5/1/2018

Run SEED VM on VirtualBox

User Manual

How to Use Virtualbox to Run SEED Ubuntu VM

Appendix A: Creating Multiple VMs

Appendix B: Network Configuration

Appendix C: Taking Snapshots of VM

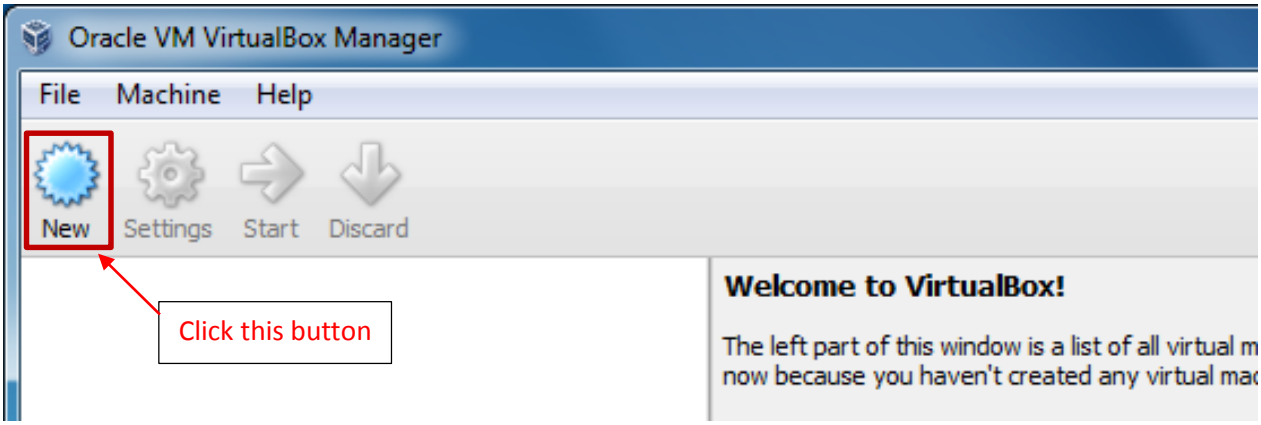
Appendix D: Creating Shared Folder

Wenliang Du
SYRACUSE UNIVERSITY

How to use VirtualBox to Run SEED Ubuntu VM?

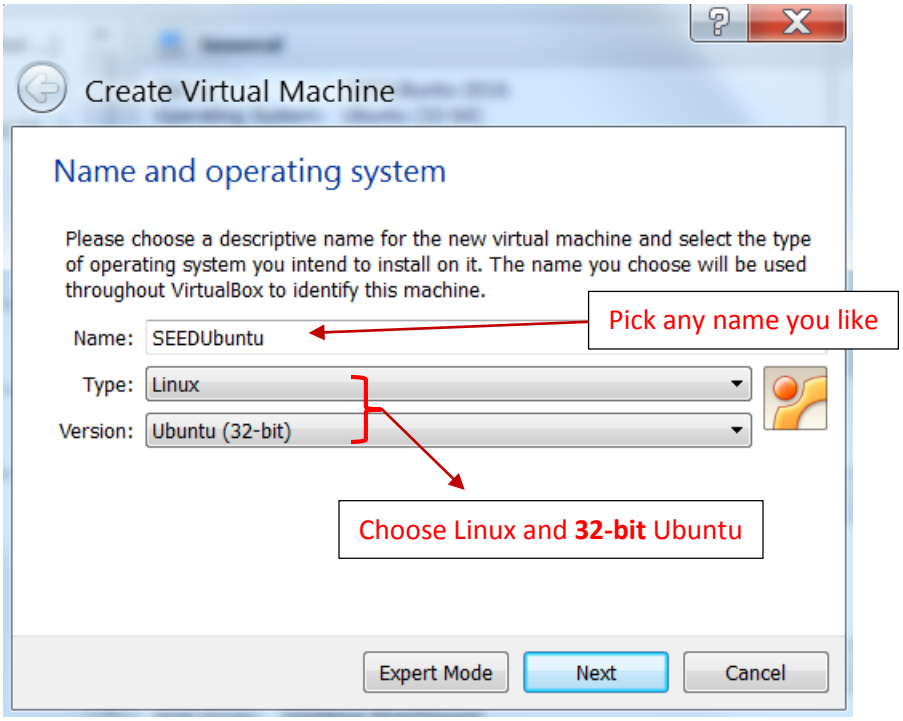
The following instructions are based on VirtualBox 5.0.16. They are similar for newer versions of VirtualBox.

Step 1: Create a New VM in VirtualBox

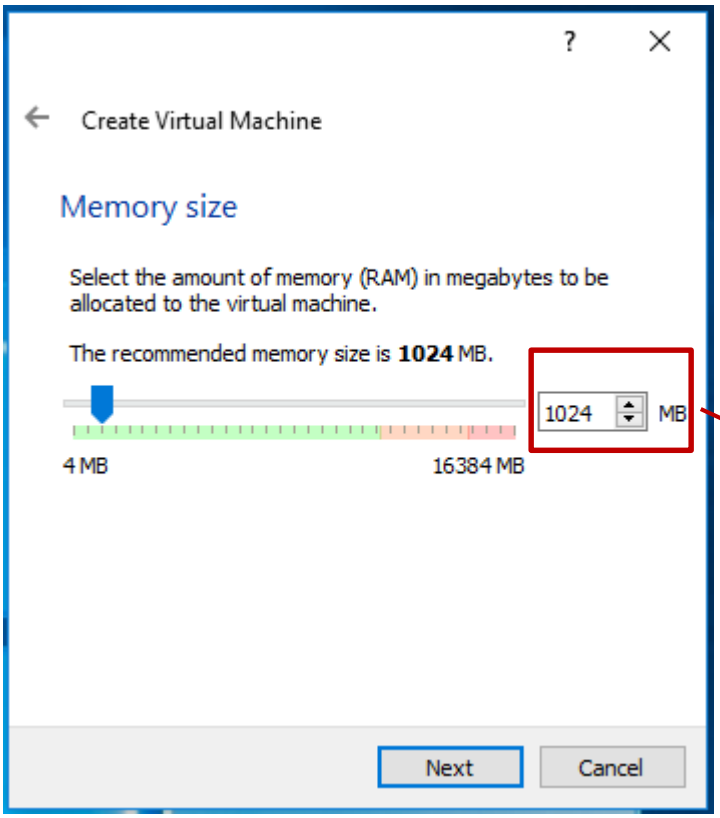


Step 2: Provide a Name and Select the OS Type and Version

Do NOT pick Ubuntu (64-bit), even though your machine is 64 bit. Our prebuilt VM is 32-bit Ubuntu.

