

## Lab a. Setup a Linux Virtual Machines and *Host only Adaptor*

### Background:

You need three Linux Virtual Machines for the training. You also need to create **ssh keys** for key based authentication.

Keep in mind the practices on Git, Maven, Jenkins, Ansible and Terraform require these VMs. Two Oracle Linux VMs are already created for you and available for download. First VM is **server1**, a GUI based machine and second is **tester1**, without GUI.

### Objectives:

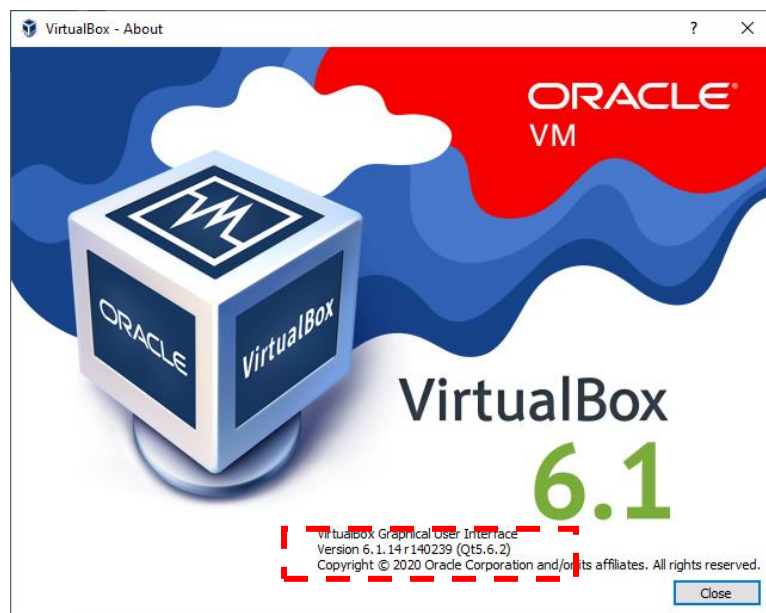
- Download and Install Oracle VM Virtualbox.
- Configure Host Only Adaptor on VirtualBox.
- Import VMs provided by Instructor
- Configure both machines to use **Host only Adaptor** and verify networking.

### Pre-Requisite

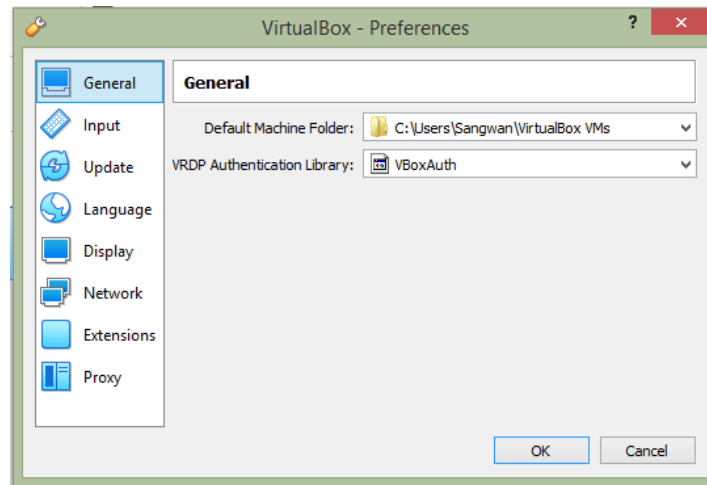
- Two VMs images downloaded as per instruction given from  
<https://tsp-vms.s3.amazonaws.com/server.ova>  
<https://tsp-vms.s3.amazonaws.com/tester1.ova>
- At least 30 GB Space on your Host machine
- Virtualization Enabled on BIOS Settings of Host Machine
- Any existing Hypervisor (such as HyperV on Windows) disabled.

