

Set up SEEDLAB at server nx.cs.uwindsor.ca

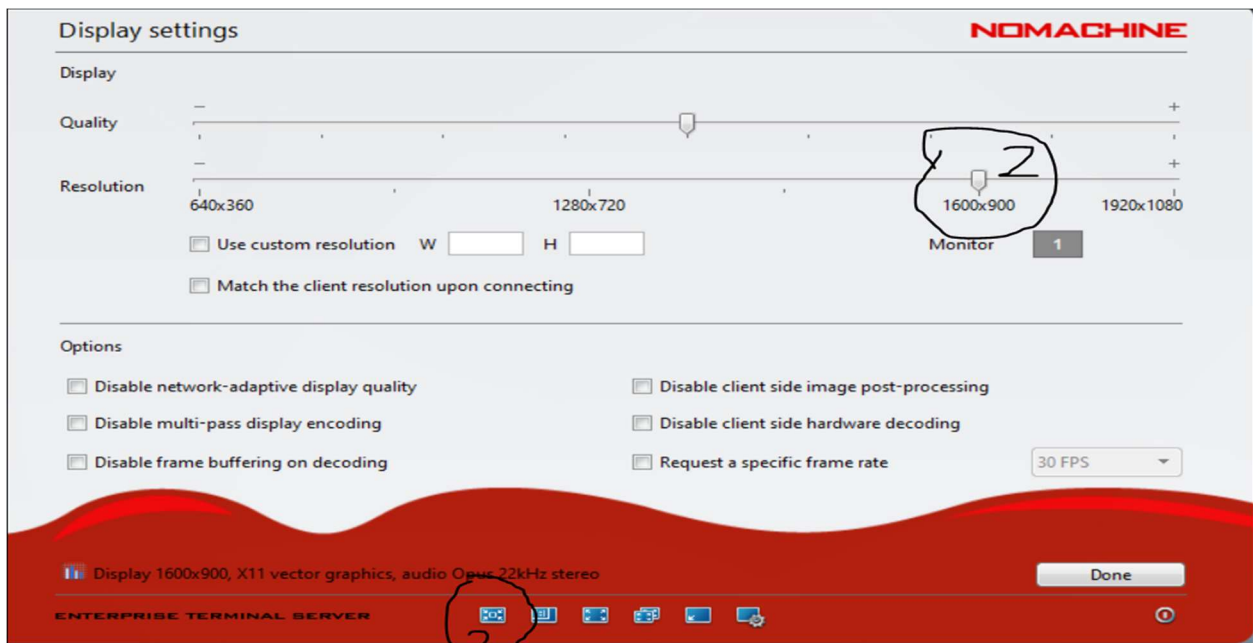
Step 1. Install NoMachine which will be used to access nx.cs.uwindsor.ca; see the link:

<http://www.help.cs.uwindsor.ca/en/servers/remote-access/no-machine/>

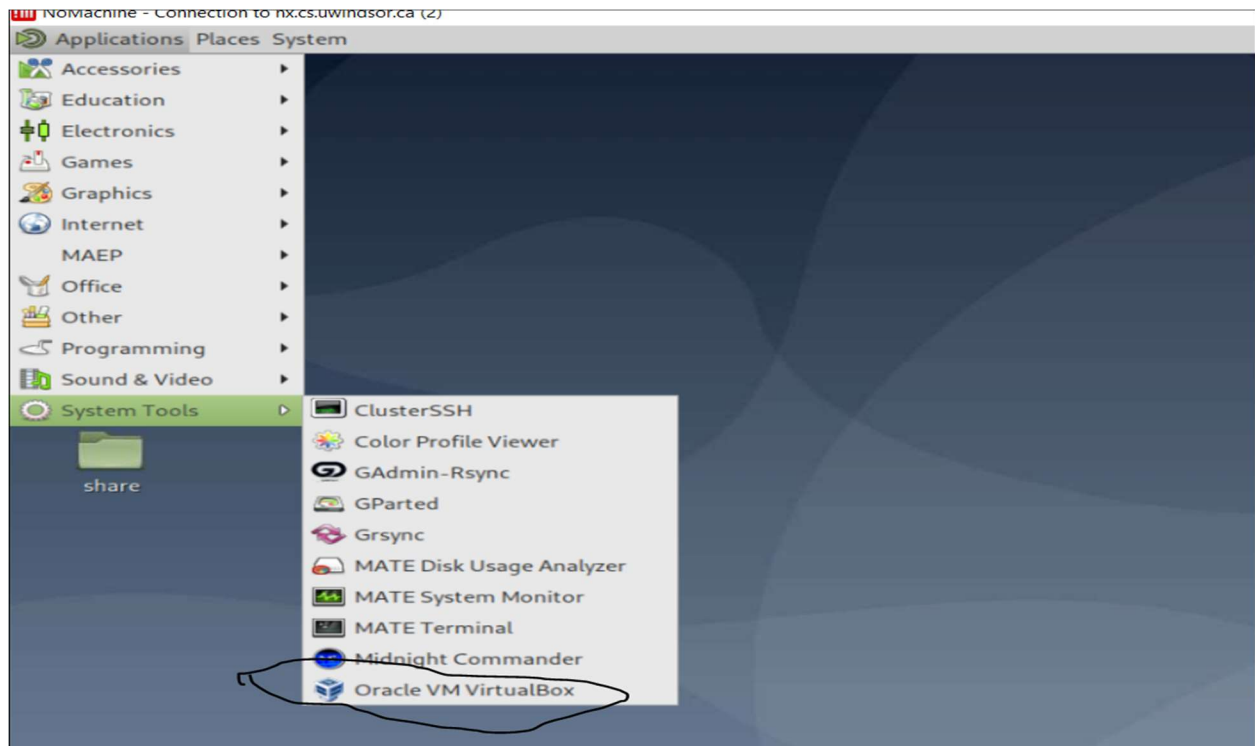
Step 2. Log in to nx.cs.uwindsor.ca using NoMachine. Change the setting when you see the following by clicking on button 1:



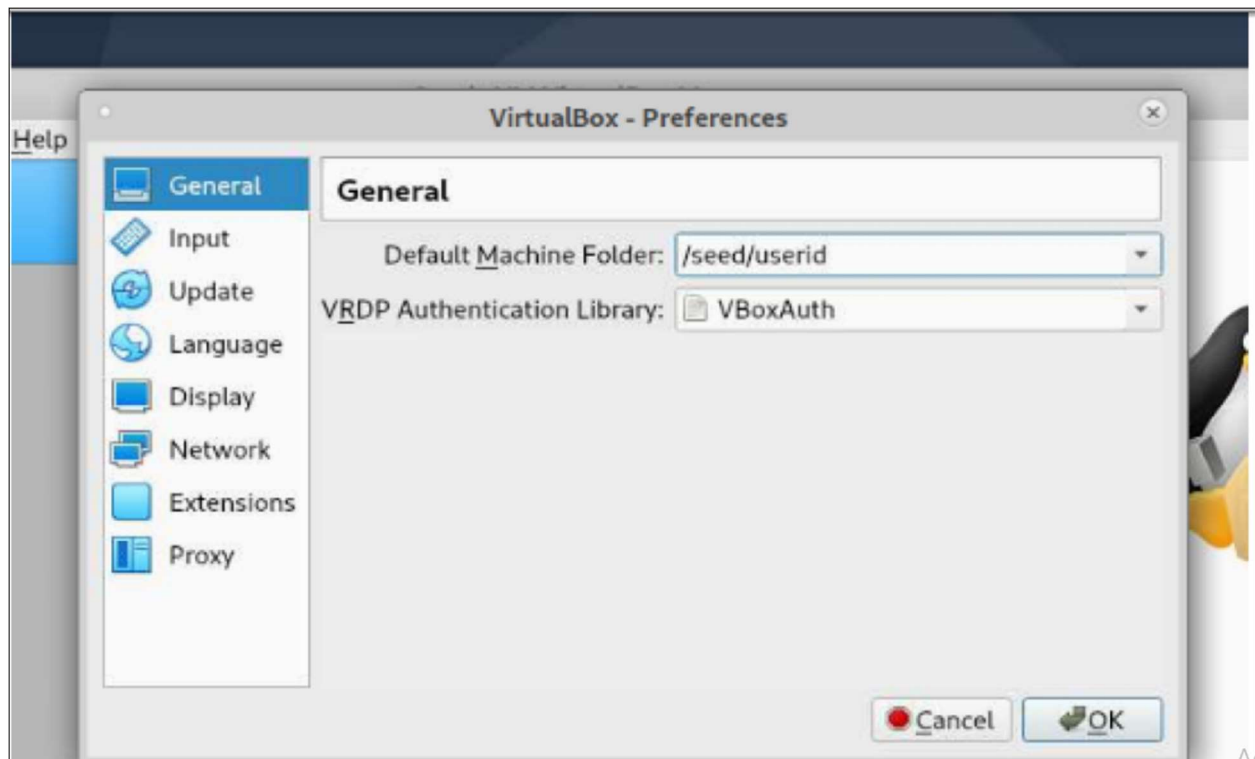
Then, set the resolution at 2 and click button 3.



Step 3. Find VirtualBox:

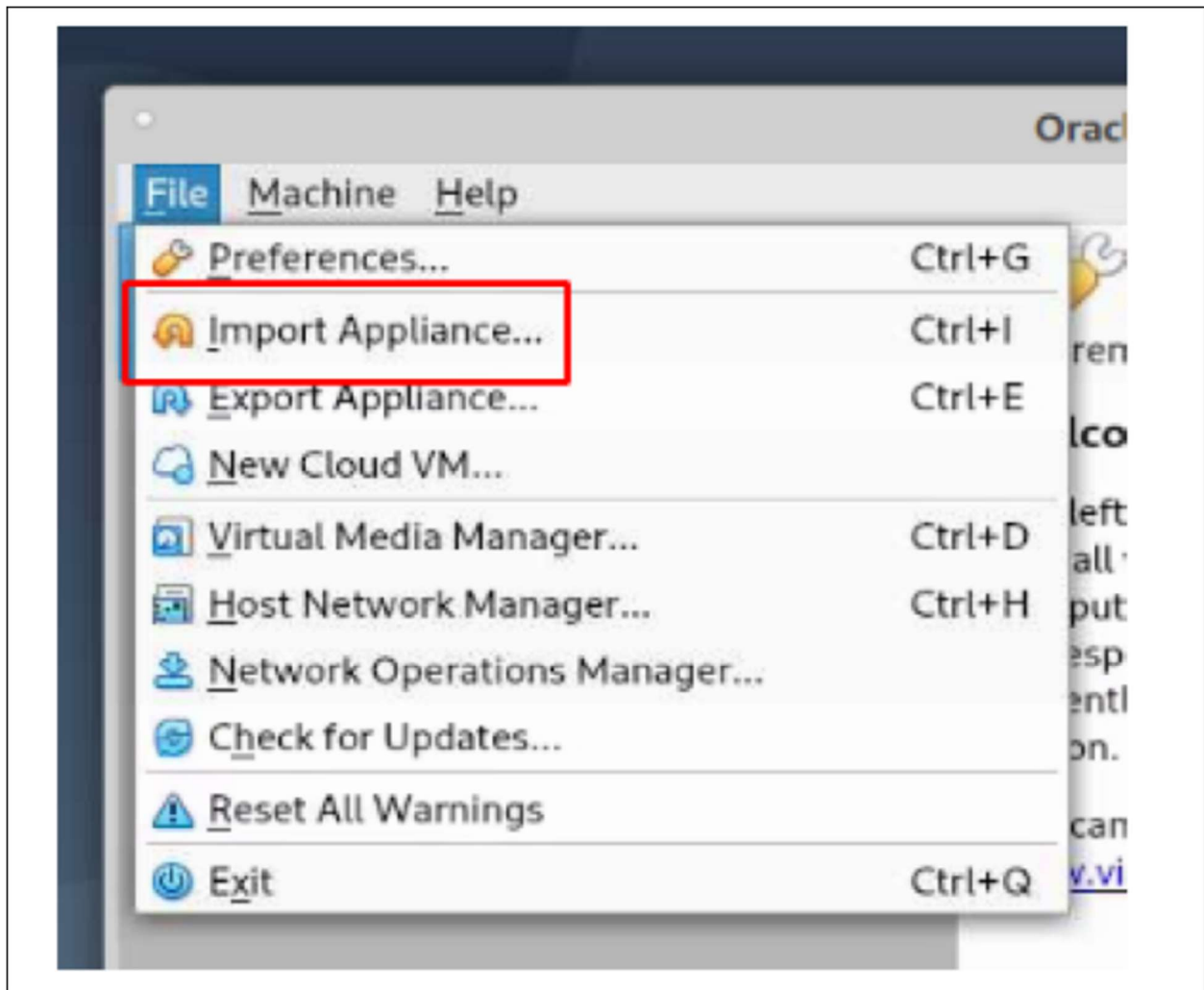


Step 4. Set your default machine folder: File→Preference-> General:

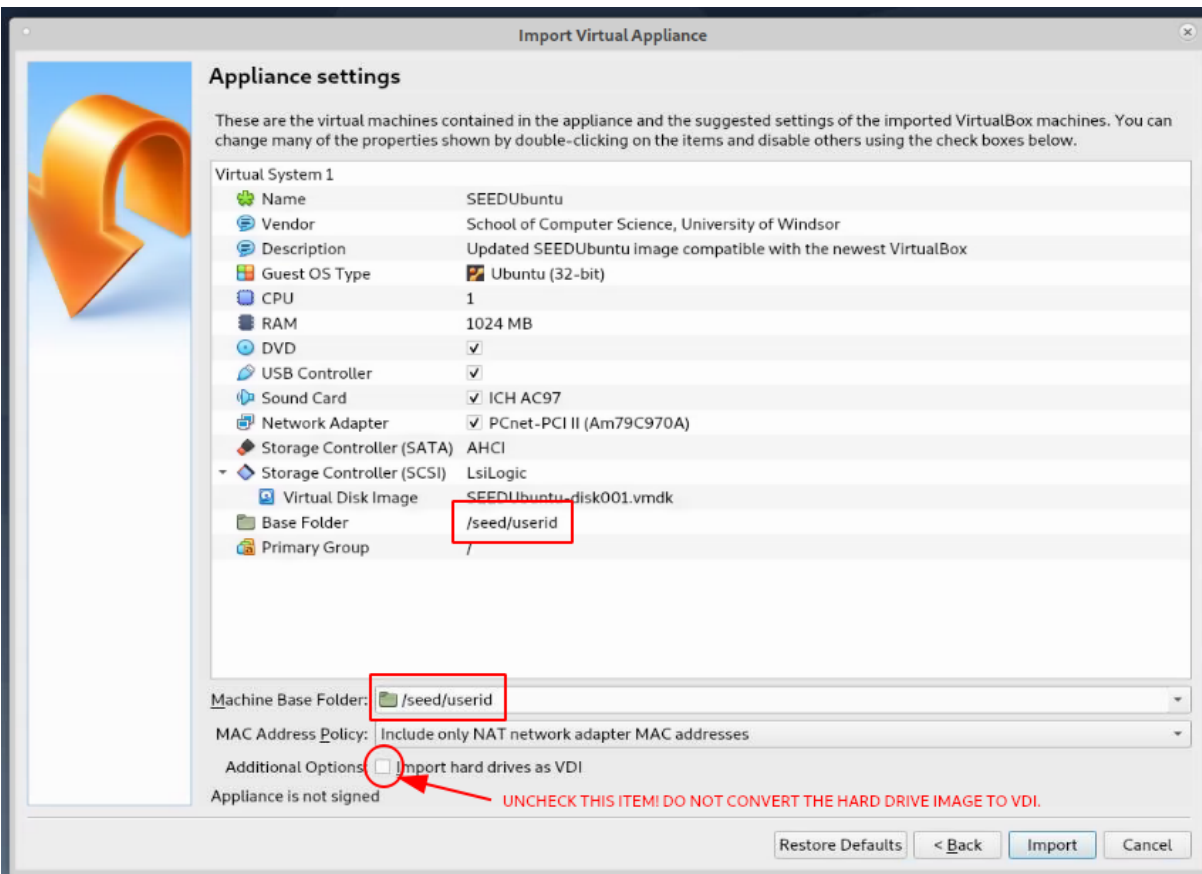
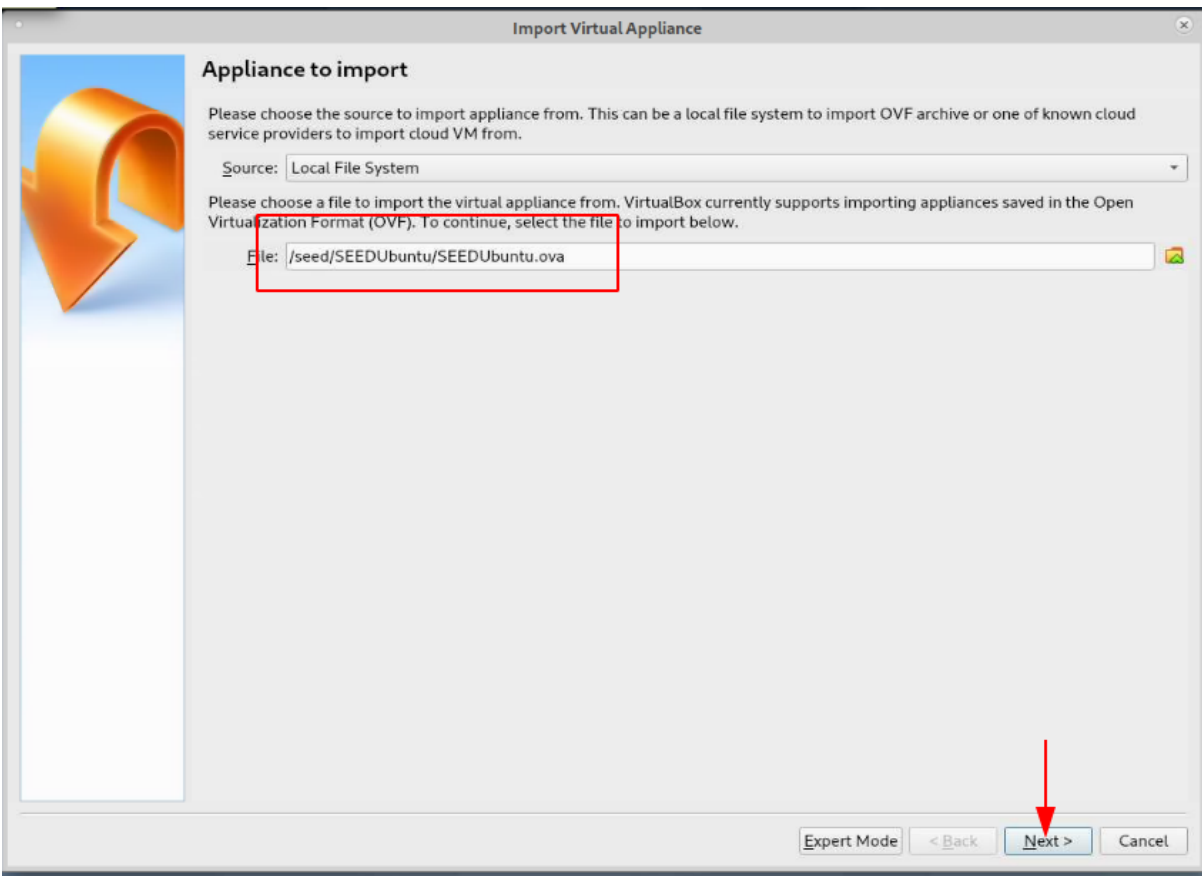


Here userid is your uwindsor user id (e.g., my id is jiangshq) and input your userid.