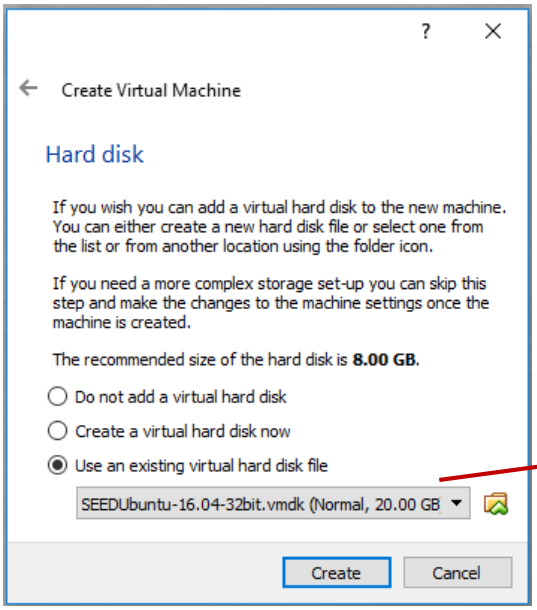


Step 3: Set the Memory Size



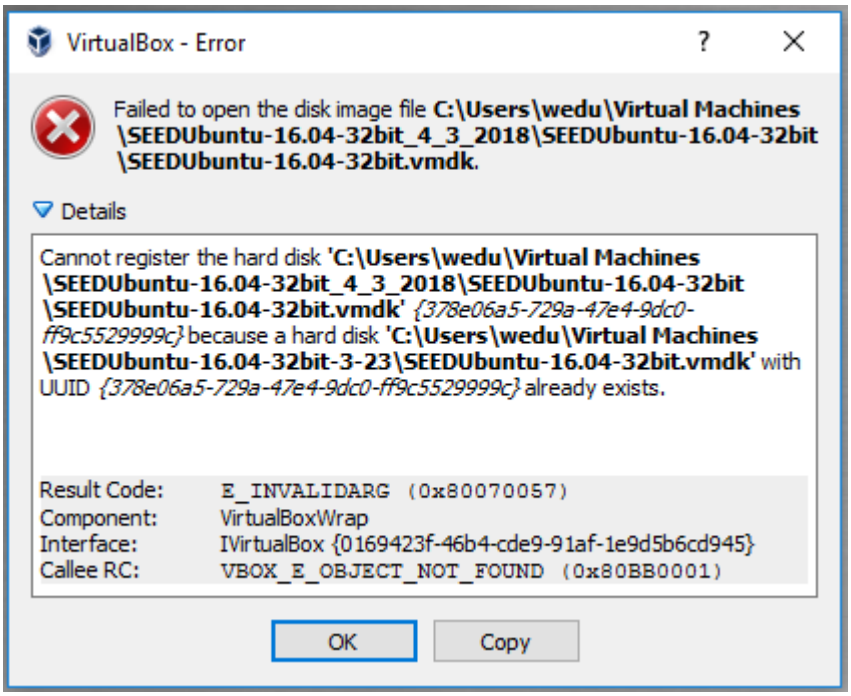
512 MB should be sufficient. If your computer has more RAM, you can increase accordingly. The more memory you give to the VM, the better the performance you will get.

Step 4: Select the Pre-built VM File Provided by Us



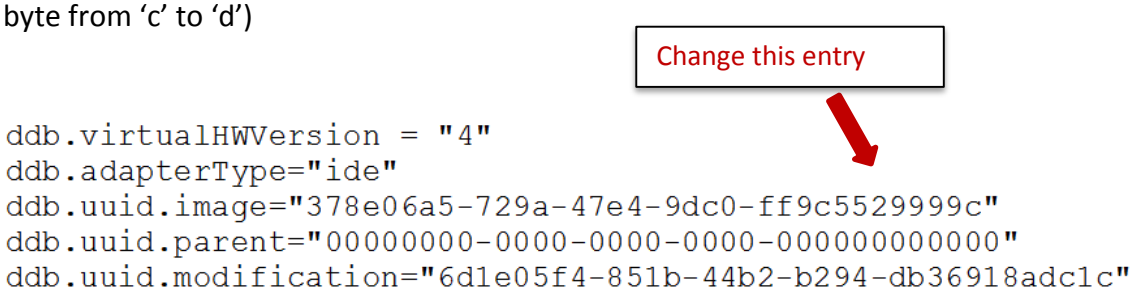
Pick this file in the unzipped folder: **SEEDUbuntu-16.04-32bit.vmdk.**

In the above step, you may encounter the following error; otherwise, directly go to Step 5.

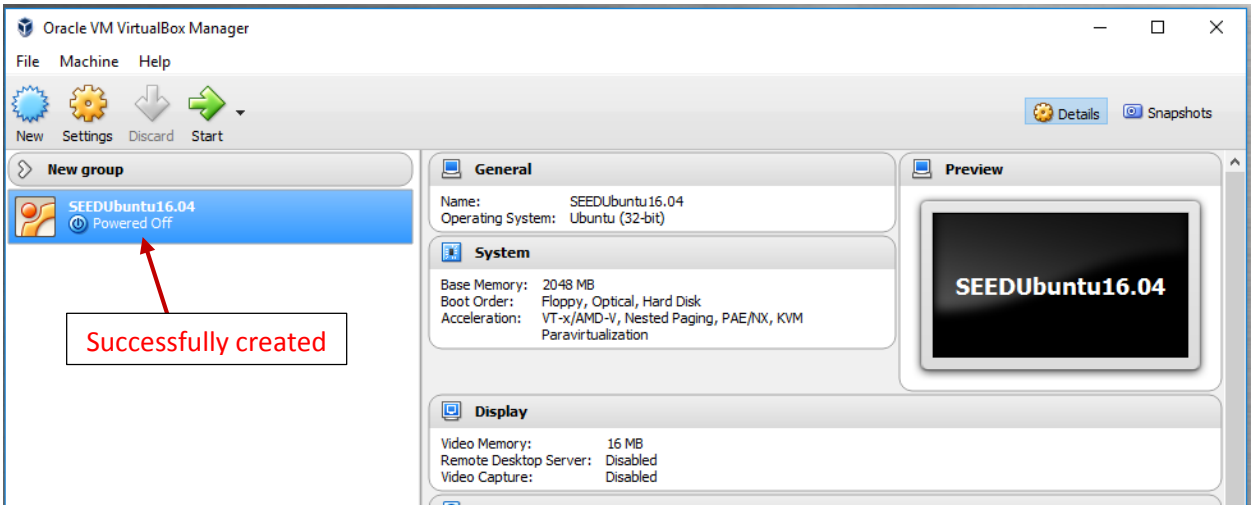


Reason and Solution: This is because you copied the VM files from another VM, which is already loaded into VirtualBox. These two VMs have the same UUID, which is not allowed by Virtualbox. Here are several solutions depending on your situations:

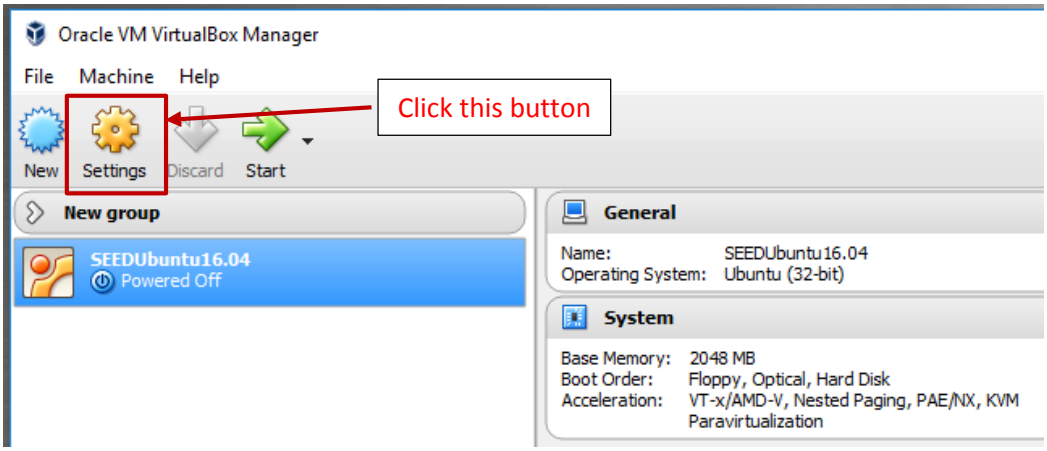
- If you plan to create multiple VMs using the same image, please use the clone mechanism (See Appendix A for details).
- If the older VM with the same UUID is no longer needed, remove it from VirtualBox will solve the problem.
- If you do want to keep the older VM, you can change the UUID of the new VM. The fastest way is to directly modify `SEEDUbuntu16.04.vmdk`, which is a text file. Search for the `ddb.uuid.image` entry, and change its value (e.g., change the last byte from 'c' to 'd')

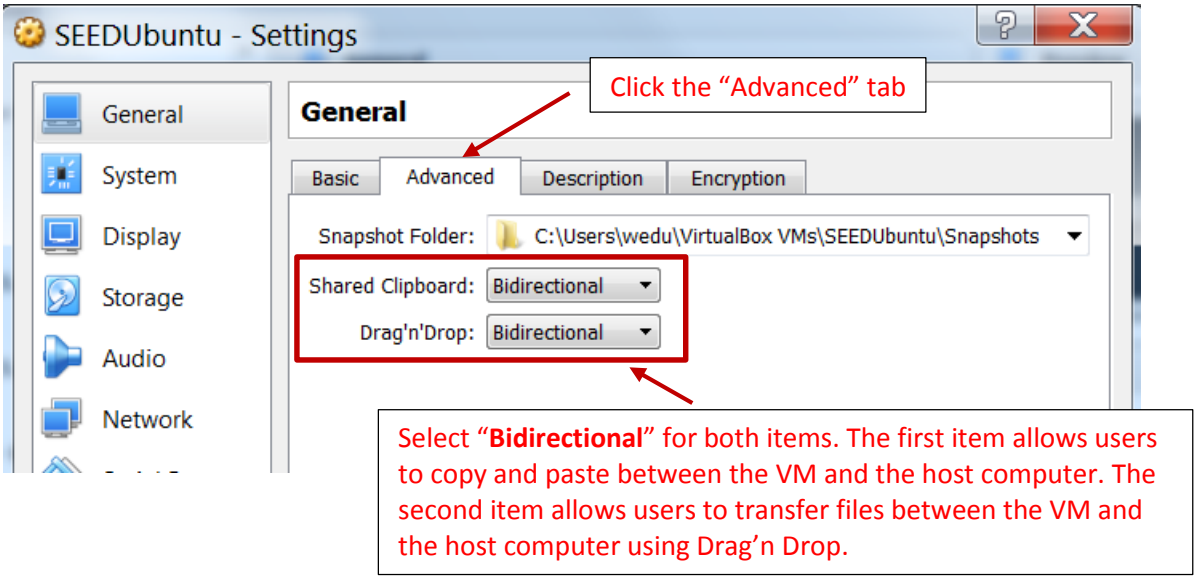


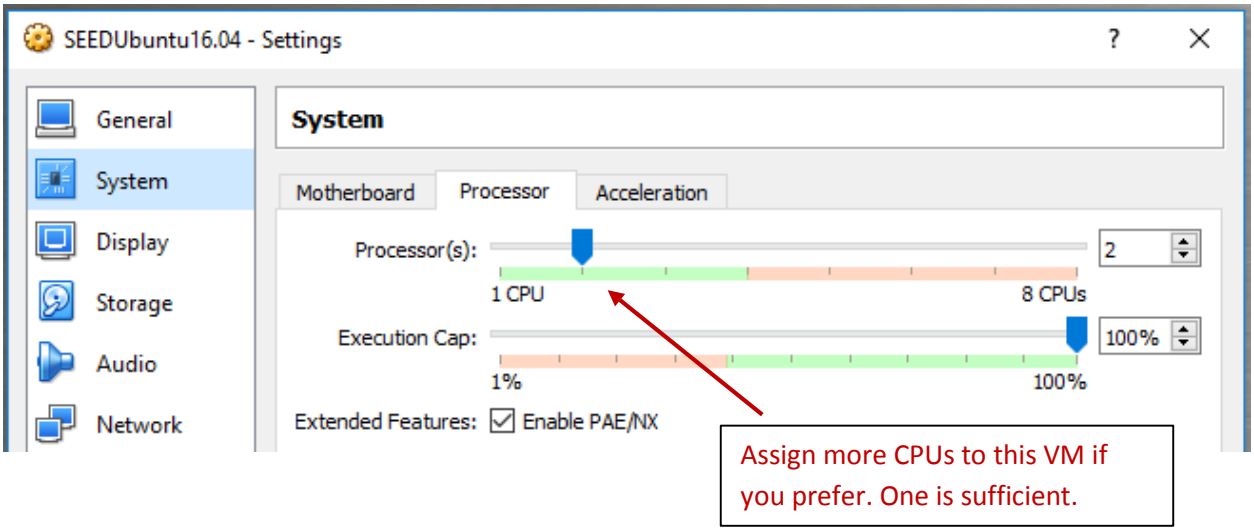
If there is no error (or after you fix the error), your VM will be created successfully.



