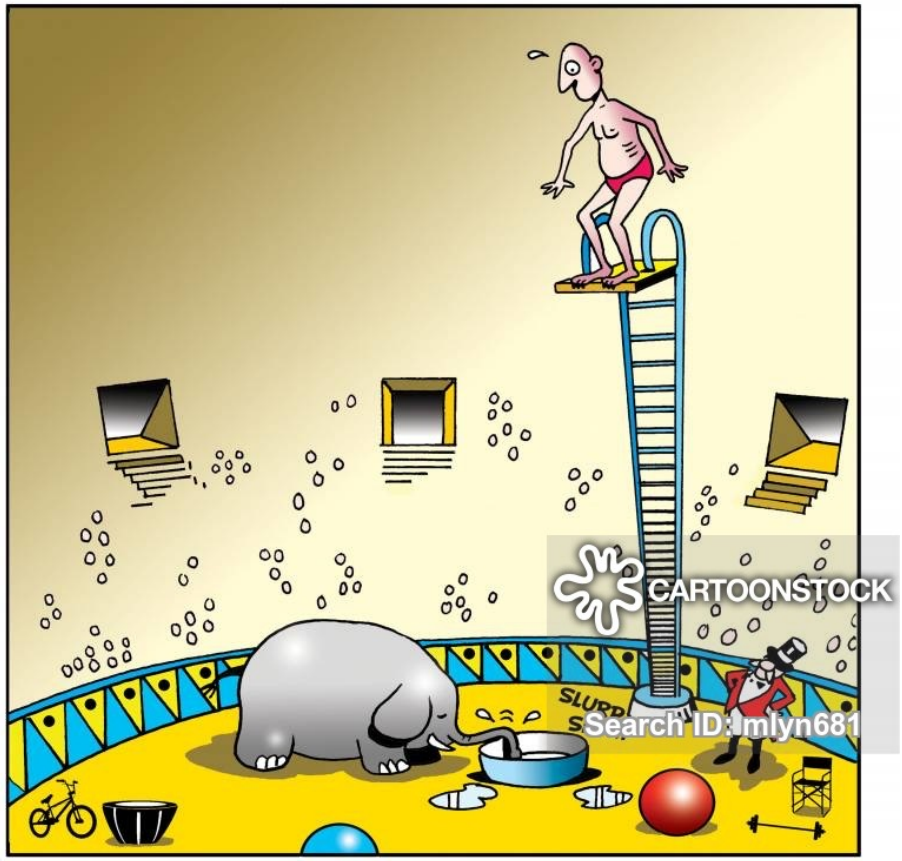
CSC 317

Wed Development

Setting Up – Diving in the deep end



- Install Virtual Box or Parallels*
- Import WebDev.ova
- Launch the Ubuntu image
- Install LAMP
- See it work!

* For Macintosh with the M1 or M2 chips

Take lots of Screen Shots

- This is part of your first assignment.

- Take a screen shot (and save it in a Word or Google document).
- Keep the screen shots large (no more than 2 per page) so that they are readable

- If you see this icon  on a slide, it is time to take a screen shot.

Step Zero – What Hardware are you on?