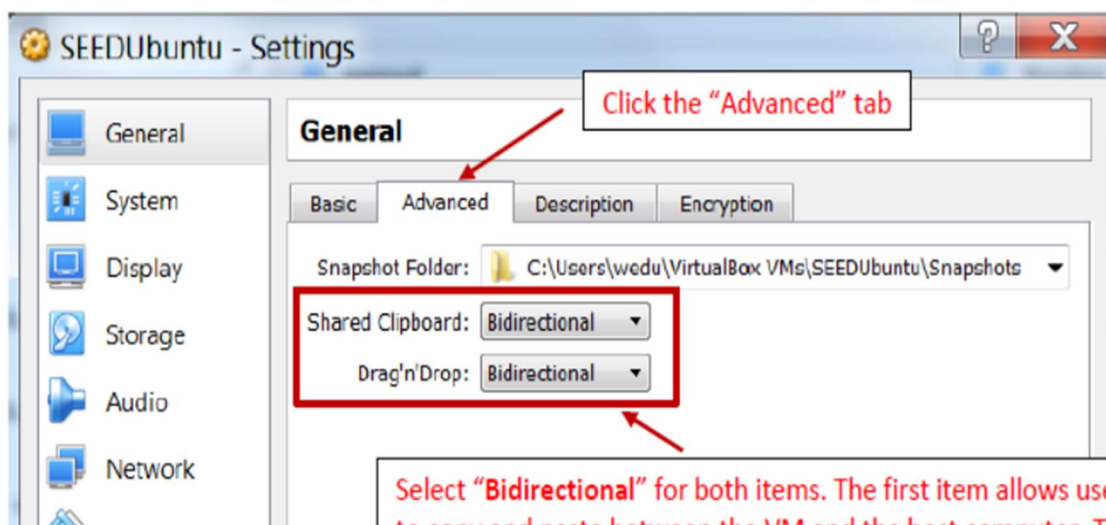
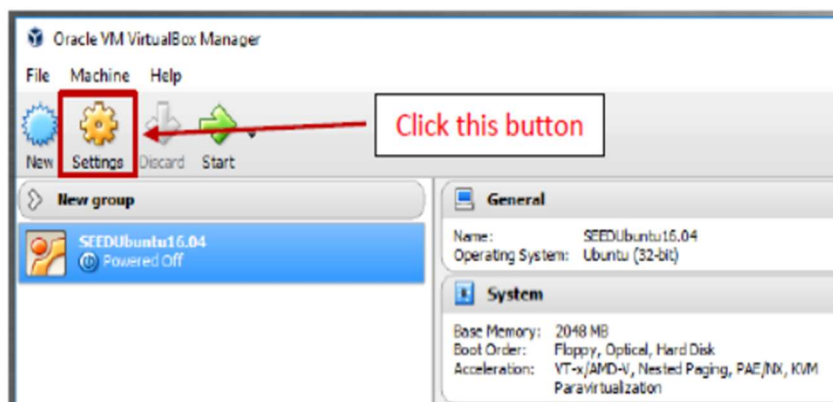
Step 4. Import Virtual Machine: first,



Second, see the next page.

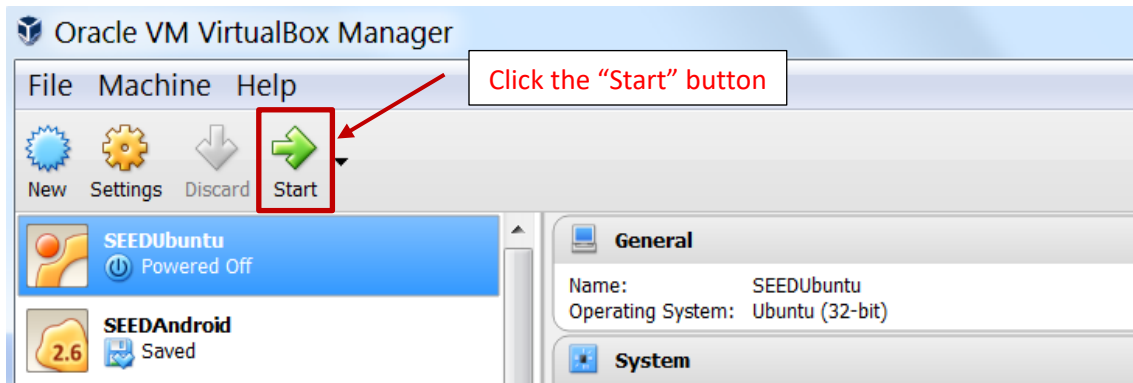


Step 5: Configure the VM



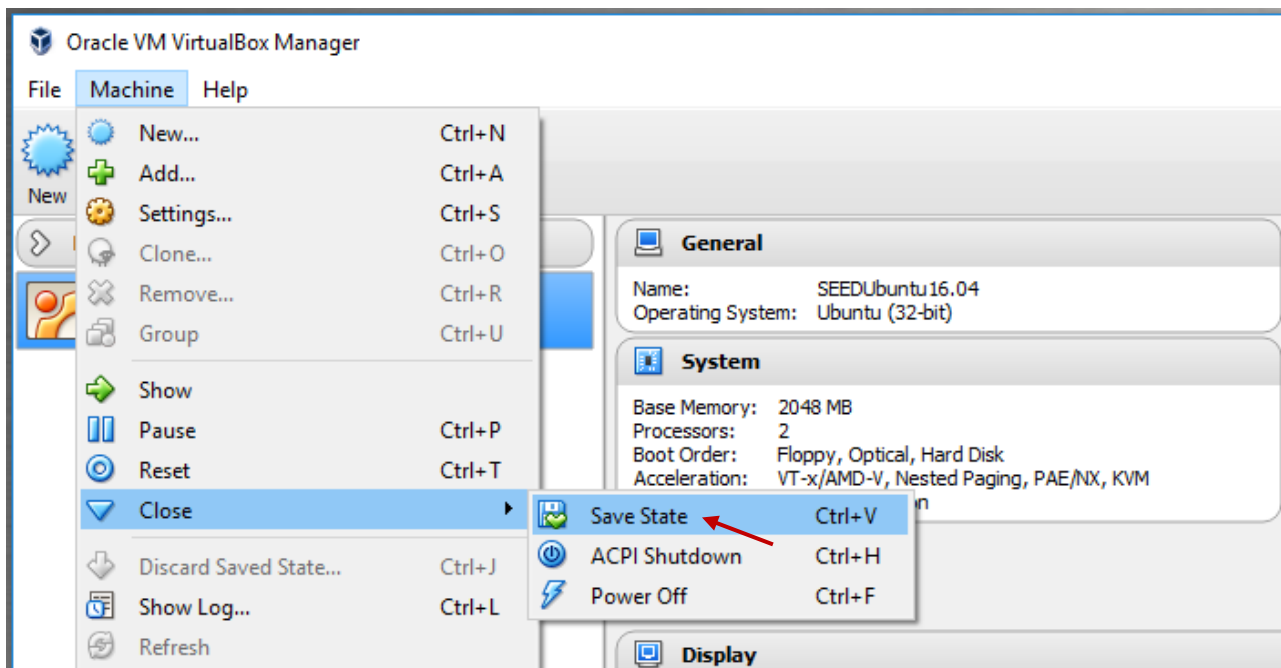
Select **"Bidirectional"** for both items. The first item allows users to copy and paste between the VM and the host computer. The second item allows users to transfer files between the VM and the host computer using Drag'n Drop.