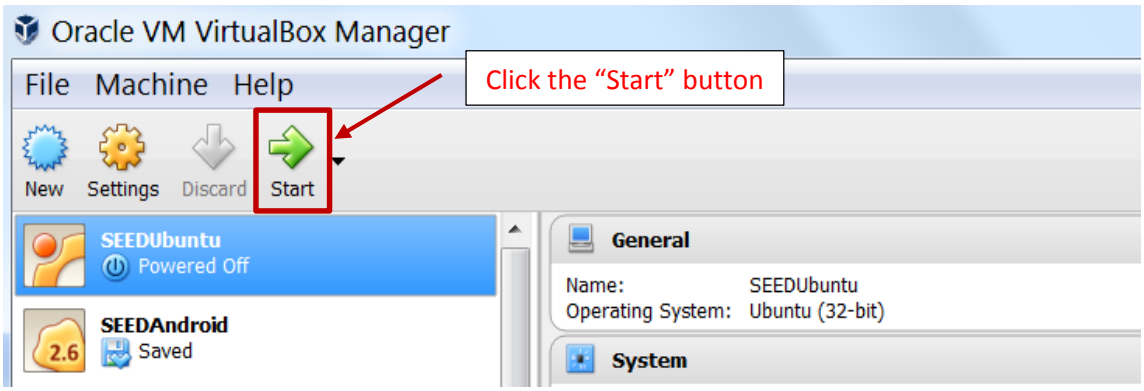
Step 5: Configure the VM





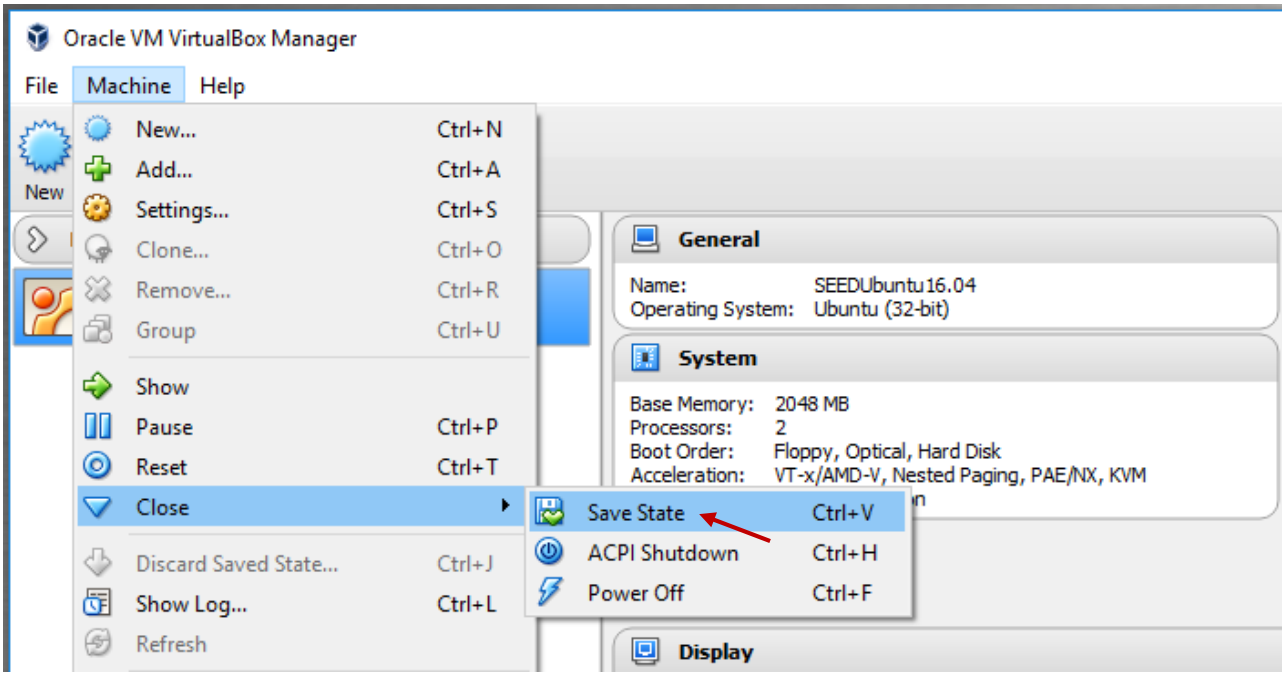


Step 6: Start the VM



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

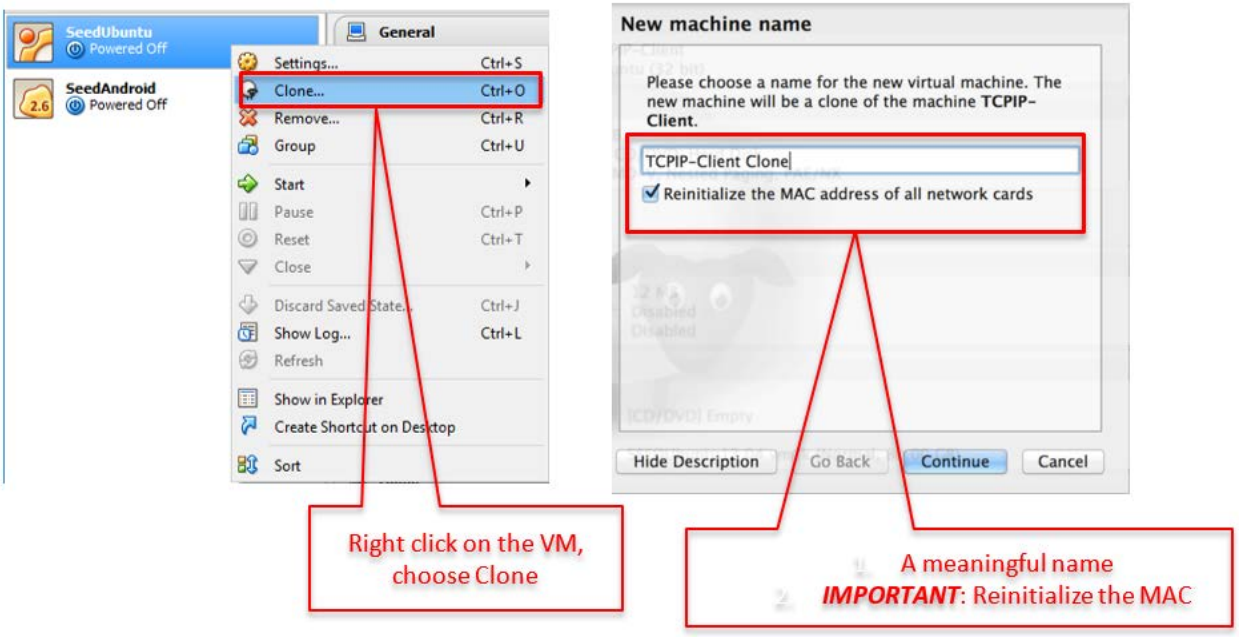
When you are done with your VM, you can always shut it down (from inside Ubuntu). A better alternative is to "freeze" the computer, so everything is saved. When you need it again, you can "unfreeze" it, and resume from where you left off. This is much faster and convenient than shutting down and rebooting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

