Labtainer Student Guide

October 17, 2019

1   Introduction

This manual is intended for use by students performing labs with Labtainers. Labtainers assume you have a Linux system, e.g., a virtual machine. If you do not have a Linux system, our website includes pre-built virtual machine images at `https://my.nps.edu/web/c3o/virtual-machine-images`. Alternately, refer to Appendix A and B for installation of Virtual-Box and a Linux system on either a Mac or a Windows computer. Note that any Linux system can be used as long as it supports Docker.

Labtainers provide a consistent execution environment for performing laboratory exercises, and can simulate the execution of several different computers interconnected via virtual networks.

1.1   Obtaining and installing Labtainers

If you are using one of the pre-configured virtual machines from `https://my.nps.edu/web/c3o/virtual-machine-images`, you may skip this section. The Labtainer framework is distributed as a tarball from: `https://my.nps.edu/web/cisr/labtainers`. Click the link named: "Download the Labtainer framework", and untar the resulting file into a permanent directory on your Linux system, e.g., into `~/home.` For example, if you downloaded the file from a browser on your Linux system:

```
cd
tar -xf ~/Downloads/labtainer.tar
```

From the directory into which you untarred the tarball start the installer script:

```
cd labtainer
./install-labtainer.sh
```

This script will install the latest version of Docker and packages required by the Labtainer framework. It will cause your Linux host to reboot when it completes. Note that older Linux distributions, e.g., Ubuntu 14.* lack the *realpath* package, which should be installed prior to using Labtainers.

After the Linux host reboots, open a terminal to your Linux host and change directory to wherever you untarred the tarball, e.g., your HOME directory.

2   Performing a Lab

All labs are run from the same Labtainer workspace directory, which is typically at:

```
cd ~/labtainer/labtainer-student
```

The prepackaged virtual machines automatically start a terminal in this directory.

To see a list of available labs, run the `labtainer` command with no arguments:

```
labtainer
```

Then run a specific lab, include the name of the lab:

```
labtainer <labname>
```

where *labname* is the name of the lab to run.

Most labs direct you to a PDF version of a lab manual, which can usually be done by right clicking on the displayed path, or you can open the file in a browser. Please note that some of the initial lab instructions repeat the steps you've already taken, and you need not perform those again.

A list of labtainer commands can be found in Appendix C of this document.

Once you start the lab, you will typically see one or more virtual terminals conneted to computers within the lab. While running the lab, if you require more virtual terminals, use:

```
moreterm.py <labname> <container>
```

where *container* is the host name of the component on which to attach a terminal. It can be omitted for labs having a single component.

The virtual terminals for most labs present bash shells via which you can interact with the attached computer, (which is actually a Docker container designed to appear like a separate computer). A single computer may have multiple virtual terminals attached to it. Each computer is independent, and may use networks to interact with other Labtainer computers within the lab.

Many labs automatically gather results of your work, which you will provide to your instructor. Note that, unless otherwise directed, exploration and experimentation you perform either before or after performing the expected activity will not diminish or dilute your results. And you typically do not have to take actions to collect or record your results. This occurs automatically as noted in the next section.

2.1 Interrupting and Completing Labs

When you want to stop working for a while or are finished and ready to turn it in to your instructor, type:

```
stoplab
```

from the Linux system from which you issued the `labtainer` command. All changes to the files, etc. will be preserved and you will be able to resume the lab just the way you started it. You can resume your work, as needed.

The `stoplab` command always displays the directory containing a zip file that should be provided to your instructor. It shows the current results of your work.

The easiest way to forward the complete zip file to the instructor is to start a browser, e.g., Firefox, on the VM from which you are running Labtainers. Then use the browser to either email the zip file, or upload it into an LMS system, e.g., Sakai. Alternately, you can configure the VM to use a shared folder, and use that to copy the zip file to the host computer.

2.2 Redoing a Lab

Sometimes you might want to redo the lab from the beginning. In this case, type:

```
labtainer -r  <labname>
```

This will delete any previous containers associated with this lab and start it fresh. **Warning**: this will cause all previous data from the named lab to be lost.

2.3 Checking your work

Some labs include criteria by which to automatically assess your progress. Where enabled and supported, this feature can be utilized by issuing the `checkwork` command from Linux system. That command can be run while the lab is still running.

2.4 Getting Help and Things to Avoid

To get help, type:

```
labtainer -h
```

from the Linux system from which you issued the `labtainer` command. A list of useful labtainer commands will be displayed.

Do not run multiple labs simultaneously. Consistent results cannot be guaranteed when more than one lab runs at the same time.

2.5 Networking

In addition to network properties defined for the lab, each component `/etc/host` file includes a "my_host entry" that names the host Linux. Most containers will include a default gateway that leads to the Linux host. This allows students to scp files to/from the container and host. It also allows the student to reach external networks, e.g., to fetch additional packages in support of student exploration.

In some instances, the lab requires one or more components to a have different default route. Typically, these components will include a *togglegw.sh* script that the student can use to toggle the default gateway between one that leads to the host, and one defined for the lab. This allows students to add packages on components having lab-specific default gateways. Use of the *togglegw.sh* script is not necessary to reach the Linux host, (e.g., to scp files).

2.6 Installing and Using Labtainers Behind a Web Proxy

If you are not behind a web proxy, ignore this section (most school environments are not behind proxies). If you are behind a web proxy, Labtainer installation requires that you have configured your Linux package management configuration to reflect the proxy, e.g., the /etc/atp/apt.conf or /etc/dnf.conf files.

Additionally, you will need to configure your Docker service as described at: `https://docs.docker.com/engine/admin/systemd/#httphttps-proxy` And set the HTTP_PROXY environment variable to your proxy, e.g.,

```
HTTP_PROXY=http://myproxy:3128
```

If you wish to use apt-get from within a container to add new software to a container, you must first modify the container's /etc/apt/apt.conf file to reflect your proxy.

2.7 Limitations

The Labtainer "computers" are individual Docker containers that are interconnected via virtual networks. These containers each share the Linux kernel of your host. Thus, a change to the kernel configuration on one computer, (e.g., enabling ASLR), will be visible on other containers, as well as your host.

It is suggested that the student's Linux host be a virtual machine that is not used for purposes requiring trust. Software programs contained in cybersecurity lab exercises are not,

in general, trusted. And while Docker containers provide namespace isolation between the containers and the Linux host, the containers run as privileged. Labtainers run as Docker containers and use the Docker group which is root-equivalent. In other words, even though you start a Docker container as a non-privileged user, software in the resulting container can modify the Linux host, e.g., the VM.

The computers each include a "local" directory beneath the HOME directory. This is used by the Labtainer framework and includes results that get packaged up for forwarding to the instructor. Do not modify any files beneath the .local directory. Otherwise, you can treat those containers as Linux systems, and explore them.

Pasting multiple commands into a labtainer terminal may result in the not all of the commands being executed.

Appendix A: Installing Virtual Box and Ubuntu on Mac OSX

The instructions below describe installation of an Ubuntu Linux VM to serve as the Labtainer host on a Mac. If you already have a Linux system that can support Dockers, you may use that system and not use this appendix.   If you already have VirtualBox installed on your Mac, be sure it is updated to the latest version.

• Install Virutal Box from : https://www.virtualbox.org/wiki/Downloads

      Click on the "OS X hosts" link.

      Install per your OS' usual installation procedure…

• Download the latest Ubuntu Desktop LTS distribution *.iso image*.   It is important you download this as an .iso image as that will be used to install Ubuntu on the VM you create. https://www.ubuntu.com/download

• Use VirtualBox to create a new VM, allocate at least **10GB of disk** storage, **4GB**

Download Ubuntu Desktop
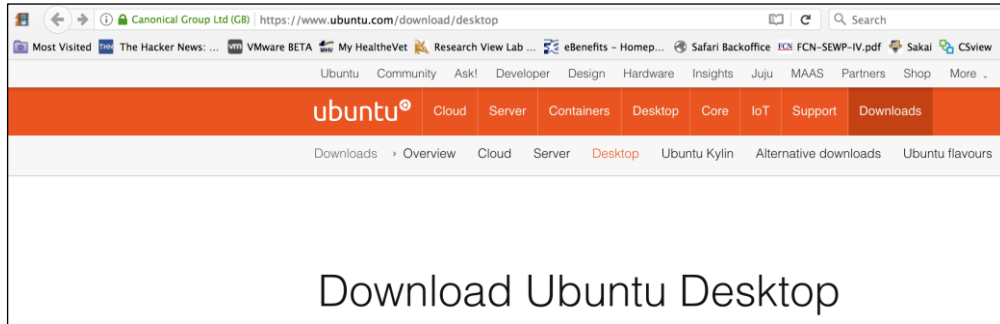
RAM and **2 CPUs**.
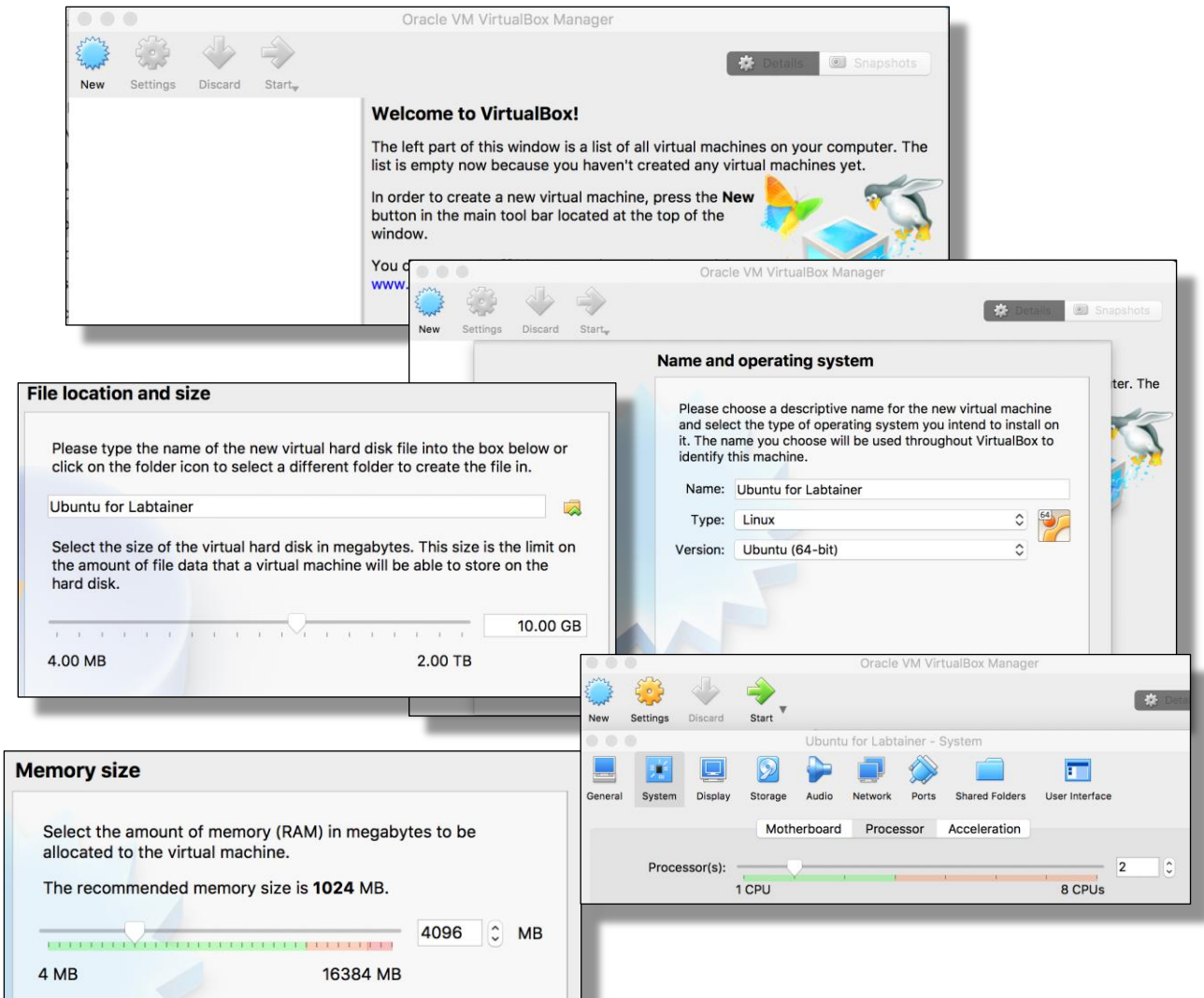
• Select the Ubuntu iso image in the VirtualBox storage settings, and select "Live CD/DVD"
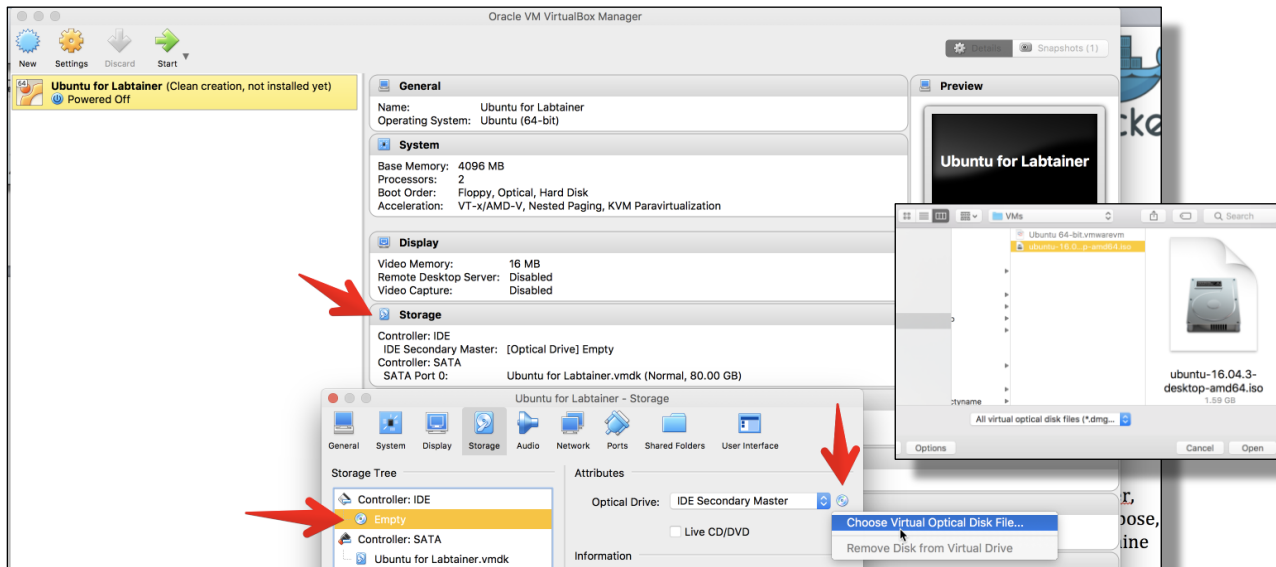
• Power on the virtual machine to install Ubuntu.