Step 6: Start the VM



Step 7: Stop the VM or Save the VM's State

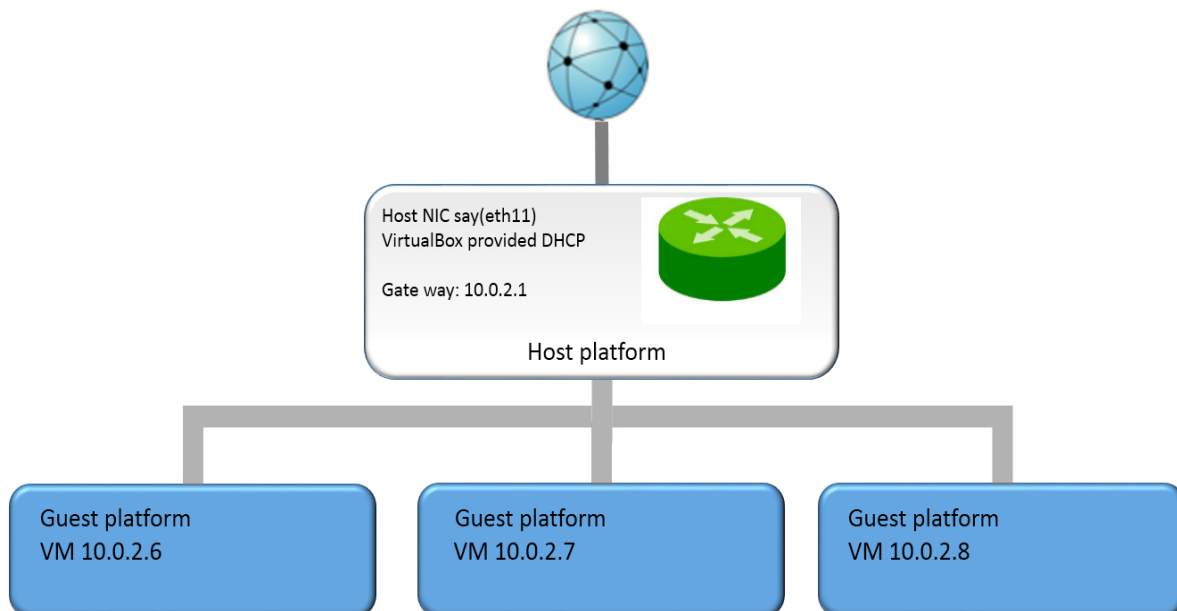
When you are done with your VM, you can always shut it down (from inside Ubuntu). A better alternative is to “freeze” the computer, so everything is saved. When you need it again, you can “unfreeze” it, and resume from where you left off. This is much faster and convenient than shutting down and rebooting the VM. To achieve this, you can use the “Save State” option.



Appendix B: Network Configuration in VirtualBox for SEED Labs

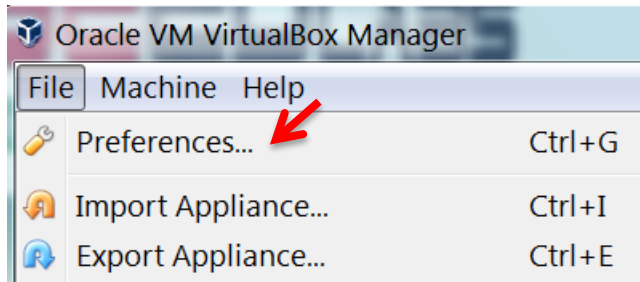
In many of the SEED labs, we need to run multiple guest VMs, and these VMs should be able to (1) reach out to the Internet, (2) communicate with each other. In Virtualbox, if we use the “NAT” setting (default setting) for each VM, we can achieve 1, but not 2, because each VM will be placed in its own private network, not on a common one; they even have the same IP address, which is not a problem because each VM is the only computer on its own private network. On the other hand, if we use the “Host-only” setting for each VM, we can achieve 2, but not 1. Using this setting, all the VMs and the host will be put on a common network, so they can communicate with each other; however, due to the lack of NAT, the VMs cannot reach out to the outside.

Therefore, in order to achieve all these 2 goals, we have to use a network adapter called “NAT Network”. The adapter works in a similar way to “local area network” or LAN. It enable VMs communication within same local network as well as the communication to the internet. All the communication goes through this single adapter. As show in Figure 1, gateway router transfers the packets among the VMs and transfers the packets from local network to Internet.