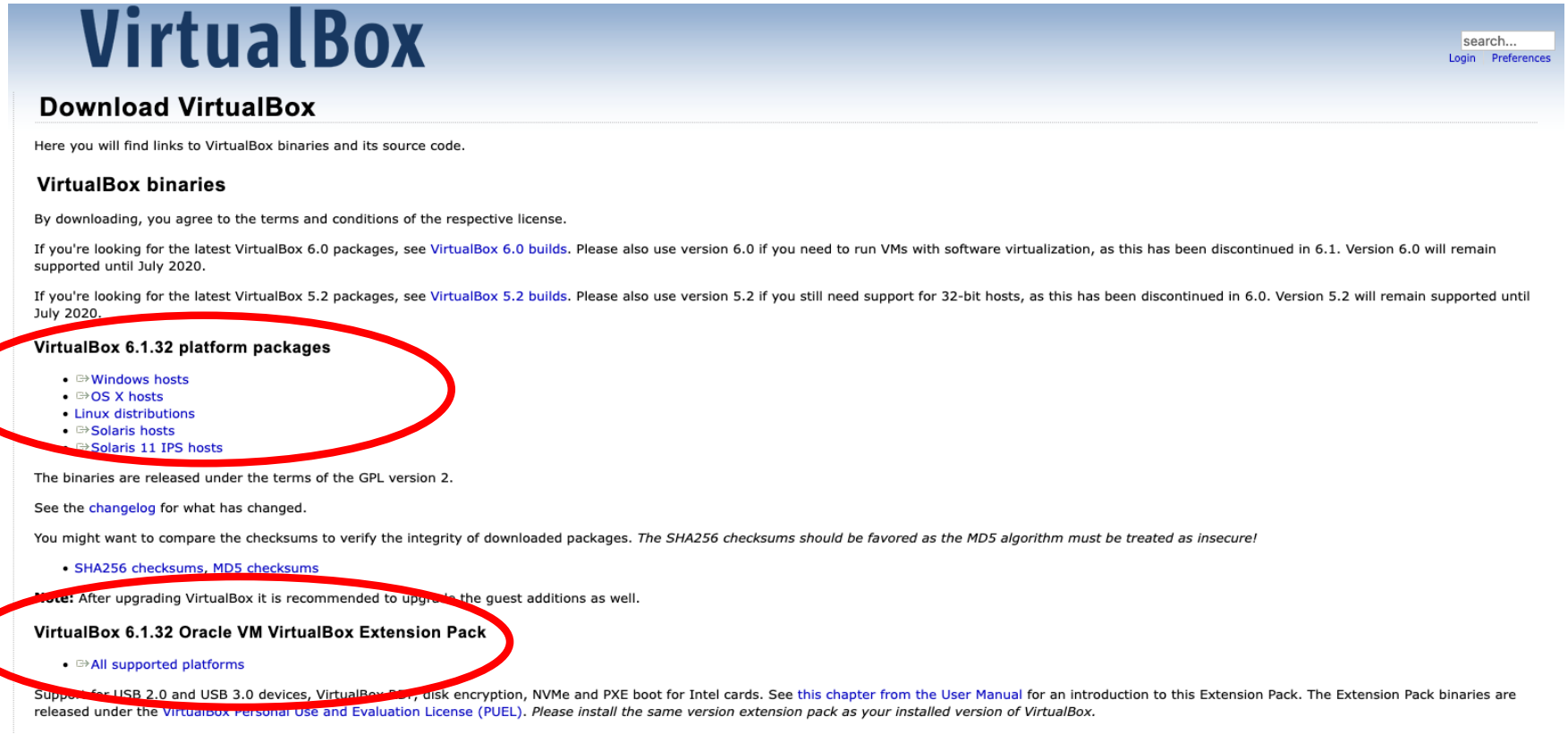
- Originally this deck was for all systems, Mac or Windows because both systems were Intel Architecture based. Recently Apple introduced their own silicon, the M1 and now the M2 chips which are ARM based architecture. Virtualbox does not run on an ARM based system.

- So, if you have Windows or an Intel based Macintosh (or Linux), proceed with the next slide and install Virtual Box and the provided OVA image.

- If, on the other hand, you have an M1 or M2 based Macintosh,
  - Skip to slide **16** and follow the instructions there.

First Step – Get and install Virtual Box

- https://www.virtualbox.org/wiki/Downloads



VirtualBox

search...
Login Preferences

Download VirtualBox

Here you will find links to VirtualBox binaries and its source code.

VirtualBox binaries

By downloading, you agree to the terms and conditions of the respective license.

If you're looking for the latest VirtualBox 6.0 packages, see VirtualBox 6.0 builds. Please also use version 6.0 if you need to run VMs with software virtualization, as this has been discontinued in 6.1. Version 6.0 will remain supported until July 2020.

If you're looking for the latest VirtualBox 5.2 packages, see VirtualBox 5.2 builds. Please also use version 5.2 if you still need support for 32-bit hosts, as this has been discontinued in 6.0. Version 5.2 will remain supported until July 2020.