- If the VirtualBox window shows the following screen, the previous step did not work, please remount the ISO by selecting the little folder icon to the right.
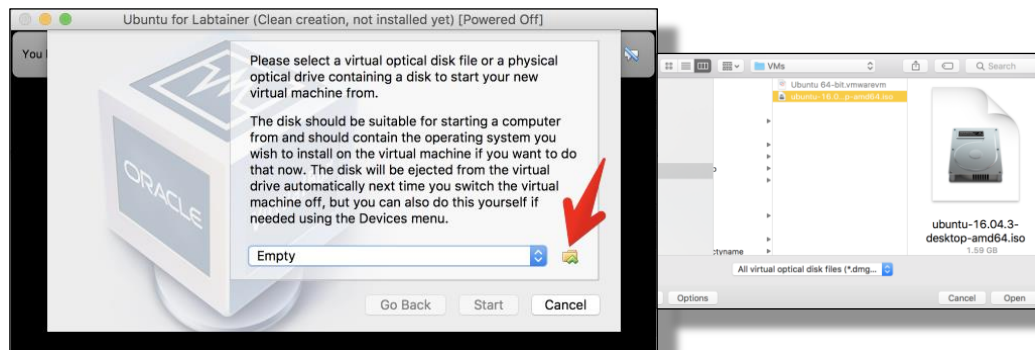
- Once you see the following desktop, you can install Ubuntu
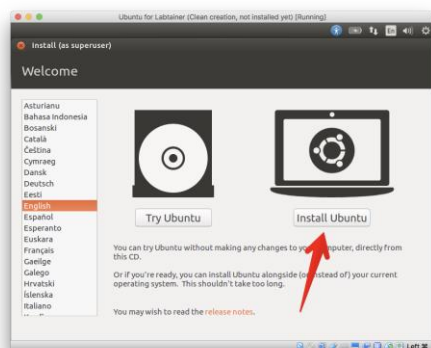
You should be able to install Ubuntu by accepting the default options provided. It is recommended, though not required, you do select the following options:

- Download Updates while Installing Ubuntu
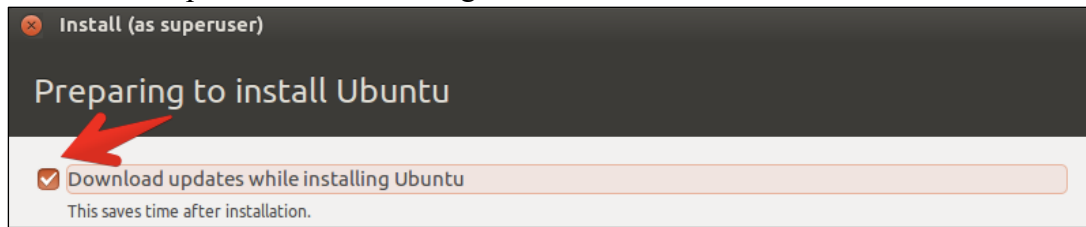
- Set your Time Zone
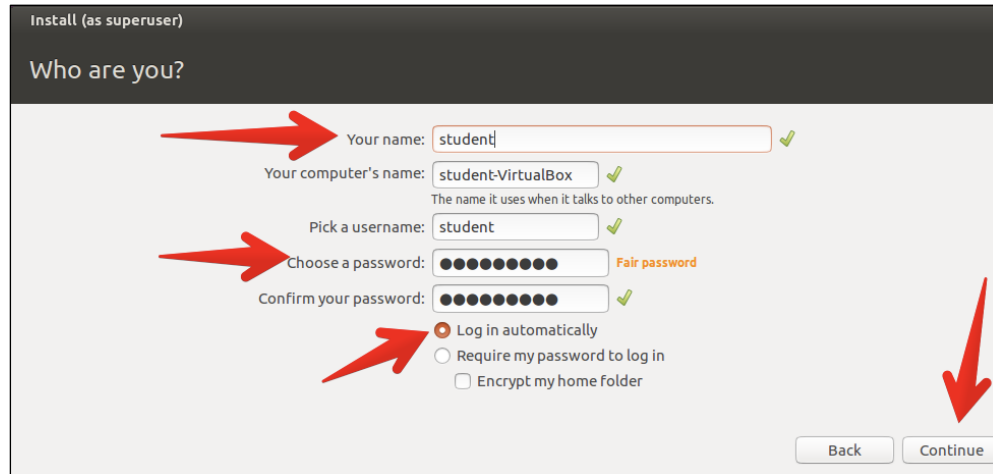
- Select your appropriate language

When prompted, enter a user name and password:

{student is used here in the example, you may use whatever name you like}

Ubuntu will then install on to the virtual hard drive.

Once the files are installed, you will need to restart the VM.
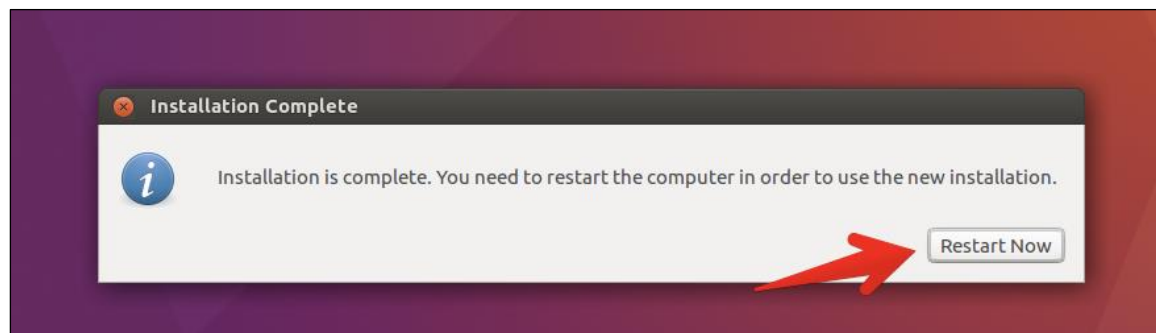
NOTE: be sure to unmount the ISO file before restarting. If you don't you will see the following screen:
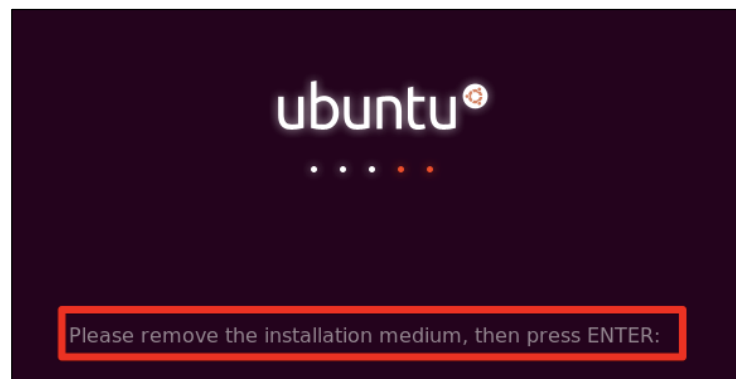
To unmount the ISO, just choose the DVD icon in the lower left corner and select **Remove disk from virtual drive** (note if that is greyed out, first click on the ISO line above it then click on the Remove disk from virtual drive)

If it displays a message asking to force the unmount, that is okay, just select the **Force Unmount**
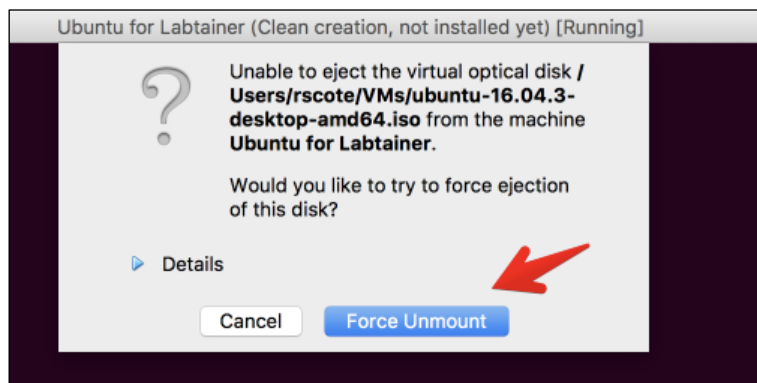
Once the VM reboots you should have a screen that looks like this:
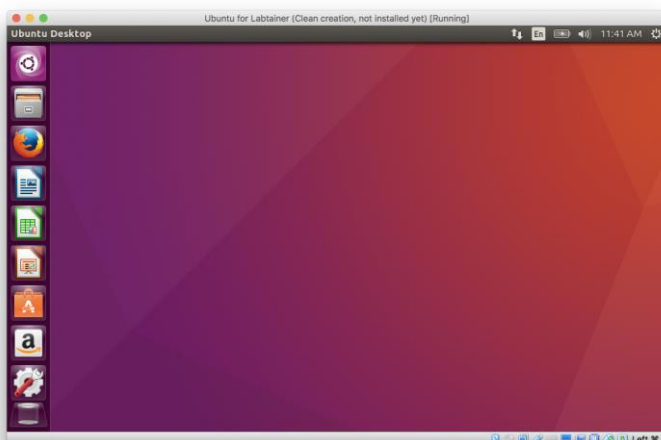
Note: If your Ubuntu desktop has a DVD icon on it, you did not unmount it correctly and should be unmounted by right clicking and ejecting the DVD ISO.