### Sequence 1. Install Oracle VM Virtual Box and Configure a Host Only adaptor.

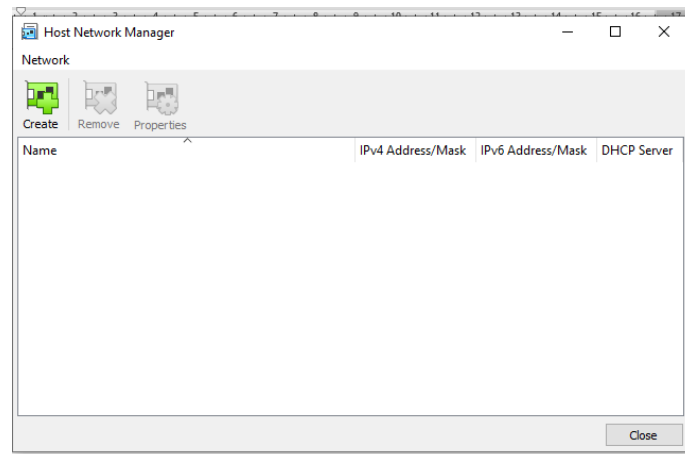
1. Download and install VirtualBox from [www.virtualbox.org](http://www.virtualbox.org). This lab assumes you are using VirtualBox 6.1.



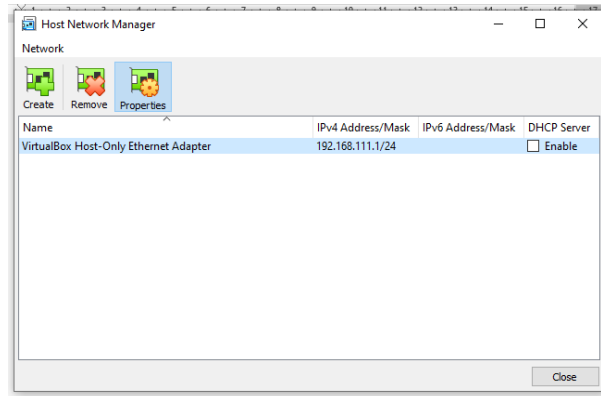
2. Set preferences for storing your VM files. This depends on the Disk Space you have on your computer. To set the preference, click on Preferences (icon) in VirtualBox



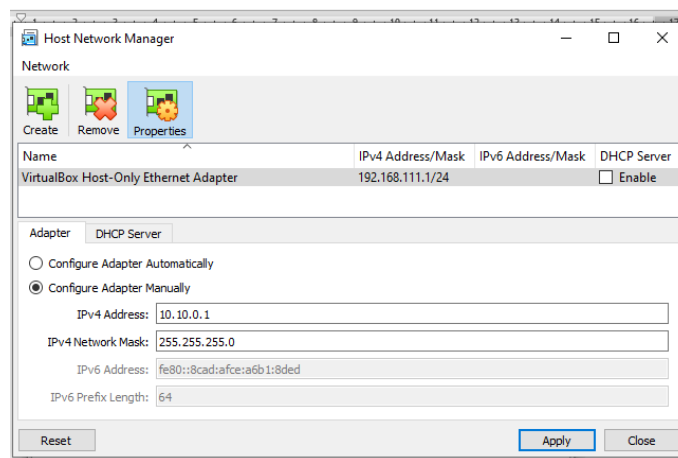
3. In the General Tab of the Preferences screen, choose the Directory in which you want to create the VM related files. Click Ok to close the Preferences screen.
4. Launch Oracle VM virtual Box and open **File -> Host Network Manager**



5. If Oracle VM Virtual box is already installed, you may have an existing Host only Adaptor. Don't change that. Create a new one for this training. For this, click on **Create** button. You may get a warning on permission. Accept that warning and you will see the adaptor is added as screenshot given



6. The adaptor name and other details are selected by default. Click on Properties and Fill/Change IP Address to “10.10.0.1” as given below:

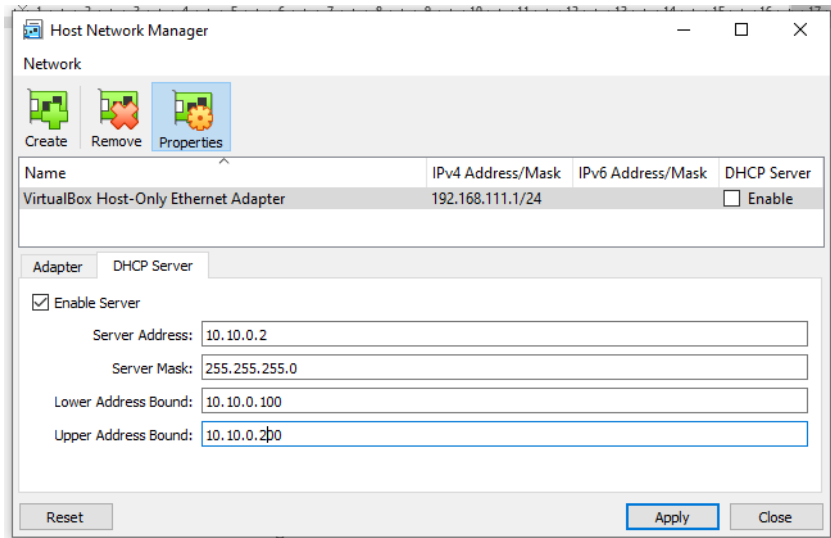


Take note of the **Adaptor Name**. You will need to enable the same in the VM for networking.

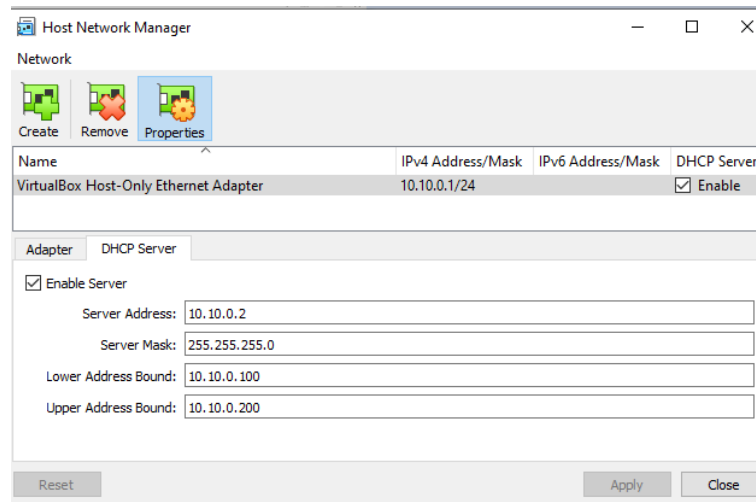
7. Click on the Second Tab “DHCP Server” and Change details to

- Check Box, Enable Server: **Checked**
- Server Address: **10.10.0.2**
- Server Mask: **255.255.255.0**
- Lower Address bound: **10.10.0.100**
- Upper Address bound: **10.10.0.200**

It should be same as screen shot given below.



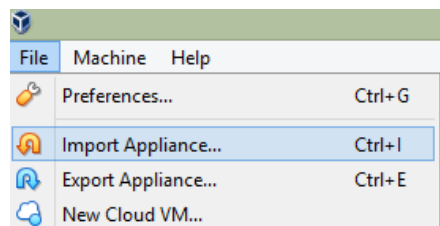
8. Click on **Apply**. You may get a warning. Accept the warning and Close the dialog box. You should see the Adaptor Details Changed as given below



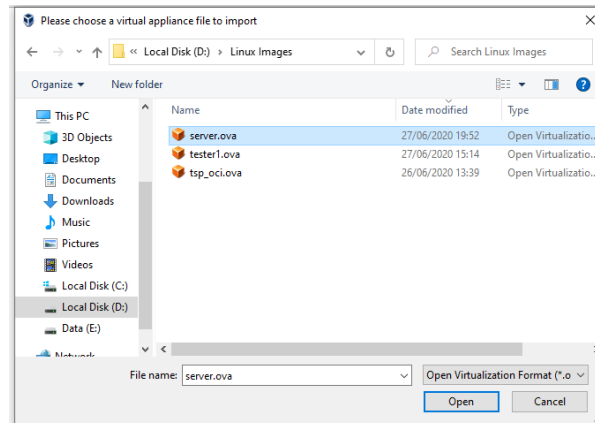
9. Host Network Adaptor is added now.