VirtualBox 6.1.32 platform packages

- Windows hosts
- OS X hosts
- Linux distributions
- Solaris hosts
- Solaris 11 IPS hosts

The binaries are released under the terms of the GPL version 2.

See the changelog for what has changed.

You might want to compare the checksums to verify the integrity of downloaded packages. *The SHA256 checksums should be favored as the MD5 algorithm must be treated as insecure!*

- SHA256 checksums, MD5 checksums

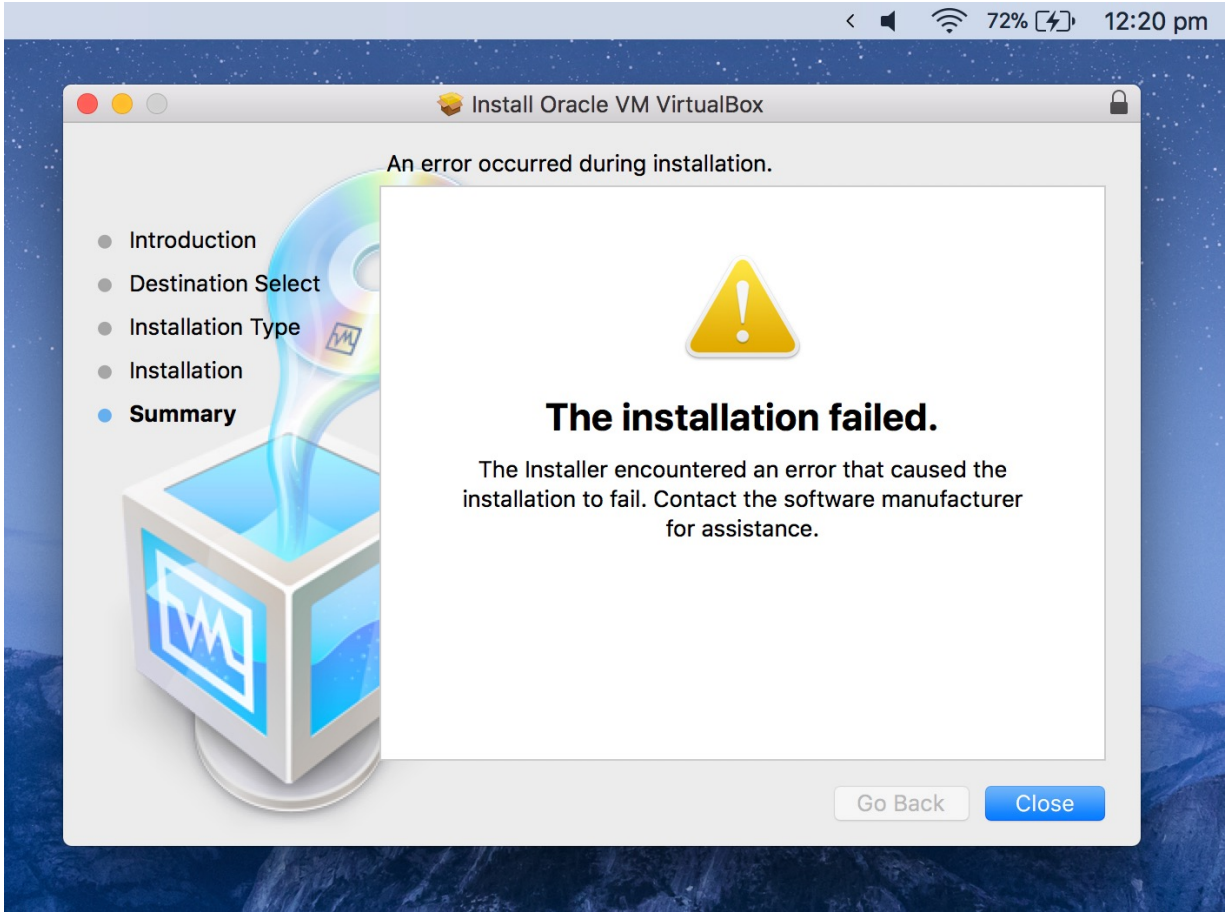
Note: After upgrading VirtualBox it is recommended to upgrade the guest additions as well.