The following step simplifies movement of files between Virtual Box guests and the host computer.

Setup shared folder:

• create or identify a folder on the host to share with the guest
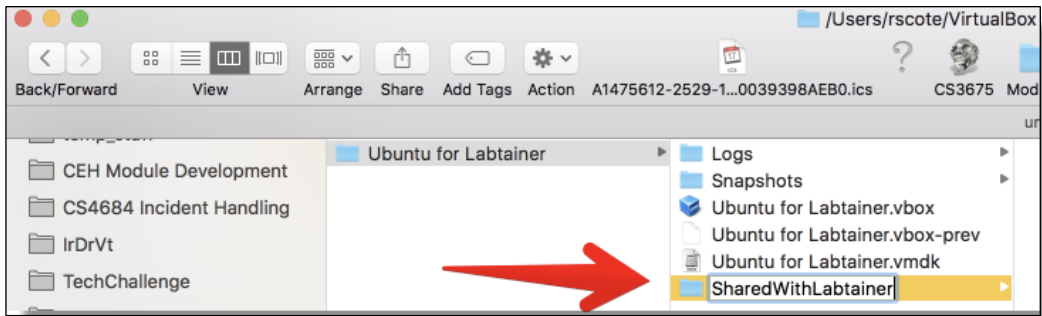
• On the VirtualBox window, select the "Shared Folders" icon.

Then select the **add folder** button on the right, and add in the folder you created above, being sure to select **Auto-mount** and **Make Permanent.**
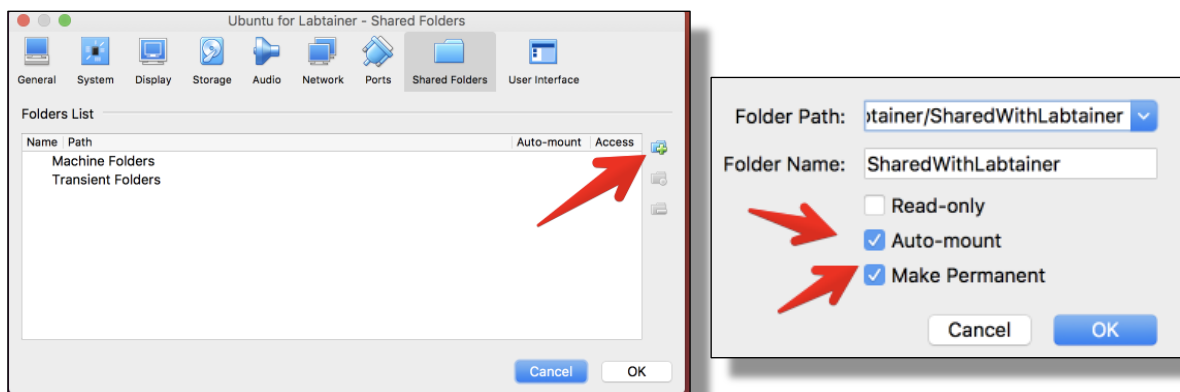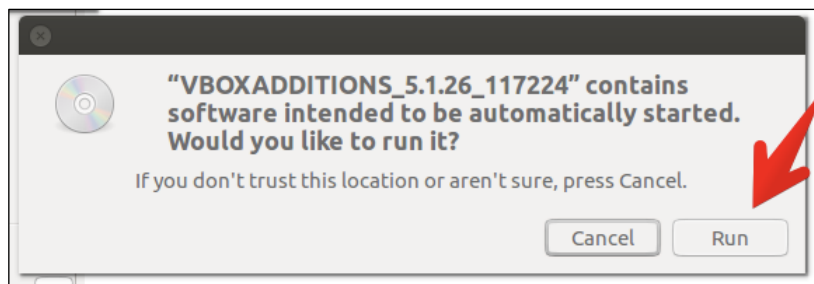
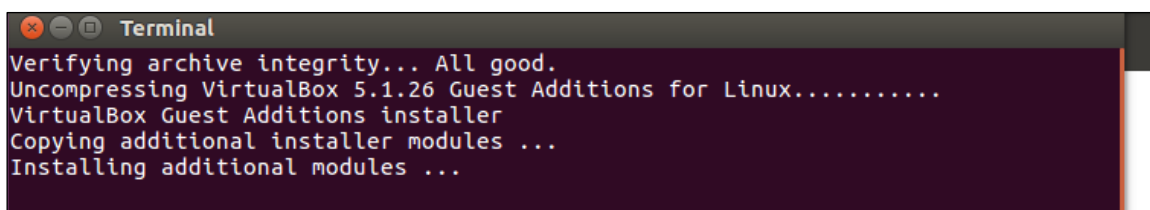- Install VirtualBox Guest Additions

  VirtualBox Guest Additions contain device drivers and system applications that optimize the operating system for improved performance.
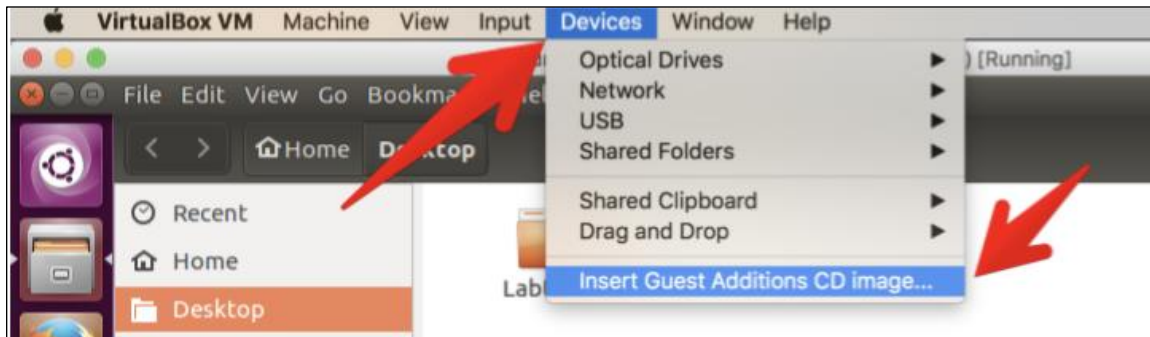
  To install them, choose **Devices** from the *VirtualBox VM Menus,* then

 select   **Insert Guest Additions CS Image…**

 This will *auto-launch the install program*. Choose **Run**

  You will then see the guest additions load, and finally ask you to *hit return to exit*.

- In order to cut and paste from and to the VM, lets enable sharing the

clipboard. To do this, go to the settings in the VirtualBox Manager (NOT the individual VM's window) then choose *Settings, General, Advanced, and allow for Shared Clipboard.*
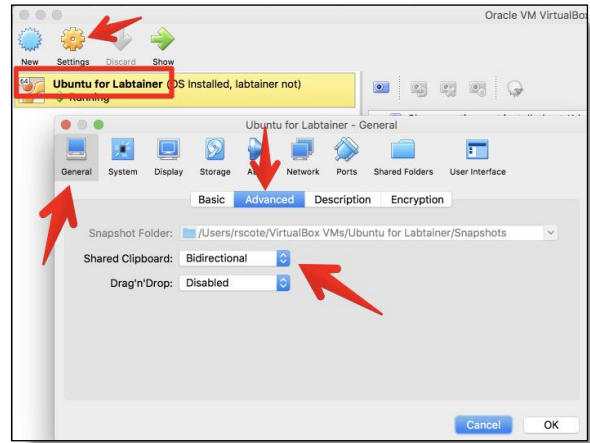
Then, open a terminal
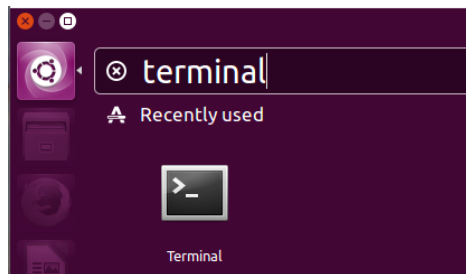
and run this command in the terminal by typing:

    sudo adduser $USER vboxsf

Then, reboot the guest Linux system:

    sudo reboot

Appendix B

Installing Virtual Box and Ubuntu for Windows

The instructions below describe creation of an Ubuntu Linux Virtual Machine (VM) to serve as the Labtainer host. If you already have a Linux system that can support Dockers, you may use that system and not use this Appendix.