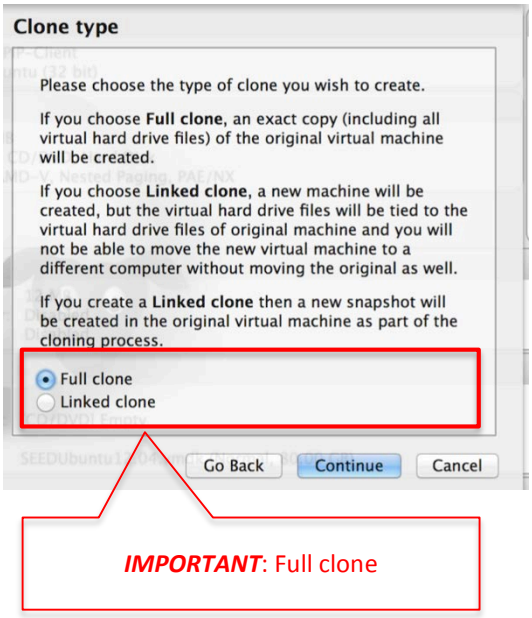

Oracle VM VirtualBox Manager
File
Machine
Help
New
New... Ctrl+N
Add... Ctrl+A
Settings... Ctrl+S
Clone... Ctrl+O
Remove... Ctrl+R
Group Ctrl+U
Show
Pause Ctrl+P
Reset Ctrl+T
Close
Discard Saved State... Ctrl+J
Show Log... Ctrl+L
Refresh
Save State Ctrl+V
ACPI Shutdown Ctrl+H
Power Off Ctrl+F
General
Name: SEEDUbuntu16.04
Operating System: Ubuntu (32-bit)
System
Base Memory: 2048 MB
Processors: 2
Boot Order: Floppy, Optical, Hard Disk
Acceleration: VT-x/AMD-V, Nested Paging, PAE/NX, KVM
Display

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.

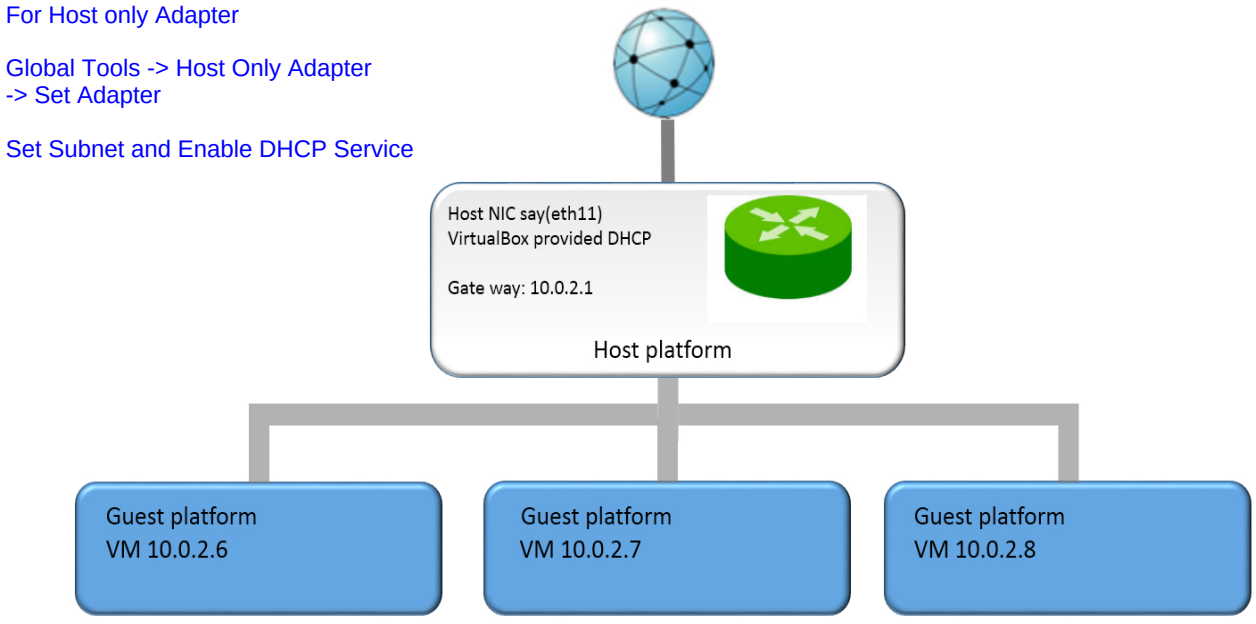




Appendix B: Network Configuration in VirtualBox for SEED Labs

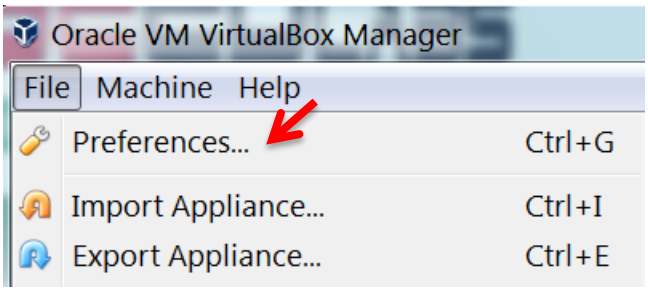
In many of the SEED labs, we need to run multiple guest VMs, and these VMs should be able to (1) reach out to the Internet, (2) communicate with each other. In Virtualbox, if we use the "NAT" setting (default setting) for each VM, we can achieve 1, but not 2, because each VM will be placed in its own 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.