## Sequence 2. Download and Import Virtual Machine

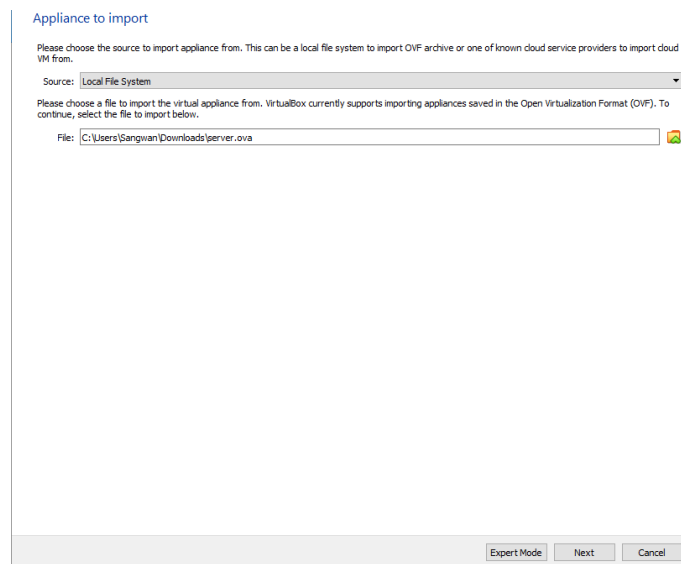
1. Download the VMs from the given URL.
2. To import the downloaded VM Images select **File - > Import Appliance**
3. You will get a pop up in which you need to specify the location of the **VM Image**.



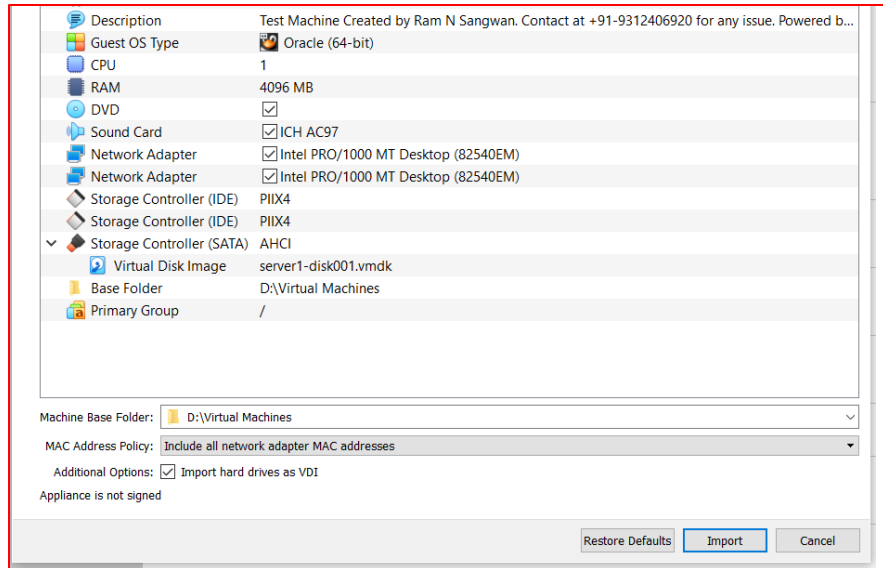
- Click on the Folder Option available at the end of the Text Box and select the downloaded VM Image, **server.ova**.



- Click Open. In the resulting screen click Next.



- In the Next Screen, select **Include All Network Adaptor MAC addresses** under **MAC Address policy**.



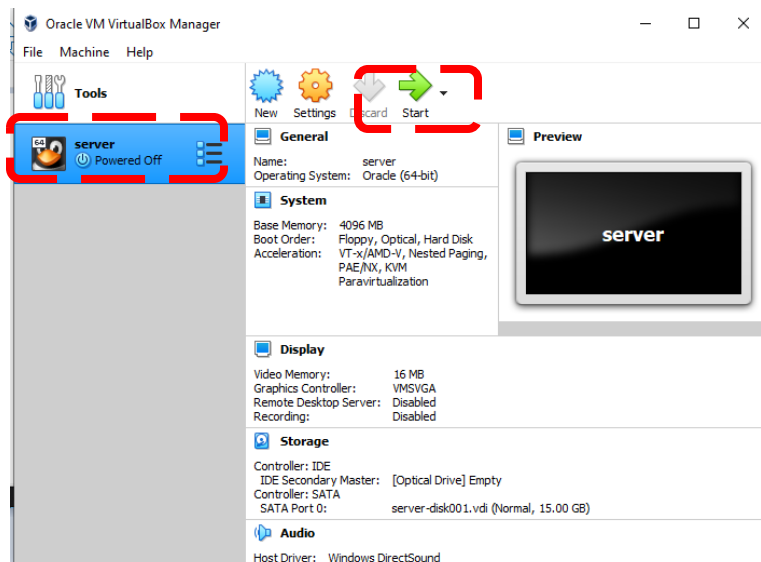
## 7. Click on **Import**

*If you don't see the Guest OS Type as Oracle (64-bit), just check in the BIOS of your computer and ensure VT is enabled for the CPU.*