Installation of the Virtual Box

You will install VirtualBox from: https://www.virtualbox.org/wiki/Downloads.
Choose VirtualBox for Windows hosts and install per the usual procedure (by downloading the .exe   le, then selecting it and clicking "Run").
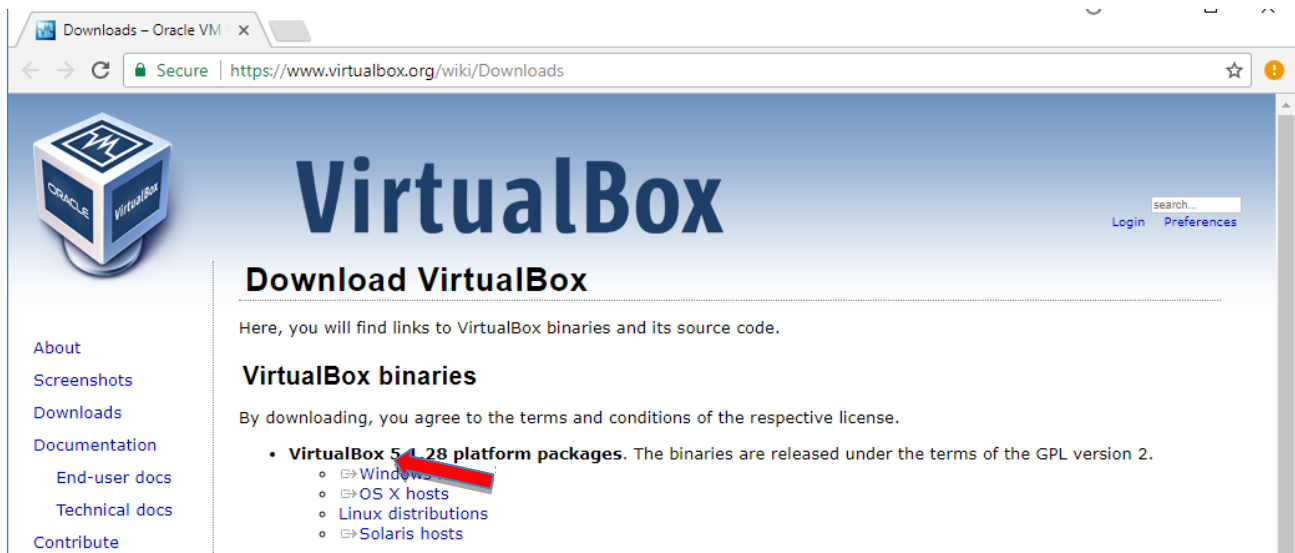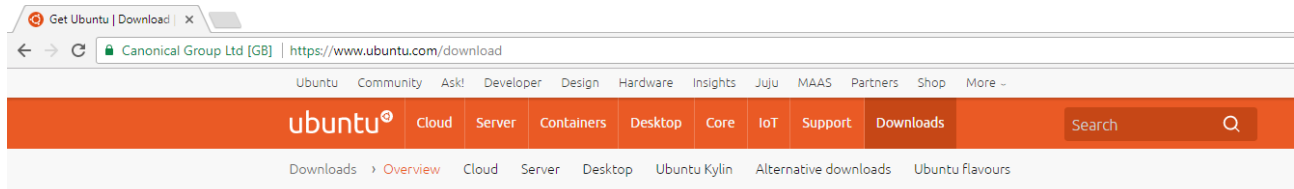
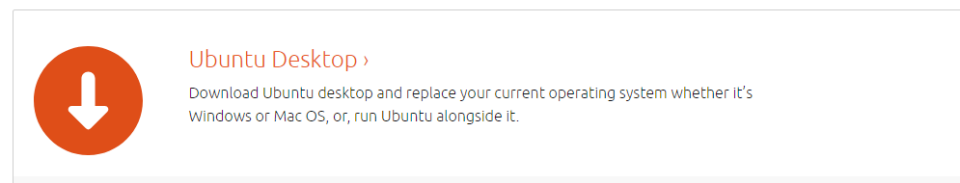Then follow the setup process keeping all the options set to their default.

.

Installation of Ubuntu

You will download the latest Ubuntu Desktop LTS distribution *.iso image* from
https://www.ubuntu.com/download. It is important that you download this as an .iso (i.e., as
Disk image   le) because it will be used to install Ubuntu on the VM you create.

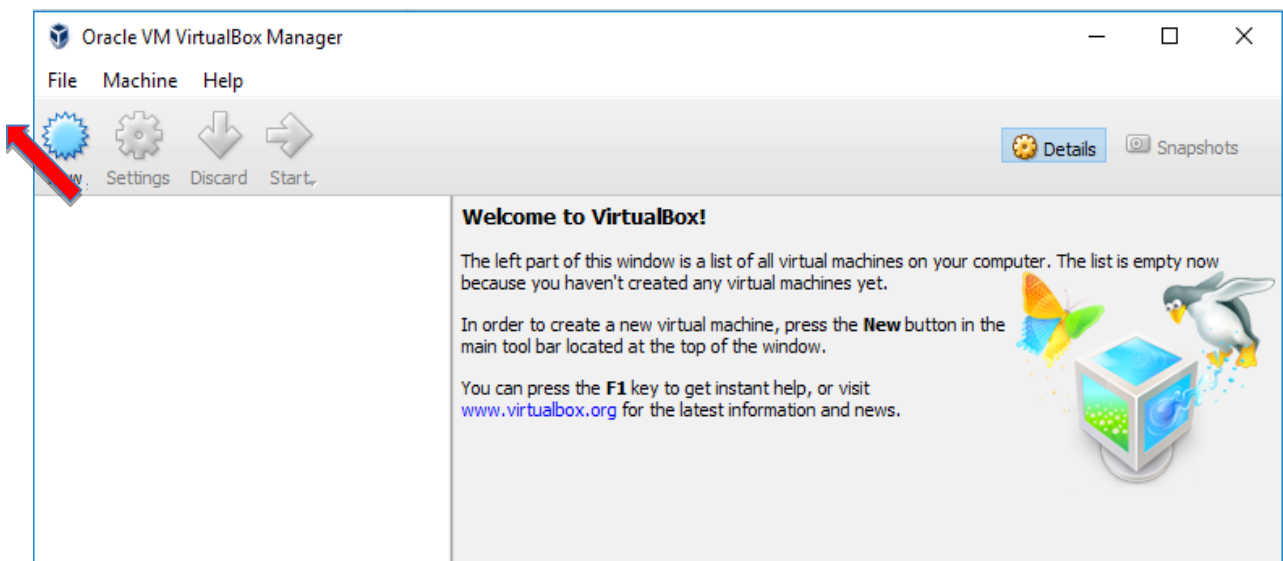## Creating a new Virtual Machine (VM)

1. You will use VirtualBox to create a new VM (new hard disk   le) by clicking on the
   "New" button in the top left corner.

2. Follow the instructions to name your VM and choose what operating system (OS) you will be installing. Choose Linux from the "Type" menu. You must install the 64-bit version of the OS, select Ubuntu 64-bit under Version.

NOTE: If you are using a machine that supports 64-bit but the 64-bit version is not available in this dropdown menu, you may need to go to your PC's **Windows Features/Control Panel** (on Win10: Control Panel / Programs / Programs and features; then click "Turn Windows features on or o " on the left side of the window), and disable Hyper V, then restart the computer. Also see https://forums.virtualbox.org/viewtopic.php?f=1&t=62339 for VirtualBox hardware requirements for running 64-bit guests.

Name and operating system

Please choose a descriptive name for the new virtual machine and select the type of operating system you intend to install on it. The name you choose will be used throughout VirtualBox to identify this machine.

Name: Ubuntu for Labtainers
Type: Linux
Version: Ubuntu (64-bit)

3. Next you need to allocate RAM (memory), disk storage space, create your new hard disk, and allocate a number of CPUs (Central Processing Units) to your new VM:

- Allocate **4 GB RAM when prompted**

Memory size

Select the amount of memory (RAM) in megabytes to be allocated to the virtual machine.
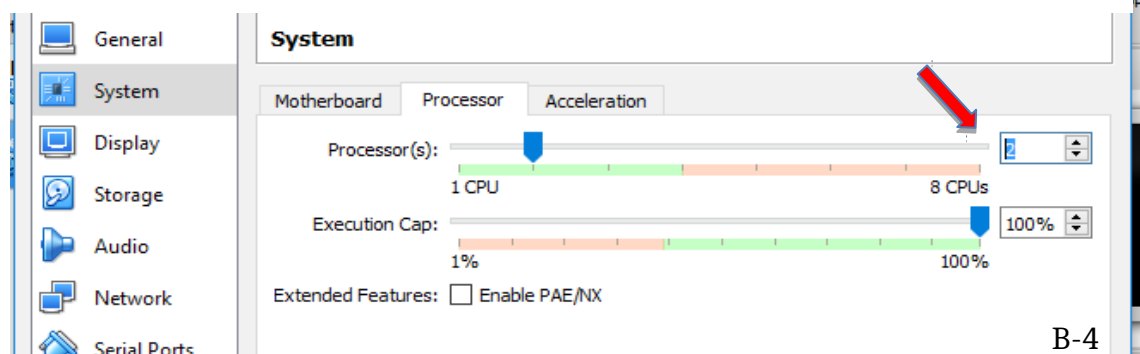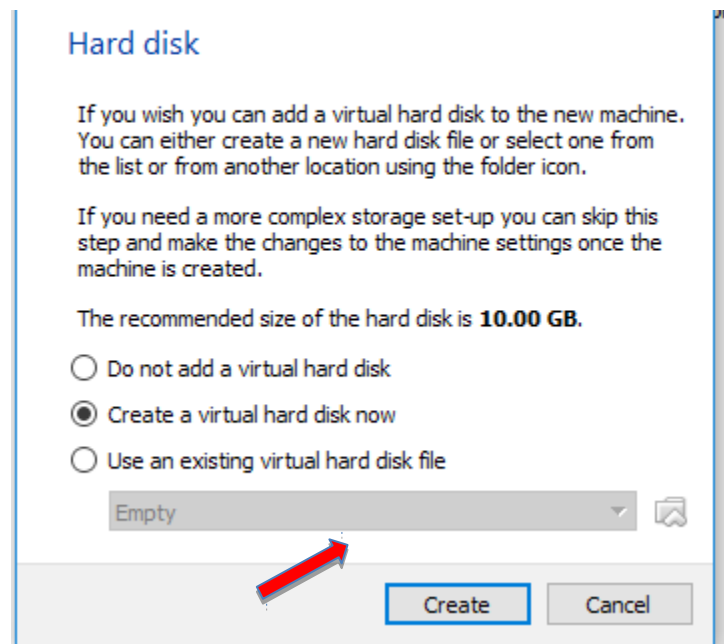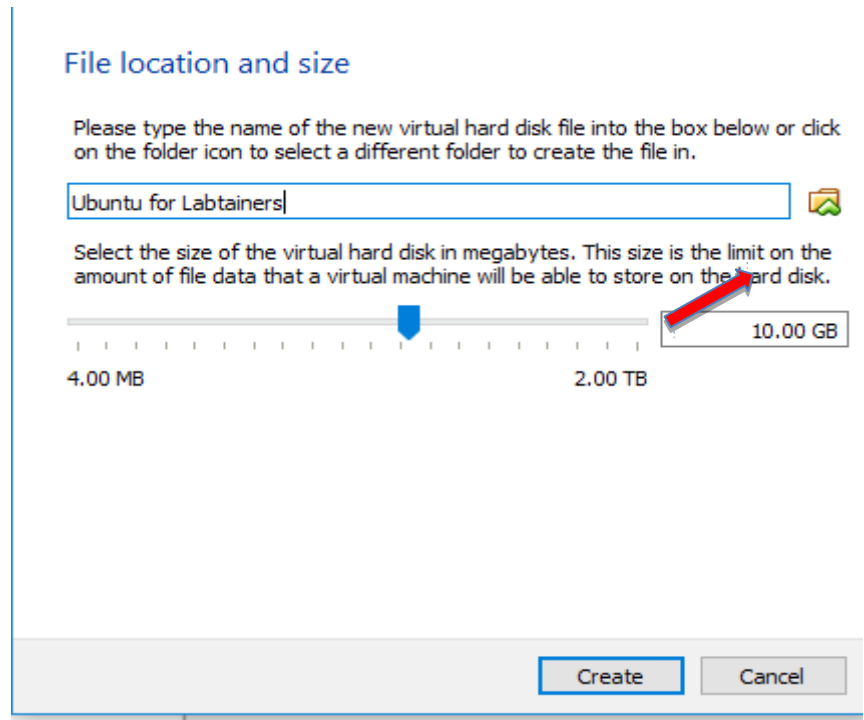
The recommended memory size is **1024** MB.

