack 44, win 501, options [nop,nop,TS val 2359579232 ecr 1674491735], length 0
01:33:12.100824 IP web0.ssh > 10.0.1.36.50614: Flags [P..], seq 44:96, ack 1, win 502, options [nop,nop,TS val 1674491737 ecr 2359579232], length 52
01:33:12.102108 IP 10.0.1.36.50614 > web0.ssh: Flags [.], ack 96, win 501, options [nop,nop,TS val 2359579234 ecr 1674491737], length 0
01:33:12.104555 IP web0.ssh > 10.0.1.36.50614: Flags [P..], seq 96:164, ack 1, win 502, options [nop,nop,TS val 1674491741 ecr 2359579234], length 68
01:33:12.105819 IP 10.0.1.36.50614 > web0.ssh: Flags [.], ack 164, win 501, options [nop,nop,TS val 2359579238 ecr 1674491741], length 0
01:33:12.106757 IP web0.ssh > 10.0.1.36.50614: Flags [P..], seq 164:232, ack 1, win 502, options [nop,nop,TS val 1674491743 ecr 2359579238], length 68
01:33:12.107780 IP 10.0.1.36.50614 > web0.ssh: Flags [.], ack 232, win 501, options [nop,nop,TS val 2359579240 ecr 1674491743], length 0
01:33:12.108322 IP web0.ssh > 10.0.1.36.50614: Flags [P..], seq 232:268, ack 1, win 502, options [nop,nop,TS val 1674491745 ecr 2359579240], length 36
01:33:12.109533 IP 10.0.1.36.50614 > web0.ssh: Flags [.], ack 268, win 501, options [nop,nop,TS val 2359579241 ecr 1674491745], length 0
01:33:30.686205 IP web0.40638 > pugot.canonical.com.ntp: NTPv4, Client, length 48
01:33:30.794396 IP pugot.canonical.com.ntp > web0.40638: NTPv4, Server, length 48
ubuntu@web0:~$
```

ECE577 - Lab 06 - Data Center Networks - Part 2 - Live Migration

In this part of the lab, we will show how VMs can be live migrated from any compute host to any other compute host while retaining their network parameters.

Step A - Configuring Live-Migration

Live migration uses SSH to transport the VM state/disk from one machine to the other. To do so requires that the stack user on each host has root SSH access to the other host. Live migration requires host names to work correctly

1. Login as ubuntu controller type:

```
$sudo passwd stack
```

and assign the stack user the password stack and logout

```
ubuntu@controller:~$ sudo passwd stack
New password:
Retype new password:
passwd: password updated successfully
ubuntu@controller:~$ _
```



2. Repeat the above step for compute0

```
ubuntu@compute0:~$ sudo passwd stack
[sudo] password for ubuntu:
New password:
Retype new password:
passwd: password updated successfully
ubuntu@compute0:~$ _
```



3. Log in as the stack user on controller.

```
ECE577_SU23_DC_Controller [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help

* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/advantage

System information as of Fri Jun 30 02:48:56 AM UTC 2023

System load: 0.36962890625    Users logged in: 0
Usage of /: 32.3% of 62.69GB   IPv4 address for br-ex: 172.24.4.1
Memory usage: 57%              IPv4 address for enp0s3: 192.168.42.11
Swap usage: 0%                 IPv4 address for virbr0: 192.168.122.1
Processes: 274

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-its. Check your Internet connection
or proxy settings

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

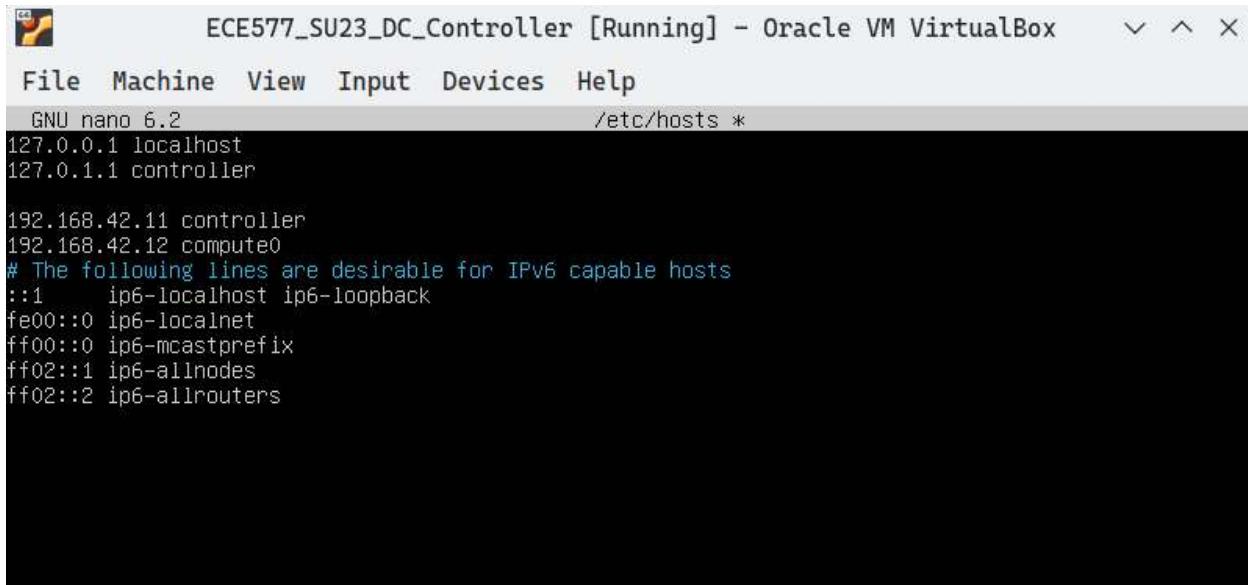
stack@controller:~$
```

4. Issue an \$sudo -i to drop into controller