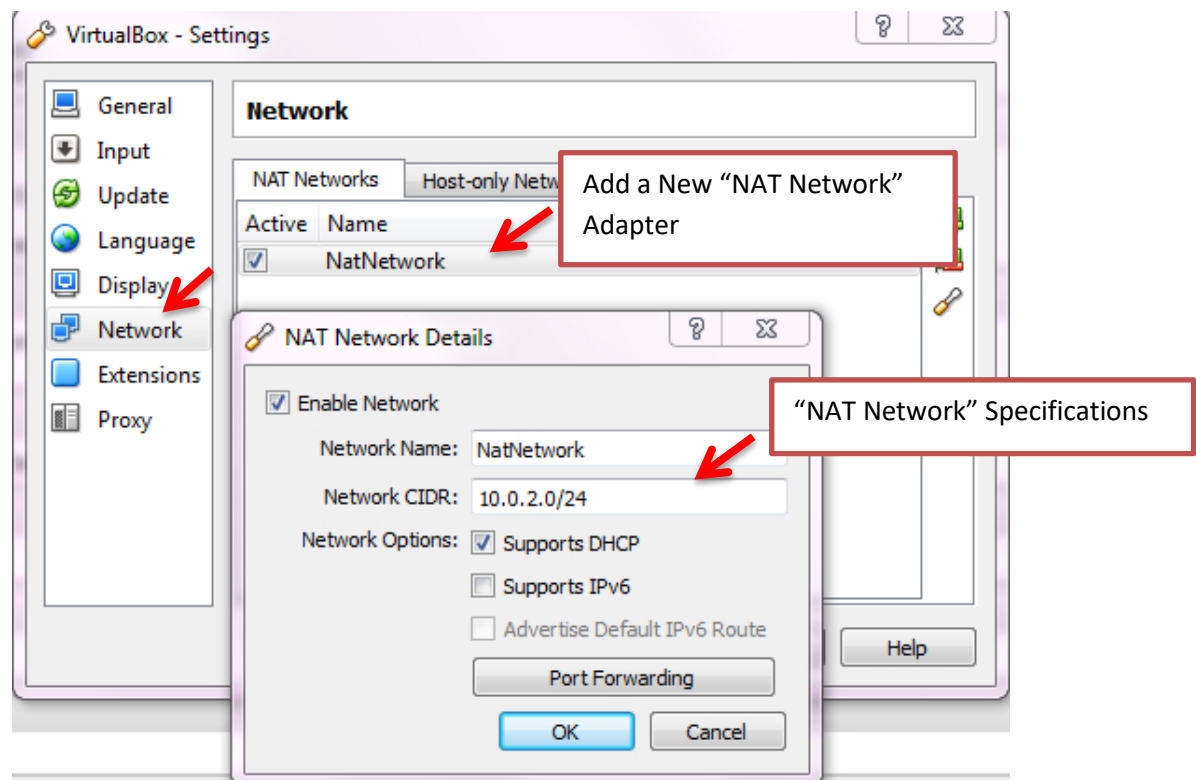


Configuration Instruction

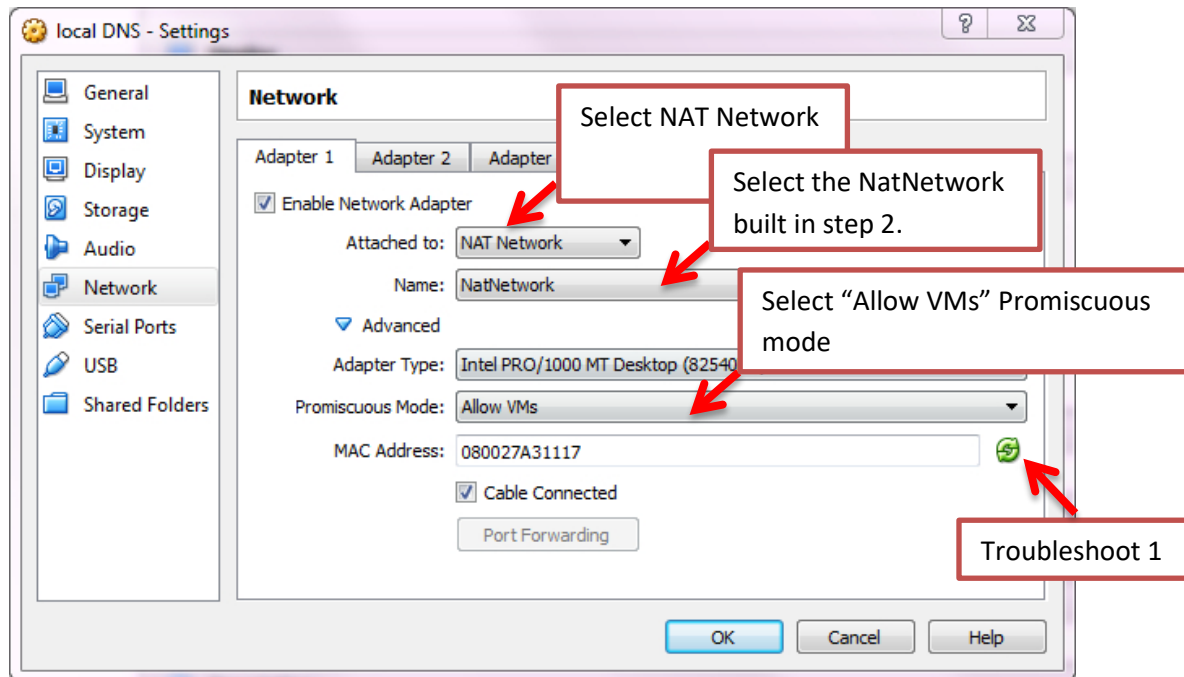
Step 1: Make sure you are using the most up-to-date VirtualBox. As show in the following figure, click the “File” on the top left of the VirtualBox main UI. Then choose “Preferences...” option.



Step 2: Click the “Network” tab on left panel. click the “+” button to create a new NAT Networks (NatNetwork) adaptor (if one does not exist). Double click on the NatNetwork, and look at its specifications. Set the specifications as the same as what is shown below.



Step 4: Go to VM setting, you need to power off the VM before making the following changes. Enable Adapter 1(at the same time, disable the other adapters), and choose “NAT Network”.



Step 5: Now power on the VM, and check the IP address.

```

seed@VM:~$ ifconfig
enp0s3  Link encap:Ethernet  HWaddr 08:00:27:f9:65:a2
        inet addr:10.0.2.34  Bcast:10.0.2.255  Mask:255.255.255.0
        inet6 addr: fe80::3b9:2676:84ea:969/64 Scope:Link
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:5 errors:0 dropped:0 overruns:0 frame:0
        TX packets:65 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:1070 (1.0 KB)  TX bytes:7216 (7.2 KB)

lo      Link encap:Local Loopback
        inet addr:127.0.0.1  Mask:255.0.0.0
        inet6 addr: ::1/128 Scope:Host
        UP LOOPBACK RUNNING  MTU:65536  Metric:1
        RX packets:68 errors:0 dropped:0 overruns:0 frame:0
        TX packets:68 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1
        RX bytes:21456 (21.4 KB)  TX bytes:21456 (21.4 KB)

seed@VM:~$

```

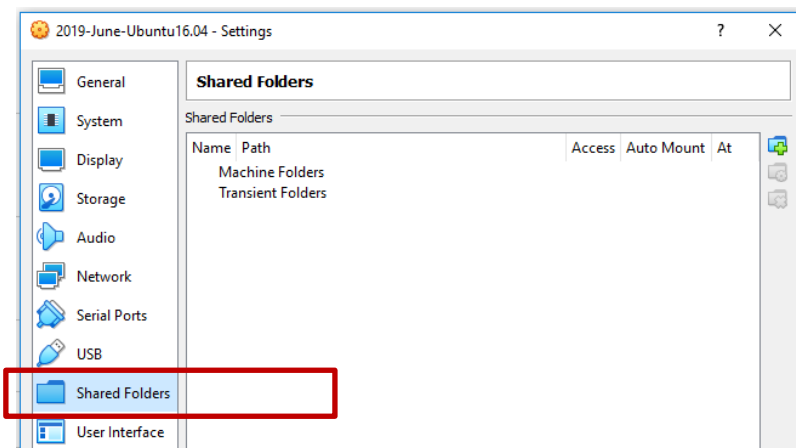
Troubleshooting:

- If VMs cannot ping each other, refresh the MAC Address can resolve the issue. The way to resolve the issue is shown in figure 4, troubleshoot 1.