*You need to shut down and boot your computer and use the F2/F8 key to get to the BIOS options. The Function Key can vary from computer to computer.*

*It is also required that no other Hypervisor/Docker based software (e.g HyperV on Windows) is enabled on your Computer, you need to disable it so that Virtual Box can use the Virtualization features of your computer.*

## 8. Once the VM Image is imported, the VM is available and listed in VirtualBox.



## 9. Click on the VM and Click on the Start Button in VirtualBox window. If the machine fails to start, ensure that:

- You have latest Virtual Box extension pack installed for your operating system. If not installed already, download it from

[https://download.virtualbox.org/virtualbox/6.1.14/Oracle\\_VM\\_VirtualBox\\_Extension\\_Pack-6.1.14.vbox-extpack](https://download.virtualbox.org/virtualbox/6.1.14/Oracle_VM_VirtualBox_Extension_Pack-6.1.14.vbox-extpack)

- b. You have at least 30GB space free on your system.
  - c. Hyper-V is disabled on Windows system.
  - d. Intel VM technology/Virtualization is enabled in Bios.
10. Once the VM boots up, enter the username and password to login.

Username – **root**

Password – **oracle**

**OR**

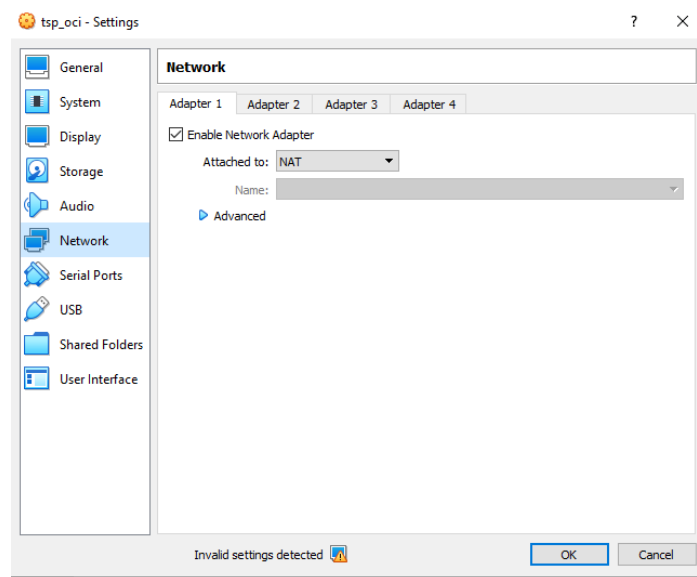
Username – **oracle**

Password – **oracle**

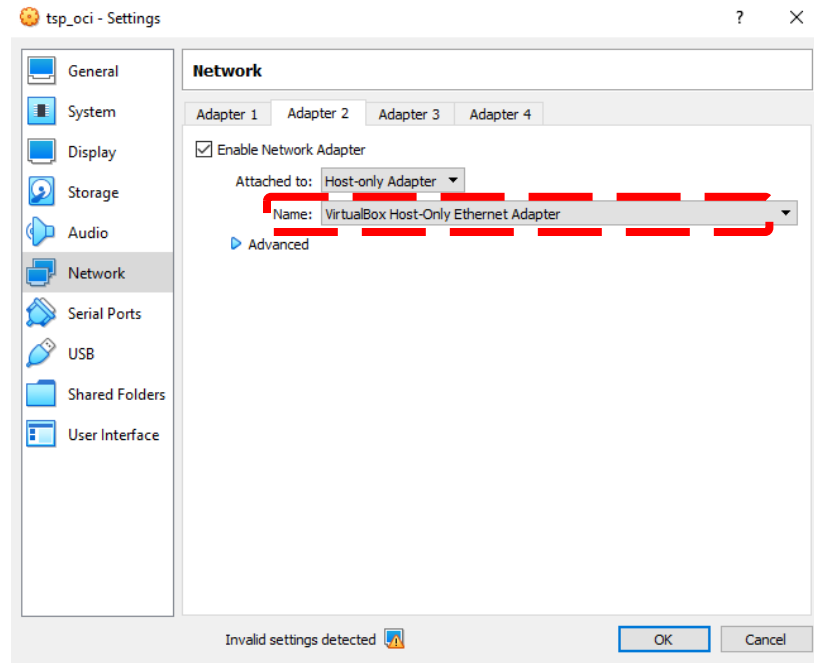
11. Similarly Import the Second VM **tester1**