```
stack@controller:~$ sudo -i
root@controller:~# _
```

5. As root, modify to /etc/hosts file to add entries for both controller and compute0 with the following lines:

```
192.168.42.11 controller
192.168.42.12 compute0
```



```
GNU nano 6.2          /etc/hosts *
127.0.0.1 localhost
127.0.1.1 controller

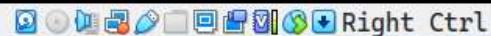
192.168.42.11 controller
192.168.42.12 compute0
# The following lines are desirable for IPv6 capable hosts
::1      ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

6. Next, as root, type:

```
# ssh-keygen -t rsa
```

And hit enter a couple times to create the key file without a password

```
root@controller:~# ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa
Your public key has been saved in /root/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:DMLjexT1T276Yo6zdHMU/jVsrM/SG1FWa7814Fv0b6s root@controller
The key's randomart image is:
+---[RSA 3072]----+
|       ...
|       .o|
|     *.. ..o+o|
|   . *. .=*o+|
|   . S   .*o=|
|   .   +=.o|
|   .   .+ooo+|
|   .oo=ooo|
|   o=.E+=.|
+---[SHA256]----+
root@controller:~#
```



7. Still as root, copy root's public key to the stack user on compute0:

```
# ssh-copy-id -i /root/.ssh/id_rsa.pub stack@compute0
```

The stack user should have the password "stack" unless you changed it.

```

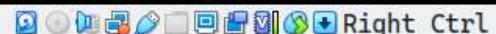
root@controller:~# ssh-copy-id -i /root/.ssh/id_rsa.pub stack@compute0
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_rsa.pub"
The authenticity of host 'compute0 (192.168.42.12)' can't be established.
ED25519 key fingerprint is SHA256:suKUObvkdrGVW6J/FsNWokEc8d0w6CxsUVZEKdywXN8.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
stack@compute0's password:

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'stack@compute0'"
and check to make sure that only the key(s) you wanted were added.

root@controller:~# 

```



8. Repeat steps 1-5 on the compute0 node, changing step 5 to be:
ssh-copy-id -i /root/.ssh/id_rsa.pub stack@controller

ECE577_SU23_Compute0 [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

GNU nano 6.2 /etc/hosts *

```

127.0.0.1 localhost
127.0.1.1 compute0

192.168.42.11 controller
192.168.42.12 compute0_
# The following lines are desirable for IPv6 capable hosts
::1 ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters

```

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location M-U Undo
^X Exit ^R Read File ^Y Replace ^U Paste ^J Justify ^V Go To Line M-E Redo

Right Ctrl

ECE577_SU23_Compute0 [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

```
Ubuntu 22.04.2 LTS compute0 tty1
compute0 login: stack
Password:
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.0-73-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

 System information as of Fri Jun 30 03:10:00 AM UTC 2023

 System load:  0.2568359375      Processes:           187
 Usage of /:   20.3% of 62.69GB   Users logged in:    0
 Memory usage: 92%
 Swap usage:  65%                IPv4 address for enp0s3: 192.168.42.12
                                  IPv4 address for virbr0: 192.168.122.1

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection
or proxy settings.

stack@compute0:~$ _
```

ECE577_SU23_Compute0 [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

```
root@compute0:~# ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa
Your public key has been saved in /root/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:VBVEQuQPZfsaEnM1EOKC9rsxUgbzLLMxNqmK2SK0DVg root@compute0
The key's randomart image is:
+--[RSA 3072]--+
| .+=X*o |
| =..+oo . |
| .=oo+ |
| E . O.o *+.. |
| .. = o BSo.o.. |
| + + o . . +. o |
| o+ . . +. |
| ... . |
+---[SHA256]---+
root@compute0:~#
```