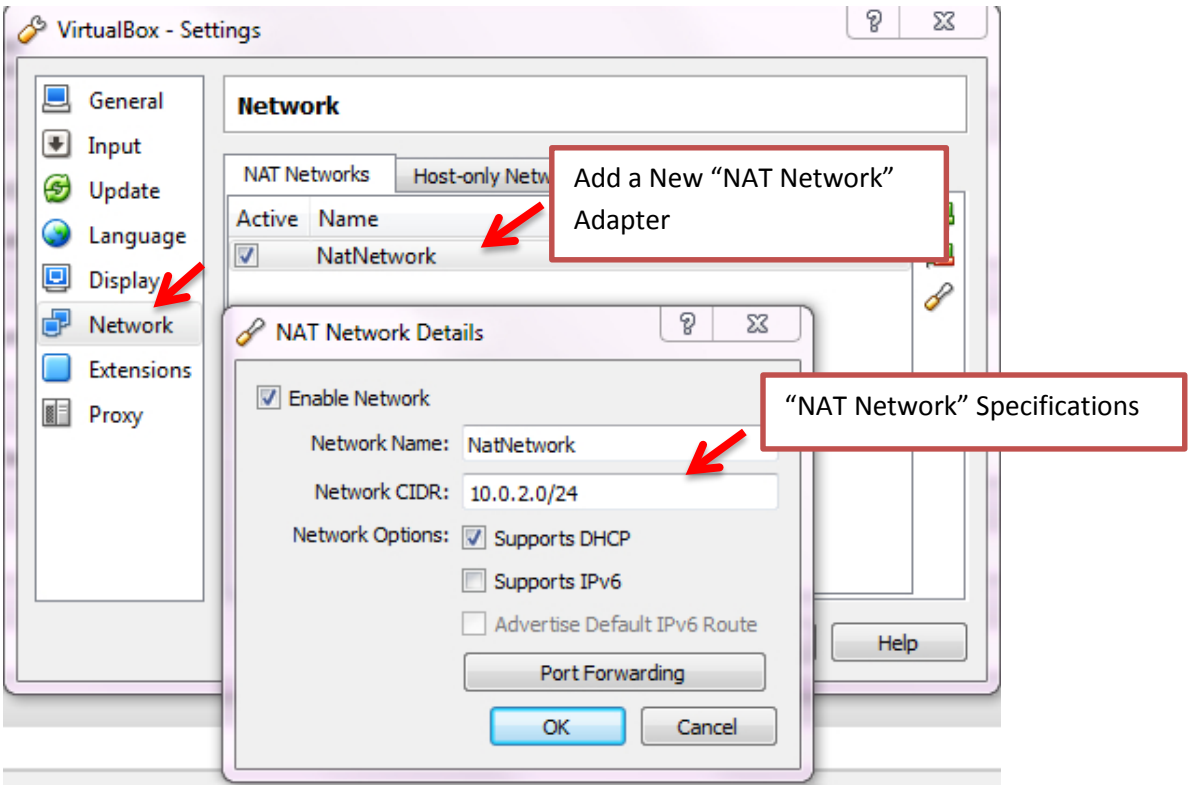


Configuration Instruction

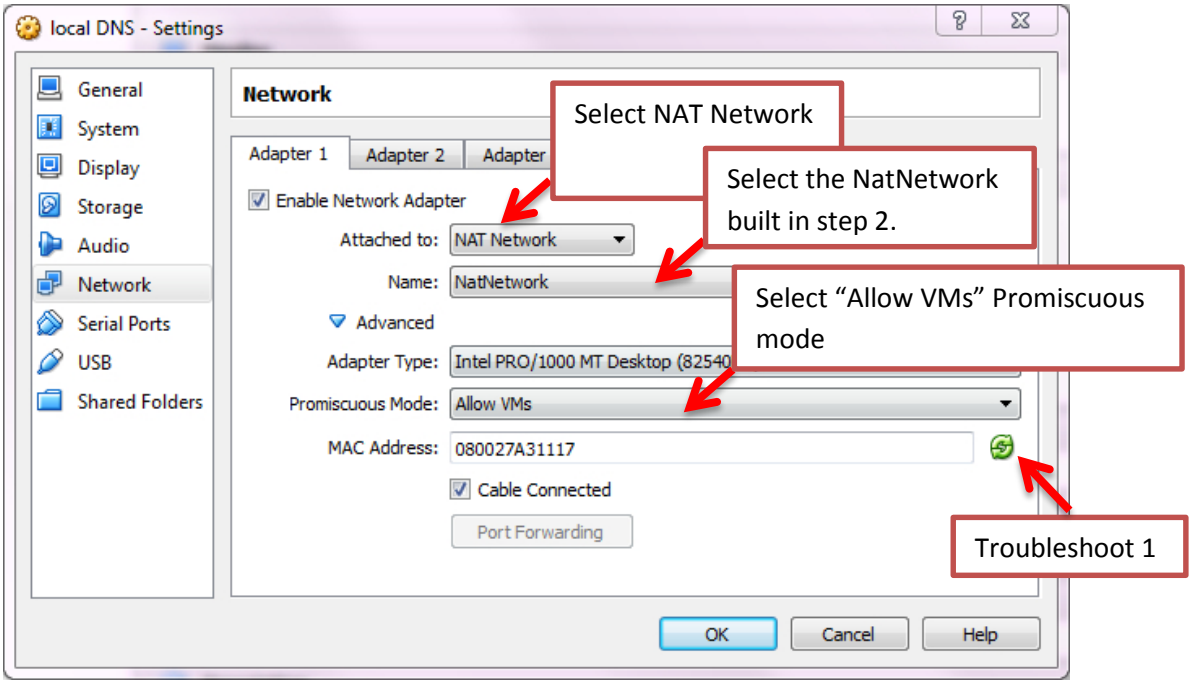
Step 1: Make sure you are using the most up-to-date VirtualBox. As show in the following figure, click the "File" on the top left of the VirtualBox main UI. Then choose "Preferences…" option.



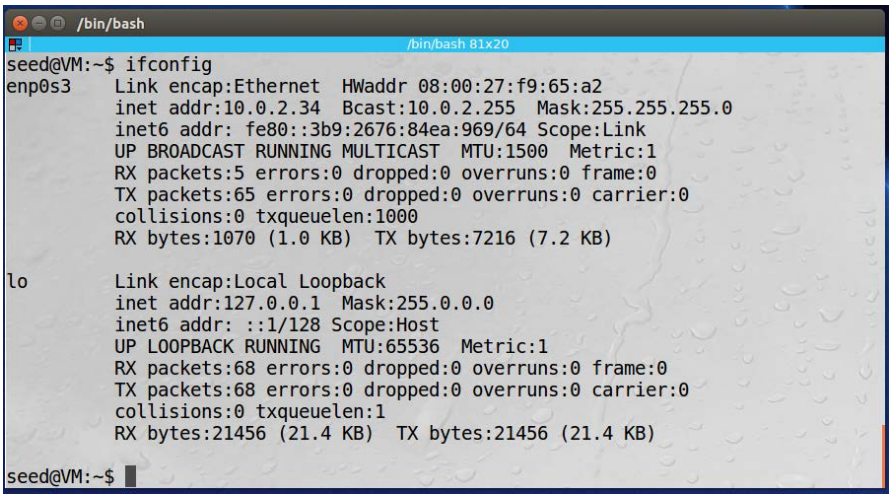
Step 2: Click the "Network" tab on left panel. click the "+" button to create a new NAT Networks (NatNetwork) adaptor (if one does not exist). Double click on the NatNetwork, and look at its specifications. Set the specifications as the same as what is shown below.