*You can do all the practices related to OCI CLI, Ansible, Docker and Kubernetes on these Linux VMs, so that you don't need to transfer the files between your VM and Laptop.*

12. You need to ensure your Linux VM has internet Access. For that, open the Firefox browser (**Applications -> Firefox**) and enter a website such as [www.theskillpedia.com](http://www.theskillpedia.com).
13. Disconnect from **Oracle VPN** if needed.
14. Right Click on the First Machine, **server**, and Click on **Settings** to open settings dialog box. Click on **Network**. It should show the dialogue as given below



15. Click on Second Tab. Ensure that checkbox “**Enable Network Adaptor**” is checked and from Attached to drop down list select “**Host only Adaptor**”. Your adaptor added will automatically appear in Name drop down list. Refer the Screen shot below.



16. Click OK to close the dialog box.

17. Follow the steps 12, 13 and 14 for the second machine **tester1**.

18. Verify the IP addresses on both VMs. For this, start both VMs and login as **root** user with **oracle** password. Now Check the IP on both VMs with

On **server**

**# ip a**

```

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 pfifo_fast state UP group default qlen 1000
link/ether 08:00:27:6e:f1:12 brd ff:ff:ff:ff:ff:ff
inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s3
valid_lft 86225sec preferred_lft 86225sec
inet6 fe80::e0cc:1e78:debc:589d/64 scope link noprefixroute
valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
link/ether 08:00:27:1e:53:ea brd ff:ff:ff:ff:ff:ff
inet 10.10.0.100/24 brd 10.10.0.255 scope global noprefixroute enp0s8
valid_lft forever preferred_lft forever
inet6 fe80::a48f:f0f3:74cf:d229/64 scope link noprefixroute
valid_lft forever preferred_lft forever
4: virbr0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default qlen 1000
link/ether 52:54:00:cd:ec:7c 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
5: virbr0-nic: <BROADCAST,MULTICAST> mtu 1500 qdisc pfifo_fast master virbr0 state DOWN group default qlen 1000
link/ether 52:54:00:12:35:70 brd ff:ff:ff:ff:ff:ff

```

On **tester1**

**# ip a**



```

[root@tester1 ~]# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 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 scope host lo
        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 pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:f3:6e:64 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s3
        valid_lft 85756sec preferred_lft 85756sec
    inet6 fe80::1f74:37ce:d42e:c040/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:d4:92:81 brd ff:ff:ff:ff:ff:ff
    inet 10.10.0.101/24 brd 10.10.0.255 scope global noprefixroute enp0s8
        valid_lft forever preferred_lft forever
    inet6 fe80::44c4:80a7:729f:2aac/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
[root@tester1 ~]# _

```

#### 19. Ping from **server** to **tester1**

```

[root@server ~]# ping tester1
PING tester1 (10.10.0.101) 56(84) bytes of data.
64 bytes from tester1 (10.10.0.101): icmp_seq=1 ttl=64 time=1.21 ms
64 bytes from tester1 (10.10.0.101): icmp_seq=2 ttl=64 time=0.584 ms
^C
--- tester1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 0.584/0.897/1.210/0.313 ms
[root@server ~]#

```

#### 20. Ping from **tester1** to **server**

```

[root@tester1 ~]# ping server
PING server (10.10.0.100) 56(84) bytes of data.
64 bytes from server (10.10.0.100): icmp_seq=1 ttl=64 time=0.808 ms
64 bytes from server (10.10.0.100): icmp_seq=2 ttl=64 time=1.06 ms
^C
--- server ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1018ms
rtt min/avg/max/mdev = 0.808/0.935/1.063/0.131 ms
[root@tester1 ~]# _

```

21. Now Host Adaptor is configured on both machines. Networking is also setup between these two machines. Therefore, you should be able to ping **server** from **tester1** and **tester1** from **server**.