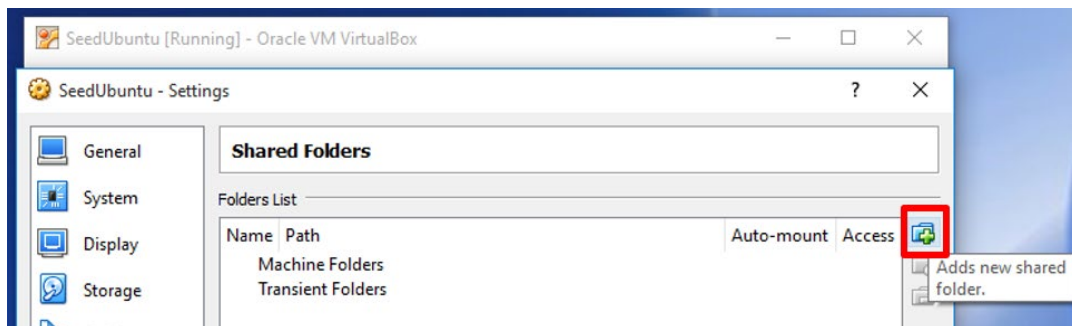
Appendix D: Folder Sharing

Files can be shared between the host computer and the guest operating system in VirtualBox. The following steps show how to do so.

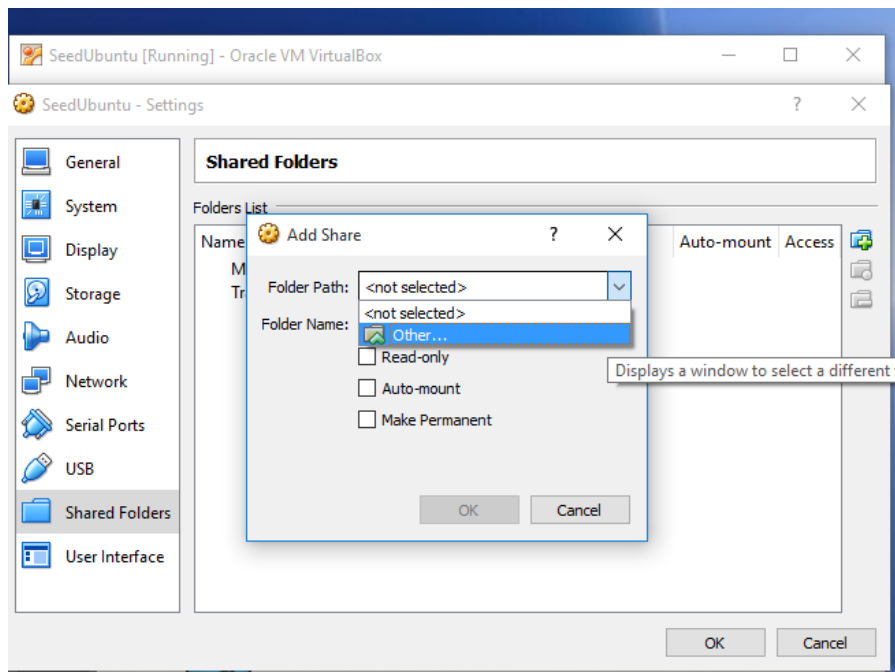
1. Create the folder to be shared on the host computer. In this tutorial we name the folder **share**.
2. Boot the Guest operating system in VirtualBox.
3. Go to the Settings popup window, and select “Shared Folders”



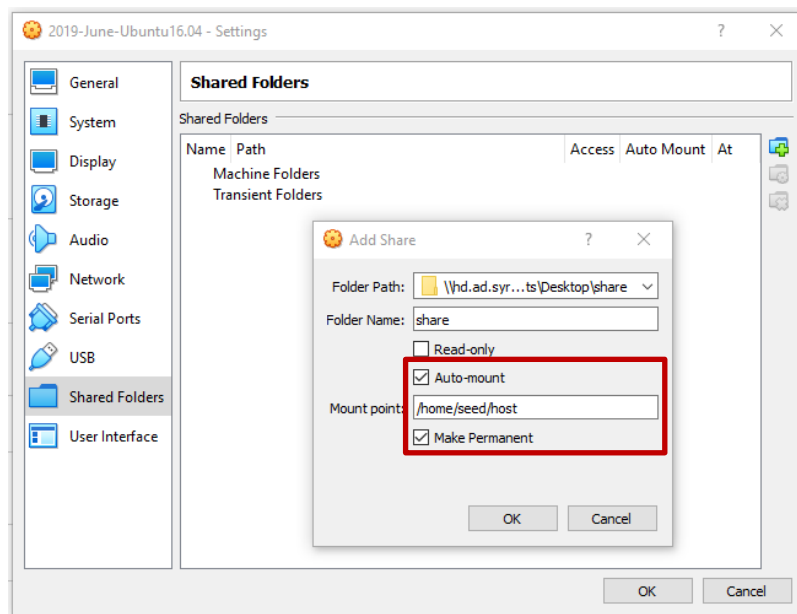
4. Choose the 'Add' button.