Right Ctrl

```
root@compute0:~# ssh stack@compute0
The authenticity of host 'compute0 (127.0.1.1)' can't be established.
ED25519 key fingerprint is SHA256:suKU0bvkdngVW6J/FsNWokEc8d0w6CxUVZEKdywXN8.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'compute0' (ED25519) to the list of known hosts.
stack@compute0's password:
```

Right Ctrl

Step B - Live Migrating The Machine

1. Make sure the web0 machine is assigned a floating IP and that you can ping that floating IP from the host.
2. Start a continuous ping on the floating IP (\$ping -t in Windows). Keep an eye on this.
3. Go to Admin → Compute → Instances and select “Live Migration” from the drop-down

next

to

web0:

Project	Host	Name	Image Name	IP Address	Flavor	Status	Task	Power State	Age	Actions	
demo	compute0	dns1	Debian10	10.77.53.144	m1.tiny2G	Active		None	Running	6 minutes	<button>Rescue Instance</button>
demo	compute0	web1	Debian10	10.77.80.117	m1.tiny2G	Active		None	Running	8 minutes	<button>Rescue Instance</button>
demo	controller	dns0	-	10.77.53.199	m1.tiny2G	Active		None	Running	25 minutes	<button>Rescue Instance</button>
demo	controller	web0	-	10.77.80.112, 172.24.4.19	m1.tiny2G	Active		None	Running	1 hour, 2 minutes	<button>Rescue Instance</button>

4. You will be prompted to select the new host node, so select compute0

Live Migrate

Current Host
controller

New Host *

compute0

Disk Over Commit

Block Migration

Description:
Live migrate an instance to a specific host.

Cancel **Submit**

5. Verify that web0 is now running on compute0

ECE577_SU23_Compute0 [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

```
System load: 0.9974609375      Users logged in: 1
Usage of /: 32.7% of 62.69GB    IPv4 address for br-ex: 172.24.4.1
Memory usage: 57%              IPv4 address for enp0s3: 192.168.42.11
Swap usage: 0%                 IPv4 address for virbr0: 192.168.122.1
Processes: 280
```

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See <https://ubuntu.com/esm> or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings

```
Last login: Fri Jun 30 03:18:43 2023 from 127.0.0.1
stack@controller:~$ ping 172.24.4.76
PING 172.24.4.76 (172.24.4.76) 56(84) bytes of data.
64 bytes from 172.24.4.76: icmp_seq=1 ttl=63 time=5.48 ms
64 bytes from 172.24.4.76: icmp_seq=2 ttl=63 time=4.03 ms
64 bytes from 172.24.4.76: icmp_seq=3 ttl=63 time=3.42 ms
64 bytes from 172.24.4.76: icmp_seq=4 ttl=63 time=0.949 ms
64 bytes from 172.24.4.76: icmp_seq=5 ttl=63 time=2.64 ms
64 bytes from 172.24.4.76: icmp_seq=6 ttl=63 time=2.77 ms
64 bytes from 172.24.4.76: icmp_seq=7 ttl=63 time=2.55 ms
64 bytes from 172.24.4.76: icmp_seq=8 ttl=63 time=2.06 ms
^C
--- 172.24.4.76 ping statistics ---
8 packets transmitted, 8 received, 0% packet loss, time 7014ms
rtt min/avg/max/mdev = 0.949/2.987/5.482/1.269 ms
stack@controller:~$
```

Right Ctrl

ECE577_SU23_Compute0 [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.0-78-generic x86_64)

* Documentation: <https://help.ubuntu.com>
* Management: <https://landscape.canonical.com>
* Support: <https://ubuntu.com/advantage>

System information as of Fri Jun 30 03:57:51 AM UTC 2023

System load:	0.0830078125	Processes:	189
Usage of /:	20.4% of 62.69GB	Users logged in:	0
Memory usage:	91%	IPv4 address for enp0s3:	192.168.42.12
Swap usage:	65%	IPv4 address for virbr0:	192.168.122.1

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See <https://ubuntu.com/esm> or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings

Last login: Fri Jun 30 03:31:45 UTC 2023 on tty1
ubuntu@compute0:~\$ virsh list --all

Id	Name	State
1	instance-00000003	running
2	instance-00000004	running
3	instance-00000005	running

ubuntu@compute0:~\$

Right Ctrl

Result: The project demonstrates how a data center can be virtualized using overlay networks and microsegmentation, ensuring secure communication between VMs. It also illustrates the effectiveness of live migration in maintaining service continuity.