4096 MB

4 MB   16384 MB

B-3

- Create your new virtual hard disk

- On the screen that will follow the one on the right (not shown here), accept the default disk le type (**VDI**)

- And on the screen after that one (also not shown), accept "Dynamically allocated"

    - Allocate at least **10 GB of disk storage** and click "Create"

    -

    - After you click "Create", you should be taken to the VirtualBox main window

Your new VM is visible in the top left corner. It is currently powered off.

- Select "Settings", then "System", then click on the "Processor" tab as shown below and assign 2 CPUs if your system allows.

File location and size

Please type the name of the new virtual hard disk file into the box below or click on the folder icon to select a different folder to create the file in.

Ubuntu for Labtainers

Select the size of the virtual hard disk in megabytes. This size is the limit on the amount of file data that a virtual machine will be able to store on the hard disk.

4.00 MB   2.00 TB

10.00 GB

Create   Cancel

Hard disk

If you wish you can add a virtual hard disk to the new machine. You can either create a new hard disk file or select one from the list or from another location using the folder icon.

If you need a more complex storage set-up you can skip this step and make the changes to the machine settings once the machine is created.

The recommended size of the hard disk is **10.00 GB**.

○ Do not add a virtual hard disk
◉ Create a virtual hard disk now
○ Use an existing virtual hard disk file

Empty

Create   Cancel

| General | **System** |
System	
Display	Motherboard	Processor	Acceleration
Storage	Processor(s):
	1 CPU	8 CPUs
Audio	Execution Cap:
	1%	100%
Network	Extended Features: ☐ Enable PAE/NX
Serial Ports	

2
100%

B-4

4. Now you need to mount your .iso image (OS, Ubuntu). Please note that the Optical Drive is now shown as "Empty" in your VirtualBox main window. Click on "Storage" as shown below.
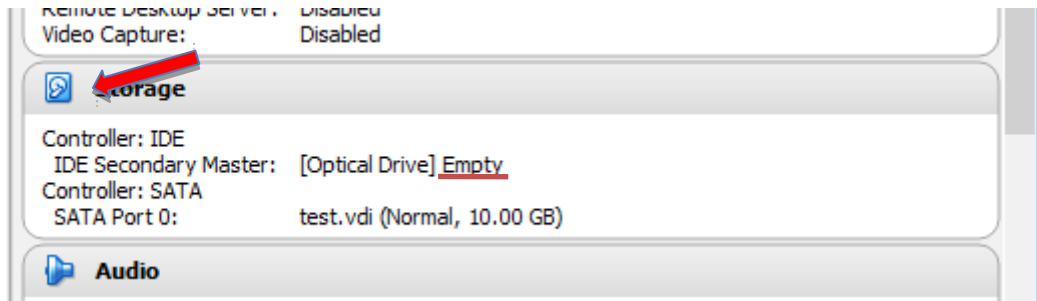
The following window will open (see below). Click to select "Empty" disk on the left under "Controller: IDE" first (not shown). Then make sure that you **check "Live CD/DVD"** as shown below (it is very important that you add your Ubuntu .iso as Live CD/DVD)
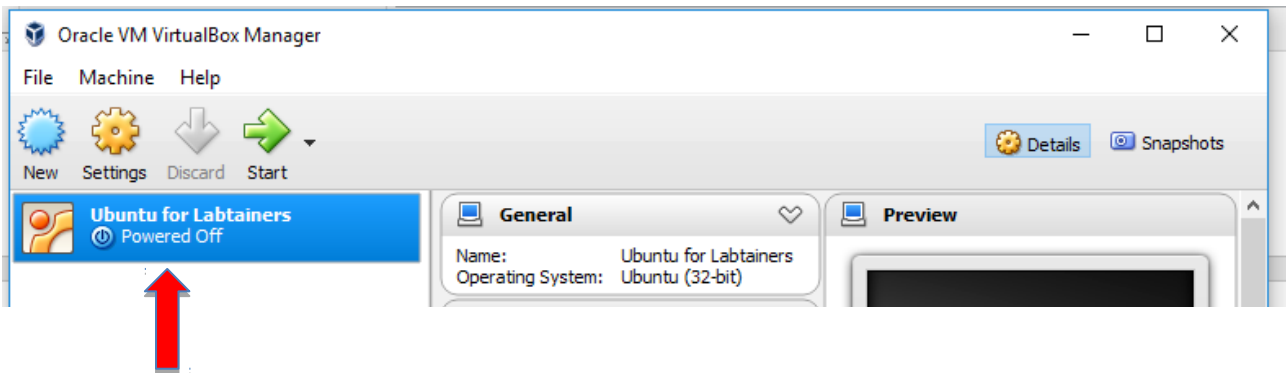
Now click on the small disk icon on the right (as shown), select and add the Ubuntu .iso that you downloaded earlier.
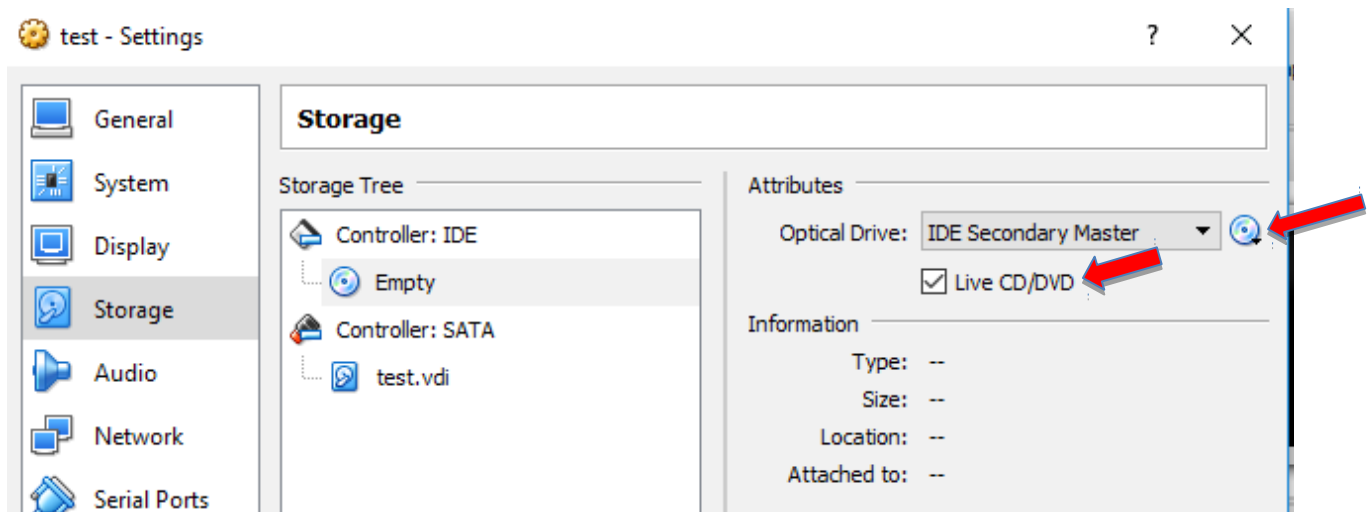
Your Optical Drive should now show your Ubuntu .iso image (as shown below)
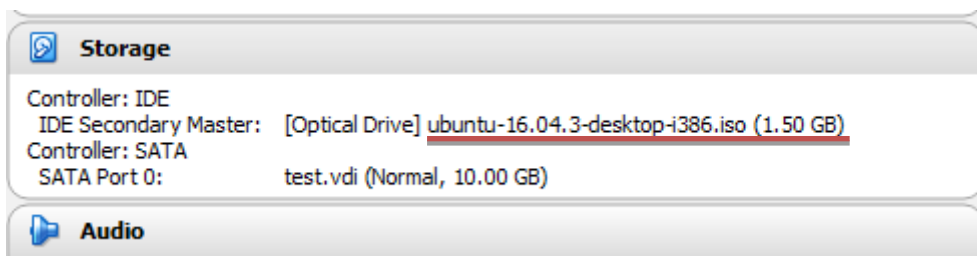
5. Double-click your new VM to power it on. (As an alternative you can click on "Start" and select "Normal start".)


Wait patiently until the Ubuntu Welcome screen (below) appears because the installation will take a few minutes.

Dismiss the noti cations about **Auto capture keyboard** and **Mouse pointer integration** when they appear (both shown in the image below). They are merely telling you that your keyboard keystrokes and mouse clicks will be captured by your VM.
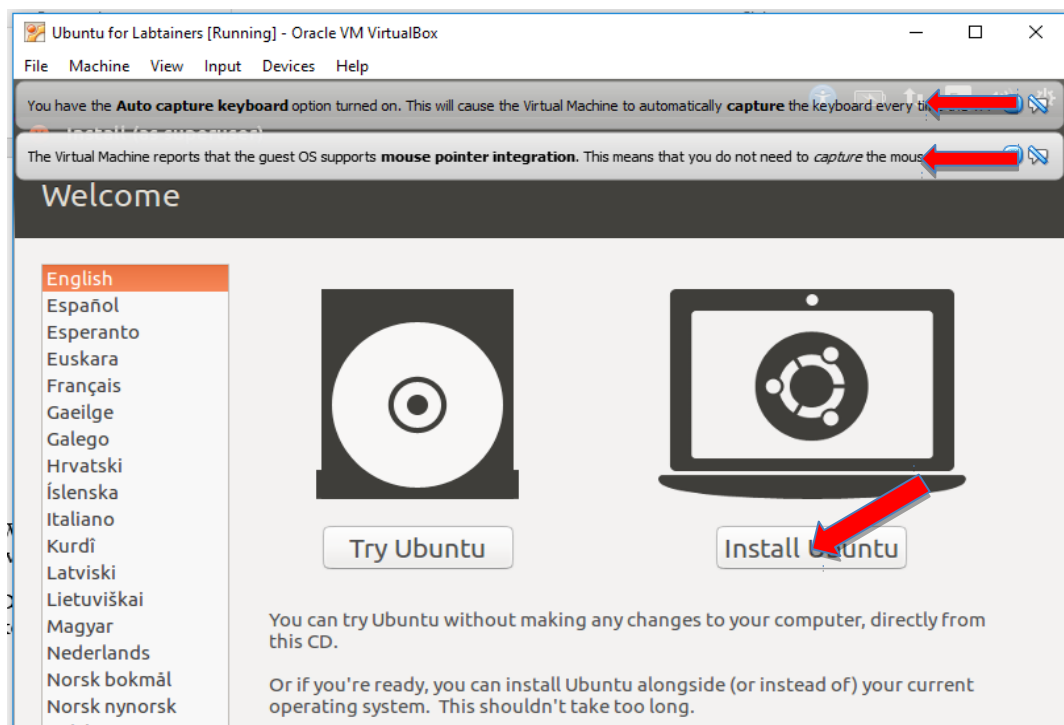
IF at this point, instead of the above Ubuntu Welcome screen above, the VirtualBox window shows the following screen, the previous step did not work, so please remount the .iso by selecting the little folder icon to the right and repeating the steps as prompted (i.e., select Ubuntu ISO image in the VirtualBox storage settings, and select "Live CD/DVD")
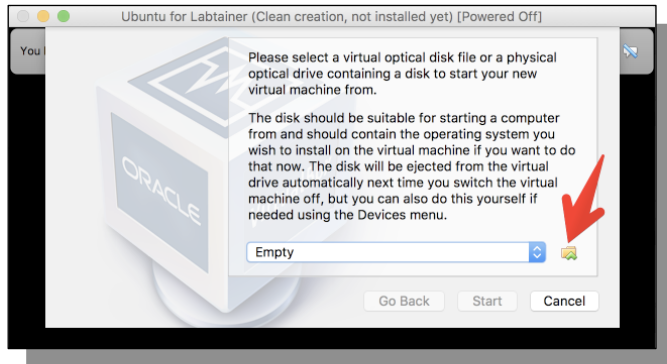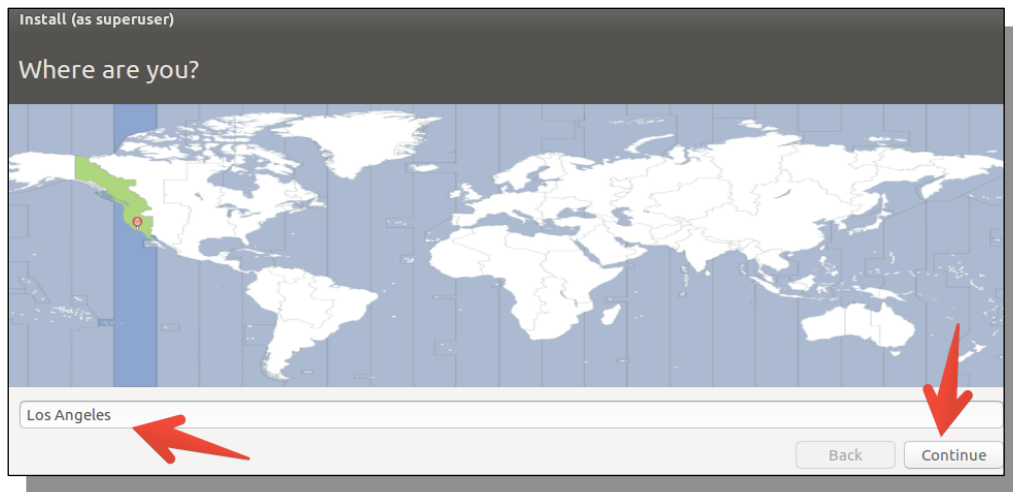