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.

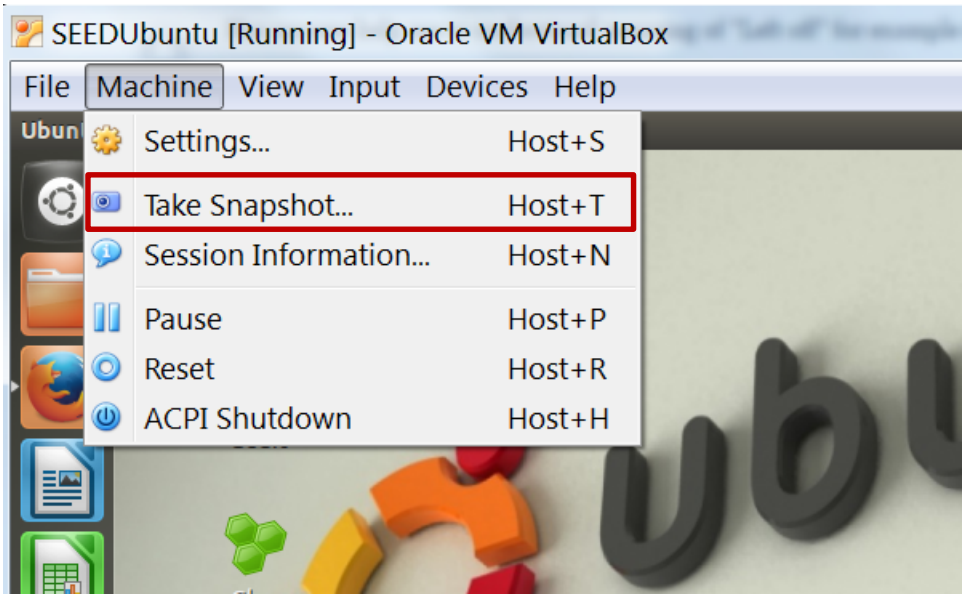


Troubleshooting:

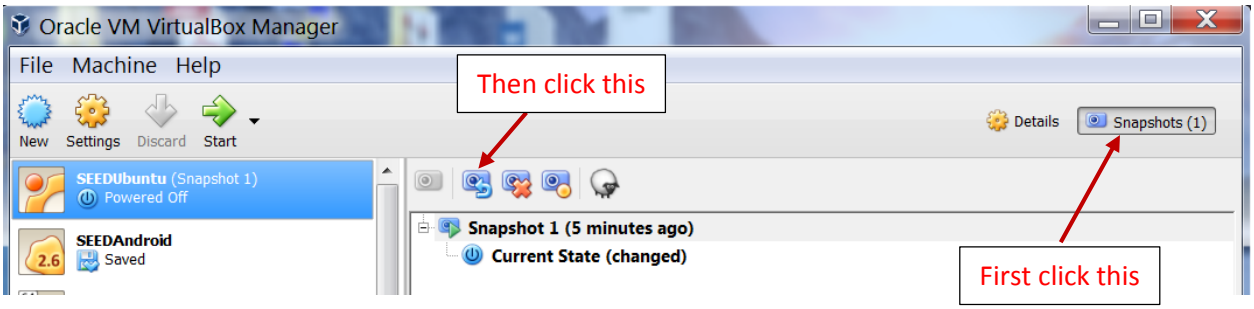
- If VMs cannot ping each other, refresh the MAC Address can resolve the issue. The way to resolve the issue is shown in figure 4, troubleshoot 1.

Appendix C: Take Snapshots and Recover from Snapshots

For some labs, you may need to make changes to the operating system. If you make a severe mistake, your VM may not be able to boot up again, and you will lost everything inside the failed VM. have done. To avoid such trouble, before doing anything dangerous to the OS, it is better to take a snapshot of your current VM. You can take as many snapshots as you want.



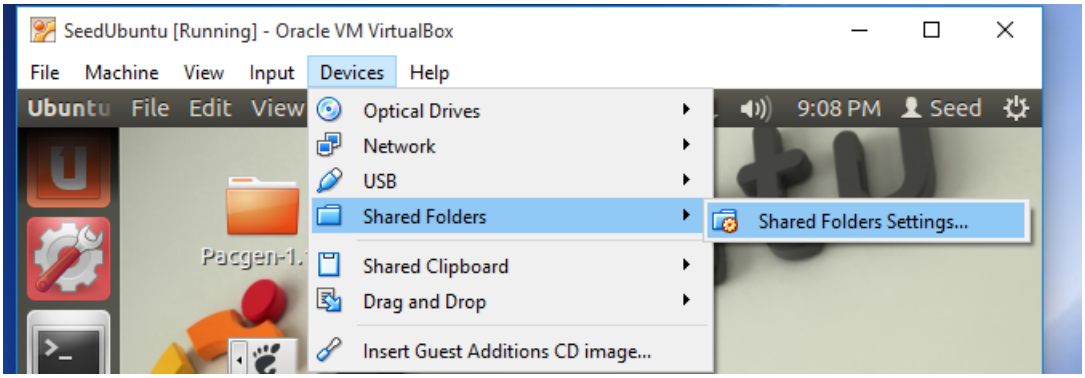
To restore from a snapshot that you have taken before, you can click the followings (you need to shut-down the VM first):