5. Choose “Other ...”, and select a folder from the popup window.

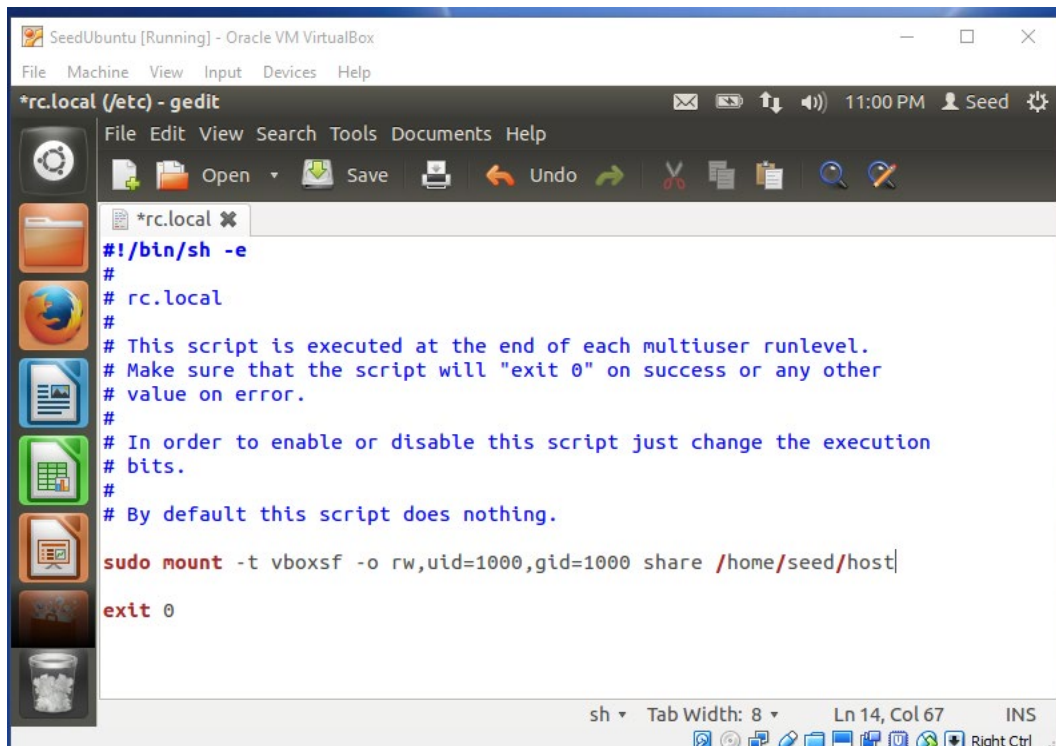


6. Select **Auto Mount** and **Make Permanent** option. Click OK. Click OK again to close the Settings Dialog.



7. Open a terminal in the VM. Make a directory and name it **host** (you can choose any name you like). Use command "**mkdir /home/seed/host**"
8. We want files in our mount point (~/.host) to be owned by the current user. Also we want the mounted shared folder to persist after reboot. Hence, we will edit the `/etc/rc.local` file (using "**sudo gedit /etc/rc.local**") and add the command below (1000 is the User ID and group ID of the user `seed`):

```
sudo mount -t vboxsf -o rw,uid=1000,gid=1000 share /home/seed/host
```

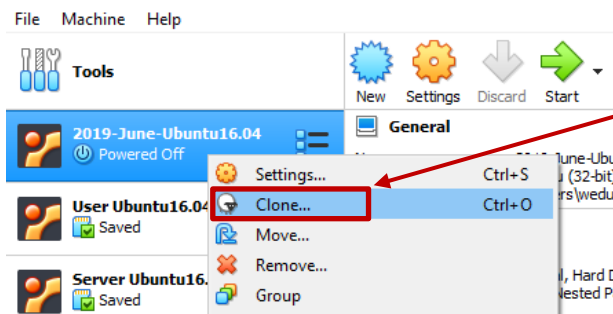


9. Save the changes and reboot VM. Now anything placed in `/home/seed/host` inside the VM should be visible from the `share` folder on the host machine, and vice versa.

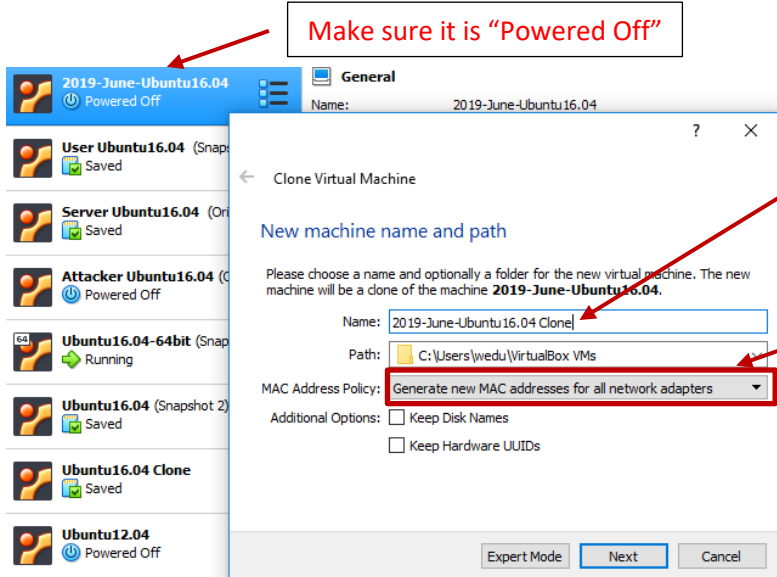
Appendix A: Use “Clone” to create Multiple VMs

Some SEED labs require multiple VMs. The easiest way to create multiple VMs is to create one first, and then use the “Clone” mechanism to clone it. Before doing the cloning, please ensure the following:

- **IMPORTANT:** make sure that the VM is **fully shutdown** (not in a “Saved” state), or there will be all sorts of problems.
- Configure network (see Appendix B); otherwise you have to do it for each VM.
- Configure folder sharing (see Appendix D); otherwise you have to do it for each VM.



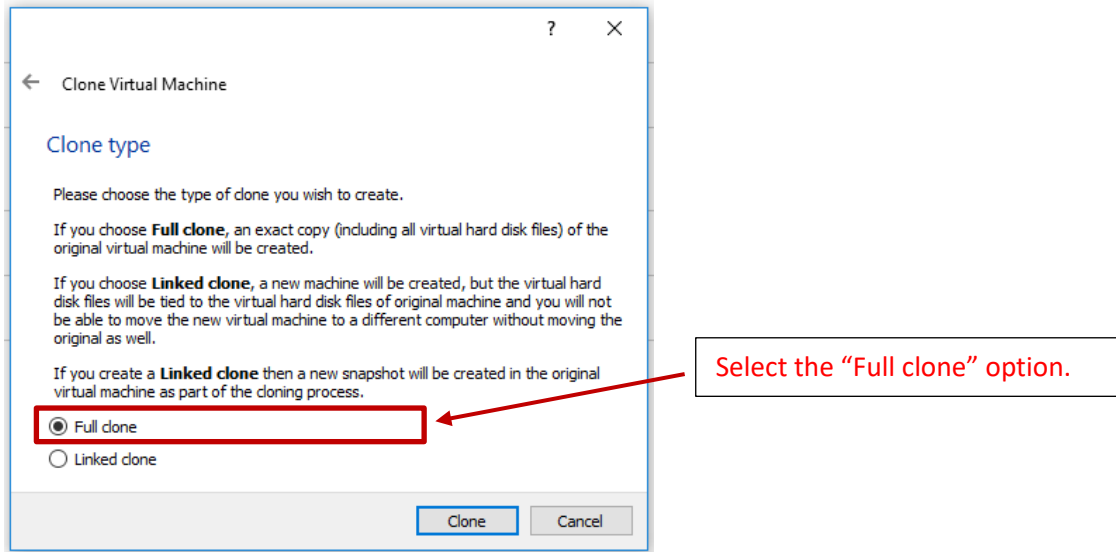
Right click the VM, and select “Clone...”



Make sure it is “Powered Off”

You can pick a meaningful name here.

IMPORTANT: we don’t want to clone the MAC address. Select this option. If you don’t, both VMs will have the same MAC address, and therefore, they will get the same IP address. That will cause problems.



The clone will take a few minutes, depending on the speed of your computer.