 - Once you see the Ubuntu Welcome Screen, you can click on "Install Ubuntu".

You should be able to install Ubuntu by accepting the default options provided. It is recommended, though not required, that you select the following option: "Download Updates while Installing Ubuntu"
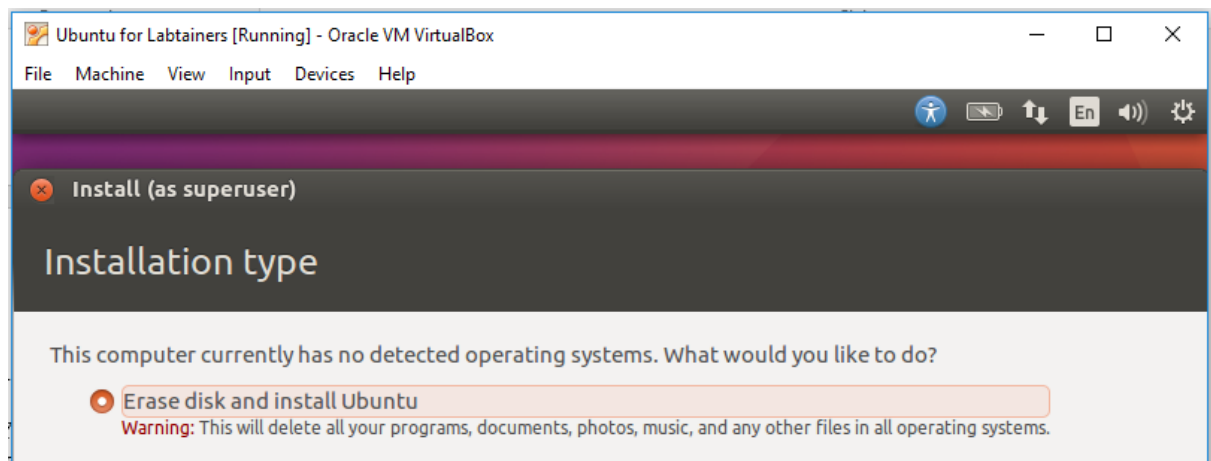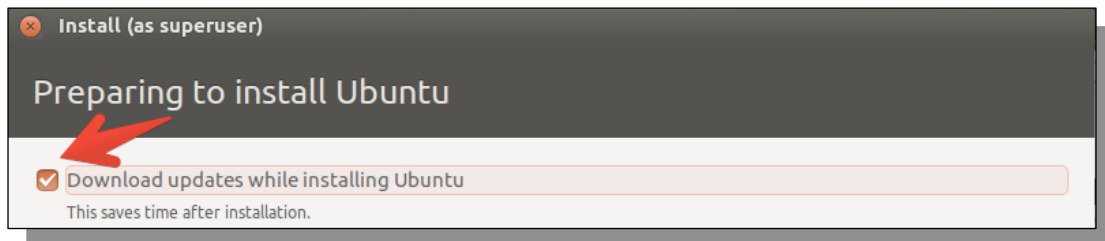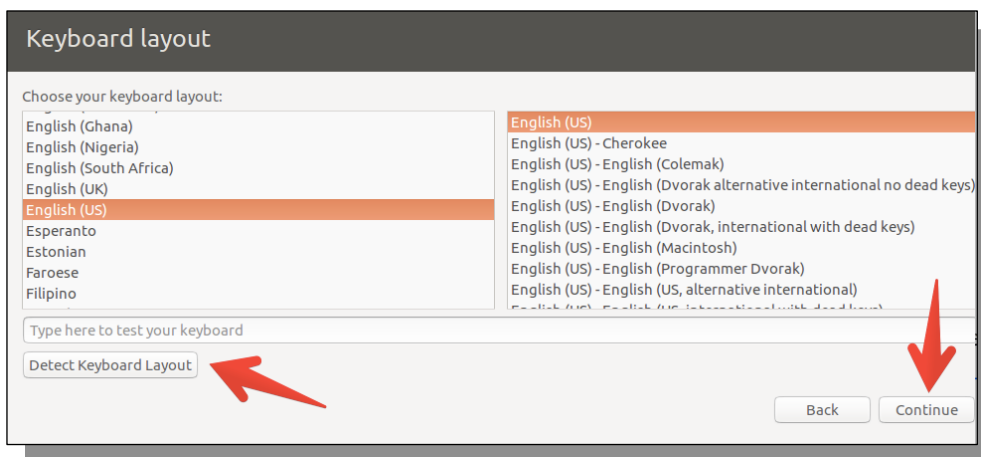

- Select "Erase disk" – this will only a ect your VM disk space (and will not harm   les on
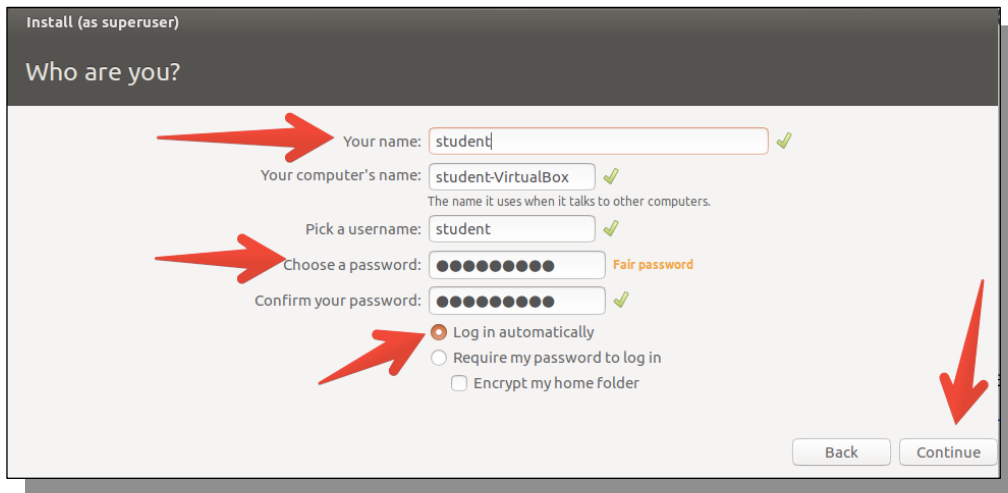
your host PC)

- Set your Time Zone to Los Angeles when prompted

- Select the appropriate language

- When prompted, enter a user name and password: "student" is used here in the example, you may use whatever name you like

Ubuntu will then install on to your new virtual hard drive (VM).

When the installation is complete, a dialogue box will be displayed instructing you to restart your VM. NOTE: **DO NOT CLICK "Restart Now" before you unmount the .iso image.**

To unmount .iso, select "Devices" in the top menu, then select "Optical Disk", make sure the Ubuntu .iso image is checked and click "Remove" disk from virtual drive"

IF you accidentally click to restart before unmounting the .iso le, you will get the following screen:
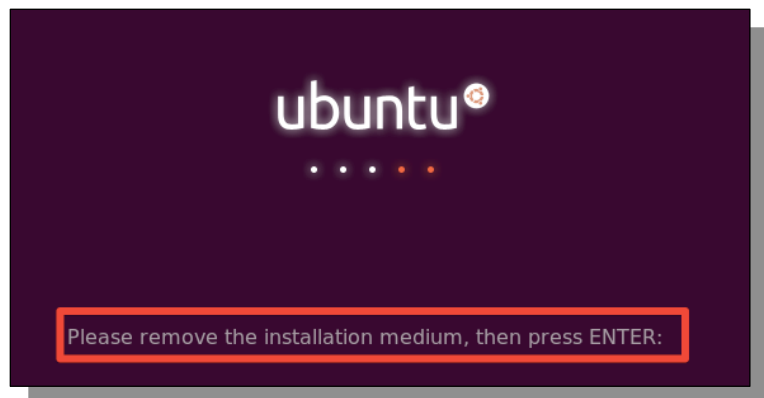

IF this occurs, in order to unmount the .iso, just choose the DVD icon in the lower right corner and select **Remove disk from virtual drive** (note if that is greyed out, rst click on the ISO line above it then click on "Remove disk from virtual drive")
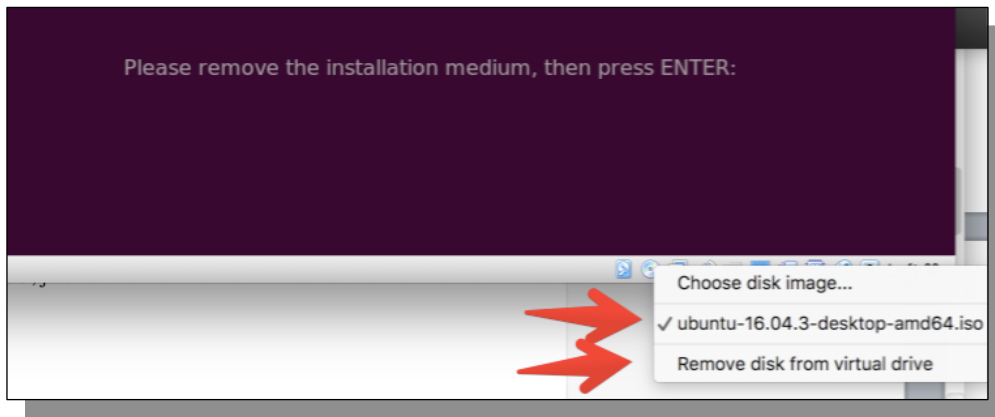
IF the small disk icon is itself greyed out, go to the "Devices" menu and then remove it following the instructions above.If asked if it okay to force the unmount, just select the **Force Unmount**.