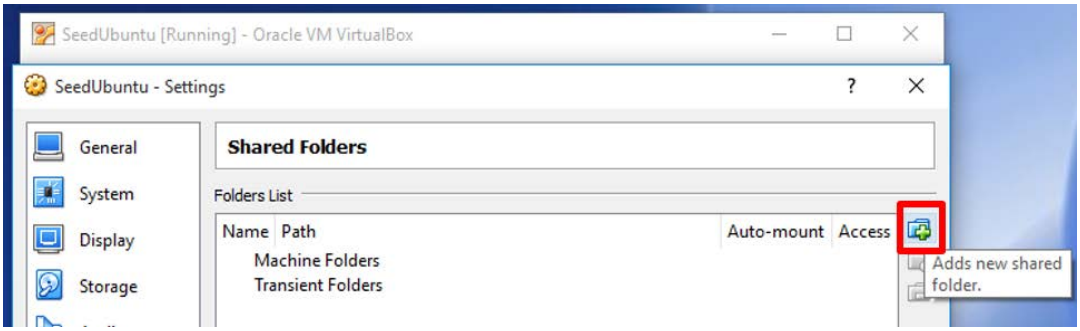
Appendix D: Folder Sharing

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

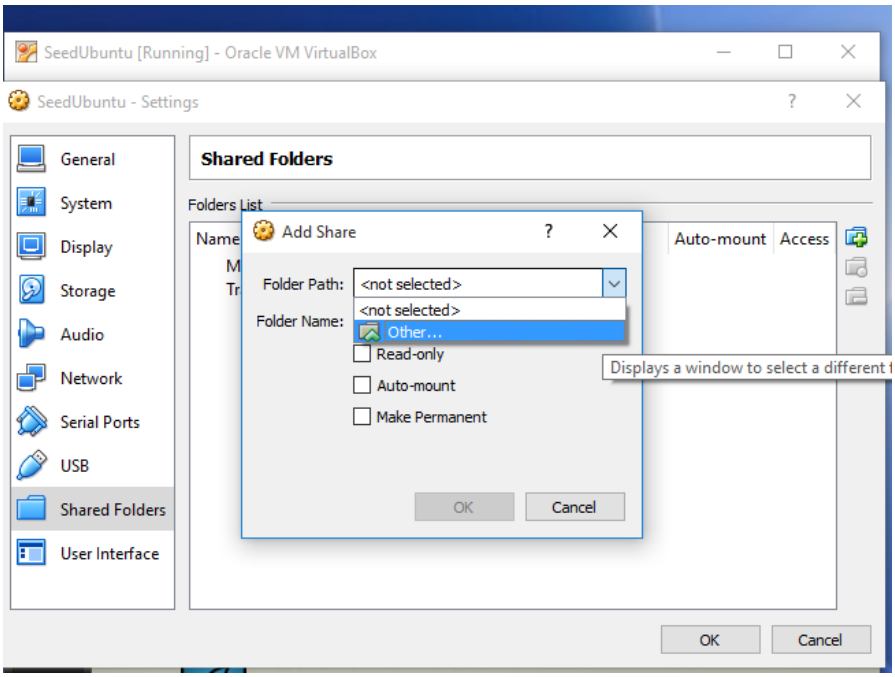
1. Create the folder to be shared on the host computer. In this tutorial we name the folder ***share***.
2. Boot the Guest operating system in VirtualBox.
3. Select Devices -> Shared Folders->Shared Folders Settings...



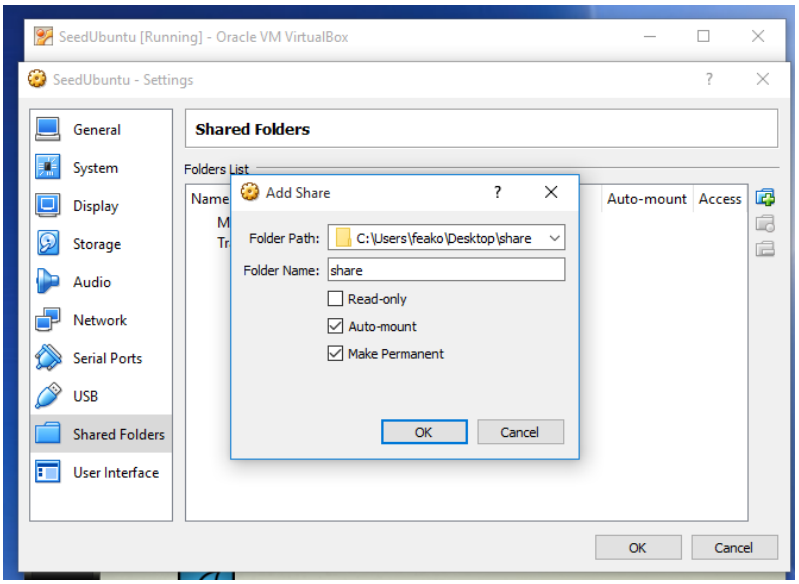
4. Choose the 'Add' button.



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



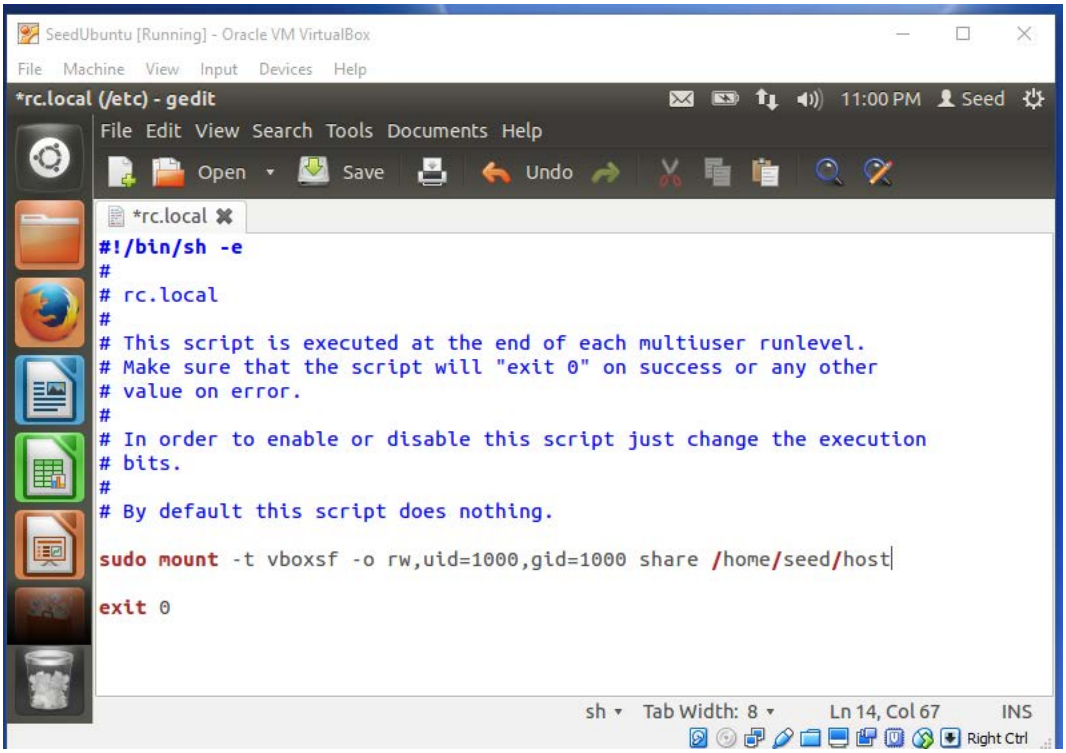
6. Select ***Auto Mount*** and ***Make Permanent*** option. Click OK. Click OK again to close the Settings Dialog.



7. Open a terminal in the VM. Make a directory and name it ***host*** (you can choose any name you like). Use command **`mkdir ~/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 `/etc/rc.local` (**`sudo gedit /etc/rc.local`**) by typing 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 ~/host should be visible from the host in the ~/share folder and vice versa.