Restarting your VM (Virtual Machine)

- Now click "Restart Now"
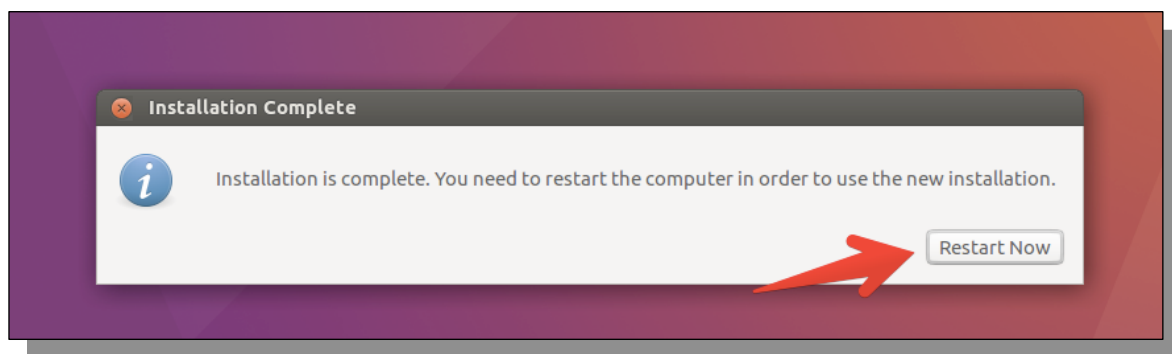
The VM reboot may take a few minutes so you may see a black screen for a while. **IF** you see the black screen for too long, select "Machine" in the top menu and "Reset". **Once the VM reboots/resets, you should have a screen that looks like this:**
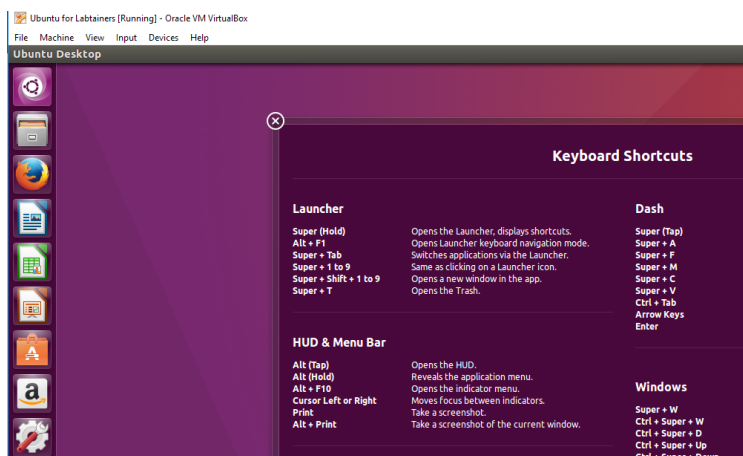
NOTE: **IF** your Ubuntu desktop has a DVD icon on it, you did not unmount it correctly and should do it now by right-clicking and ejecting the DVD .iso.
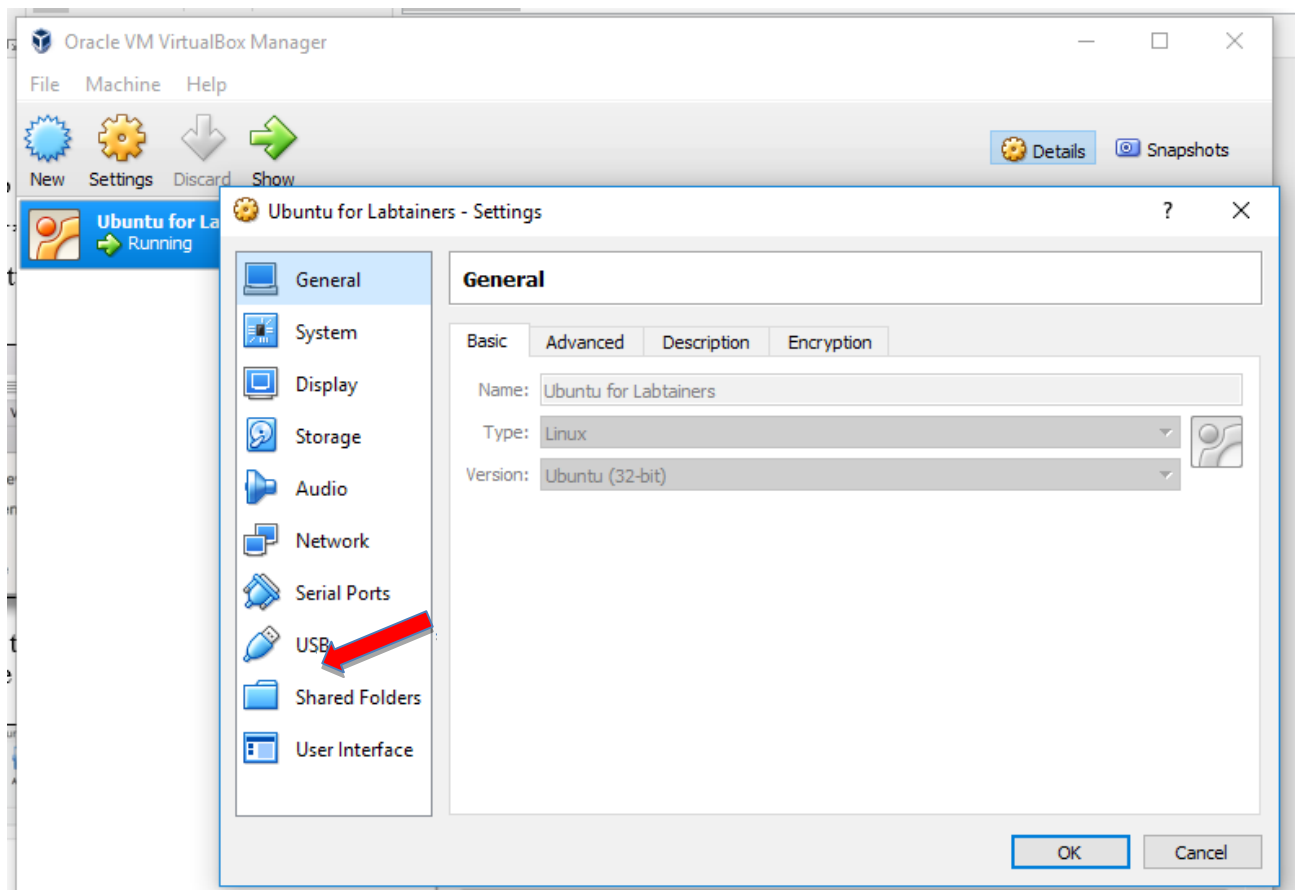
This is your VM. The VirtualBox main window should now look like this:
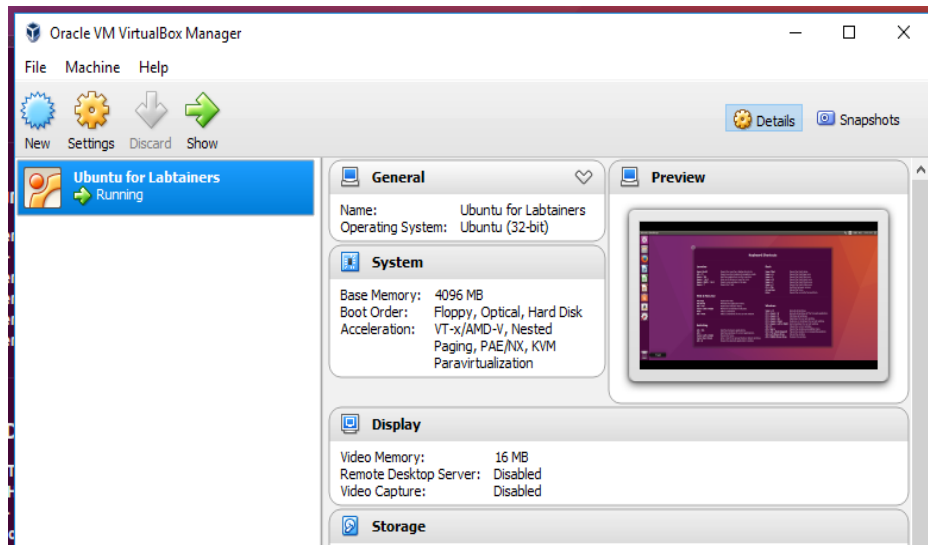
**Setting up ﬁle sharing between the host computer and the VM**

You need to create (or identify) a folder on the host machine that will be sharing ﬁles with the VM (i.e., guest).

In the VirtualBox main window, go into Settings and select "Shared Folders".

Then select the **add folder** (plus) button on the right, locate and add the folder you created/identi ed for sharing, making sure to rst check **Auto-mount** and **Make Permanent** (**IF** you have not yet created a folder on the host that will contain shared les, you can do so within this process.)
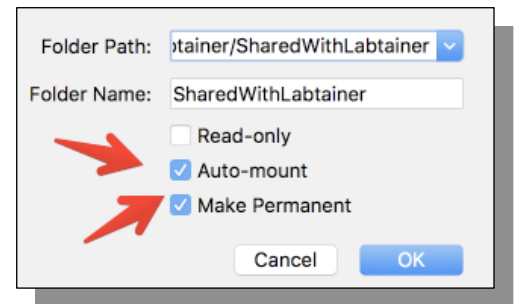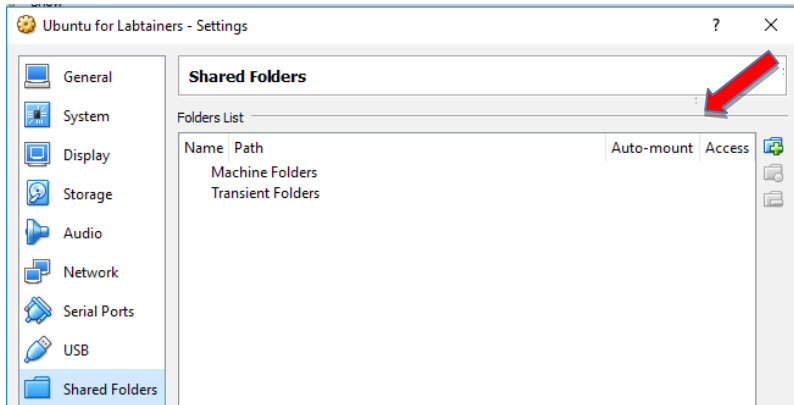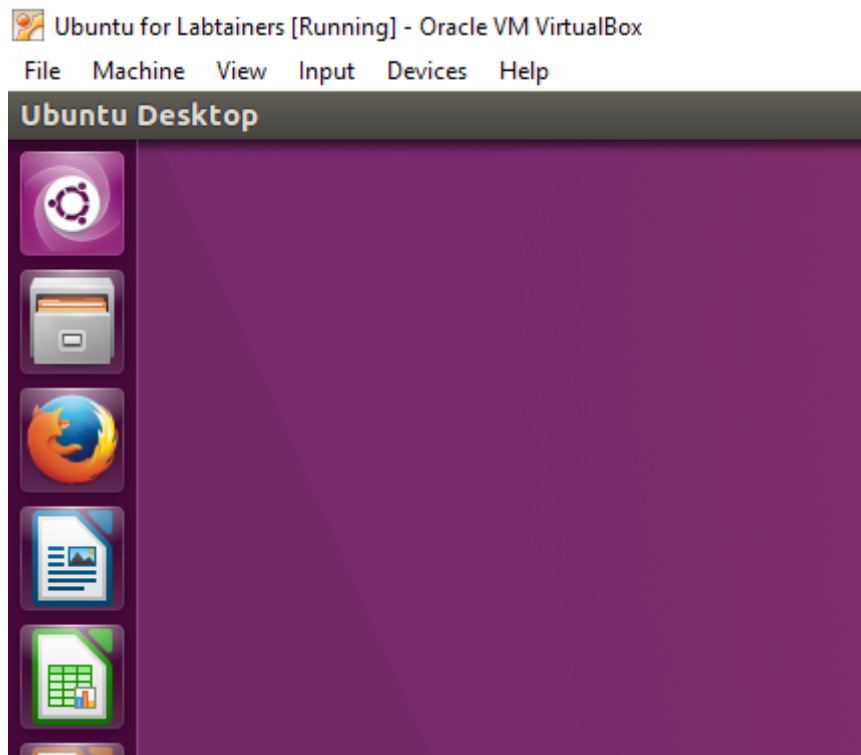

Installing Add-ons for your VM

Now that you have identi ed the shared folder, you need to take the following steps to improve the performance of your new VM. VirtualBox Guest Additions contain device drivers and system applications that optimize the operating system.

In your new VM window, go to "Devices" in the top menu and select **"Insert Guest additions CD Image"**
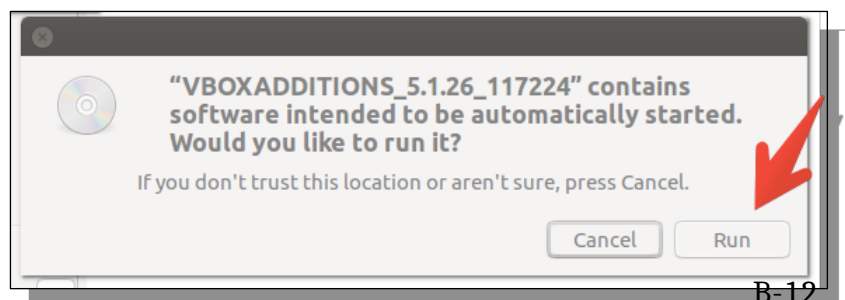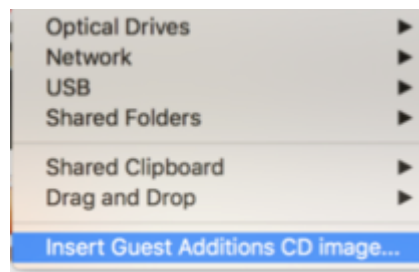
When you see the warning message below, click "Run".

If you are prompted for "authentication as superuser", just enter the password you created earlier when you were naming your VM. (Additionally, any time you are away from your VM, you may be asked to enter your password upon return.)

You will see this screen during auto-install. After the guest additions load, you will be asked to hit Return to exit.


Enabling sharing of the clipboard

In order to be able to copy and paste from and to your new VM, lets enable sharing the clipboard. To do this, choose the VirtualBox Manager (main window, NOT the individual VM's window), go to "Settings", General, Advanced, and allow for Shared Clipboard by choosing "Bidirectional".

(Alternatively, in your individual VM's window, go to Settings in the top menu and select Shared Clipboard, Bidirectional.)
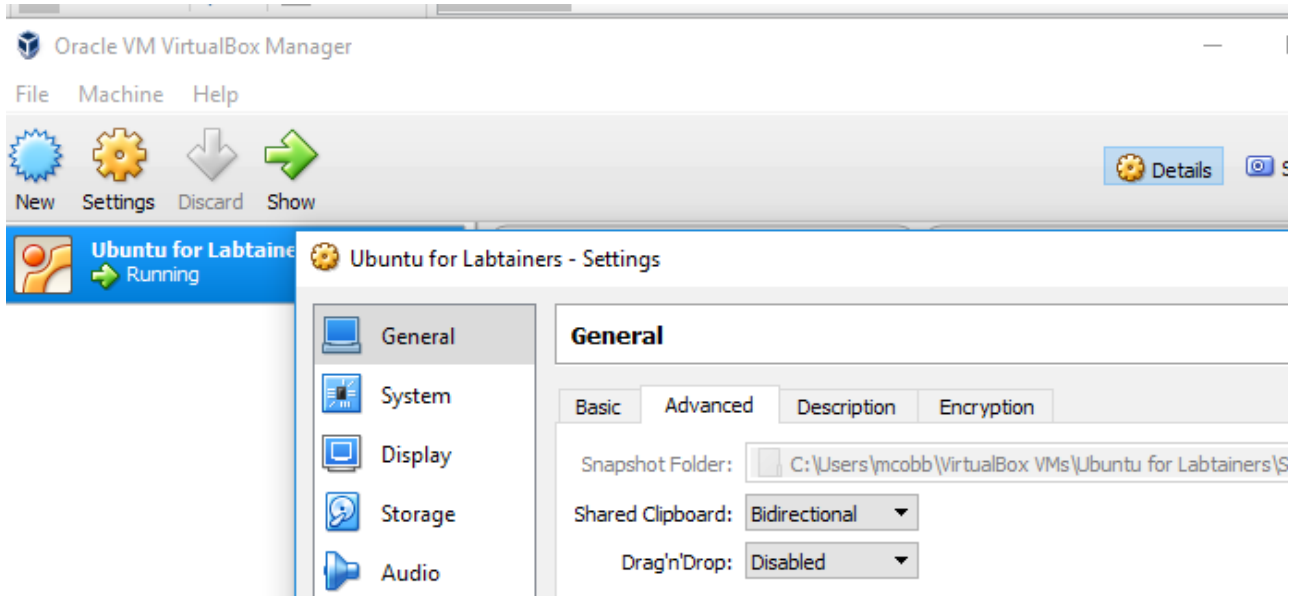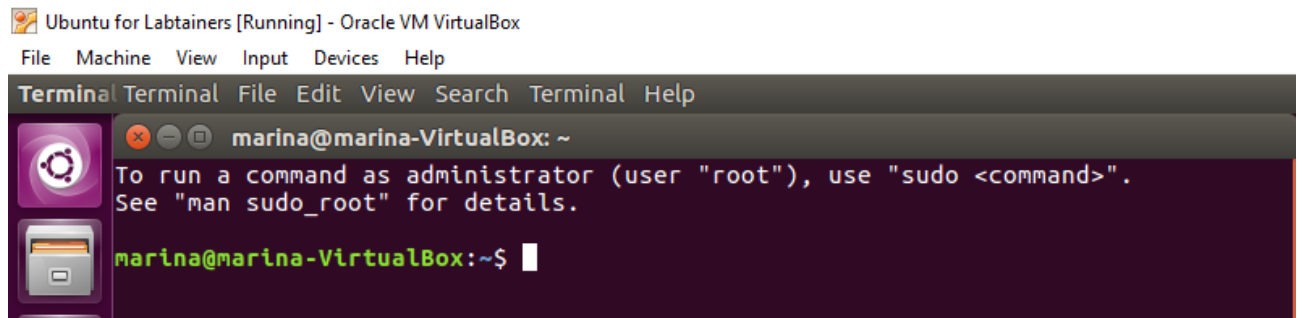

Final steps

- You will now open a terminal. Select the Search icon in the top left corner of your VM screen and type "terminal" in the window that will pop up and hit Return.
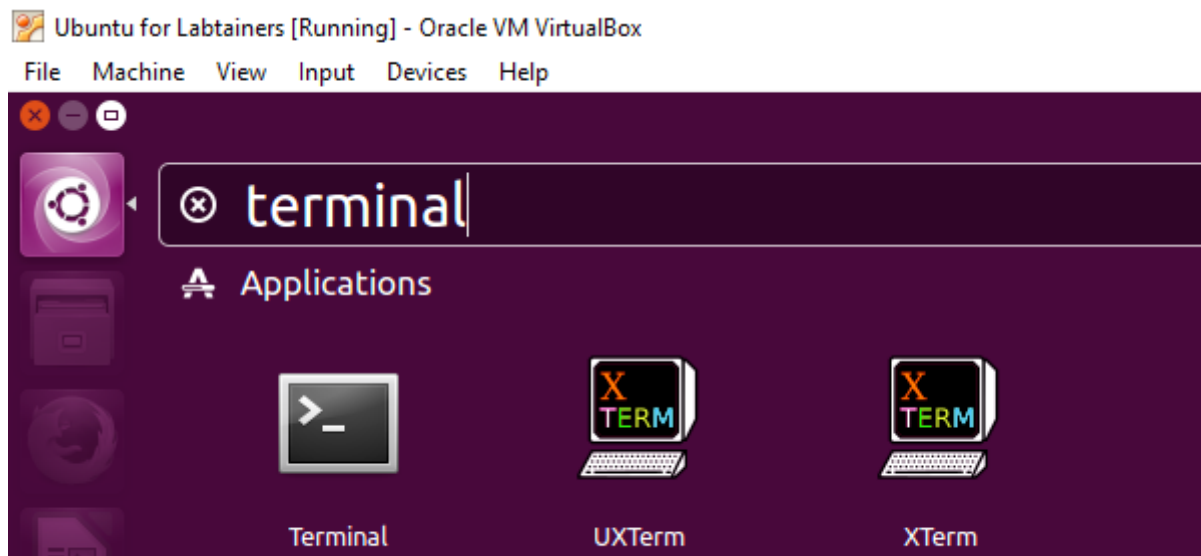
- After you select "Terminal" under Applications, you will see this screen (it will display the name of your machine):

Ubuntu for Labtainers [Running] - Oracle VM VirtualBox

File   Machine   View   Input   Devices   Help

Terminal Terminal File Edit View Search Terminal Help

marina@marina-VirtualBox: ~

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

marina@marina-VirtualBox:~$
```

- Run this command (type it at the prompt and press Enter)

  sudo adduser $USER vboxsf

- When prompted for password, enter the password you created earlier for the VM.

- Then, reboot the guest Linux system by entering the following command:

  sudo reboot

  After reboot, the system should come back to the VM screen.

Appendix C: Labtainer Command Summary

The following labtainer commands are available from the `labtainer/labtainer-student` directory:

- `labtainer <lab> --` Start the named lab. If no name is given, a list of available labs will be displayed.

- `stoplab --` Stop the currently running lab.

- `moreterm.py <lab> <container> --` create a new virtual terminal for the container.

- `labtainer <lab> -r --` Delete any previous containers associated with this lab and start it fresh. **Warning**: this will lose any previous data from the named lab.

Most labs display lab instructions in one of the windows that appears after the lab starts. If those instructions stop displaying, e.g., because "q" is pressed in that window, then type the following in a virtual terminal (e.g., in a new terminal created using the moreterm.py script:

```
less instructions.txt
```