VirtualBox 6.1.32 Oracle VM VirtualBox Extension Pack

- All supported platforms

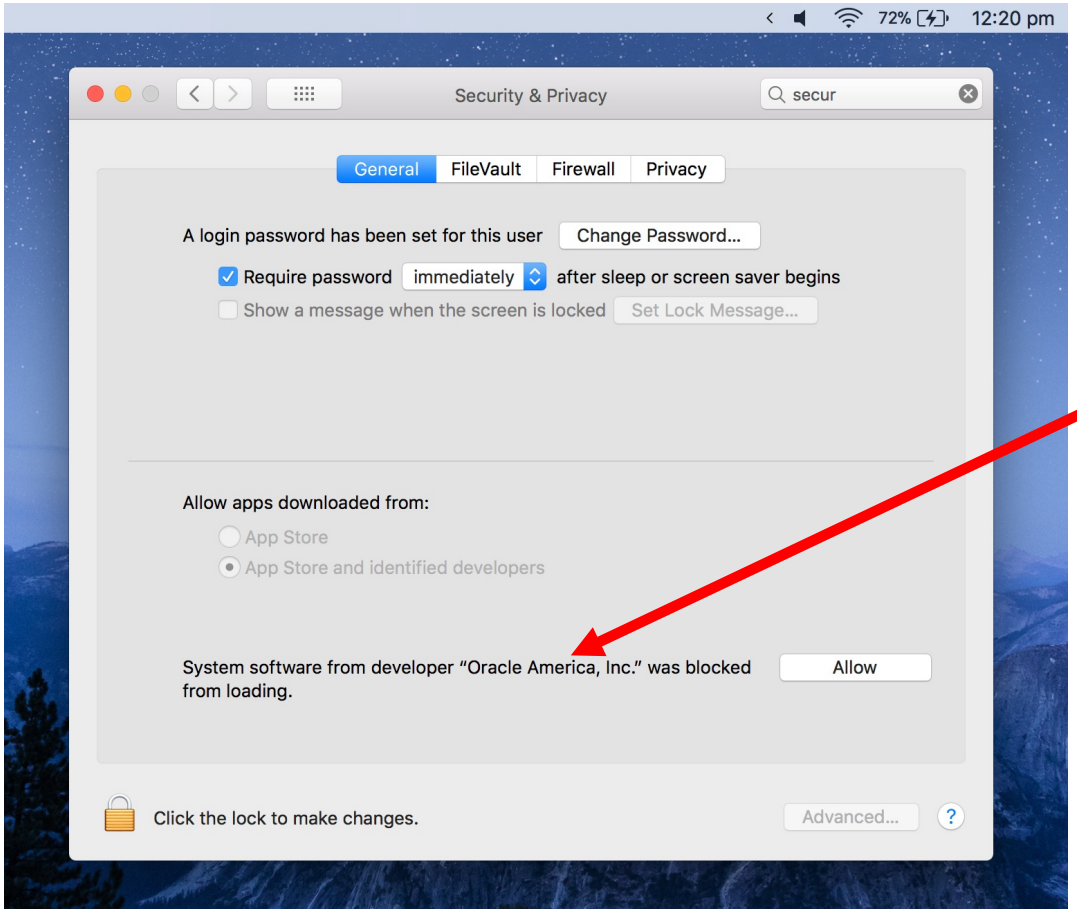
Support for USB 2.0 and USB 3.0 devices, VirtualBox RDP, disk encryption, NVMe and PXE boot for Intel cards. See this chapter from the User Manual for an introduction to this Extension Pack. The Extension Pack binaries are released under the VirtualBox Personal Use and Evaluation License (PUEL). *Please install the same version extension pack as your installed version of VirtualBox.*

Macintosh (Intel) Users – You may encounter this...



- This is a known issue and is easy to fix if you do it immediately!
- More on this issue can be found at: https://matthewpalmer.net/blog/2017/12/10/install-virtualbox-mac-high-sierra/index.html

Mac Installation Issue



- Go to the Security & Privacy panel in your system preferences.
- You should see the message here
- Click the Allow button

- You only have **about 30 minutes from the install fail to do this** or it won't appear again without more work

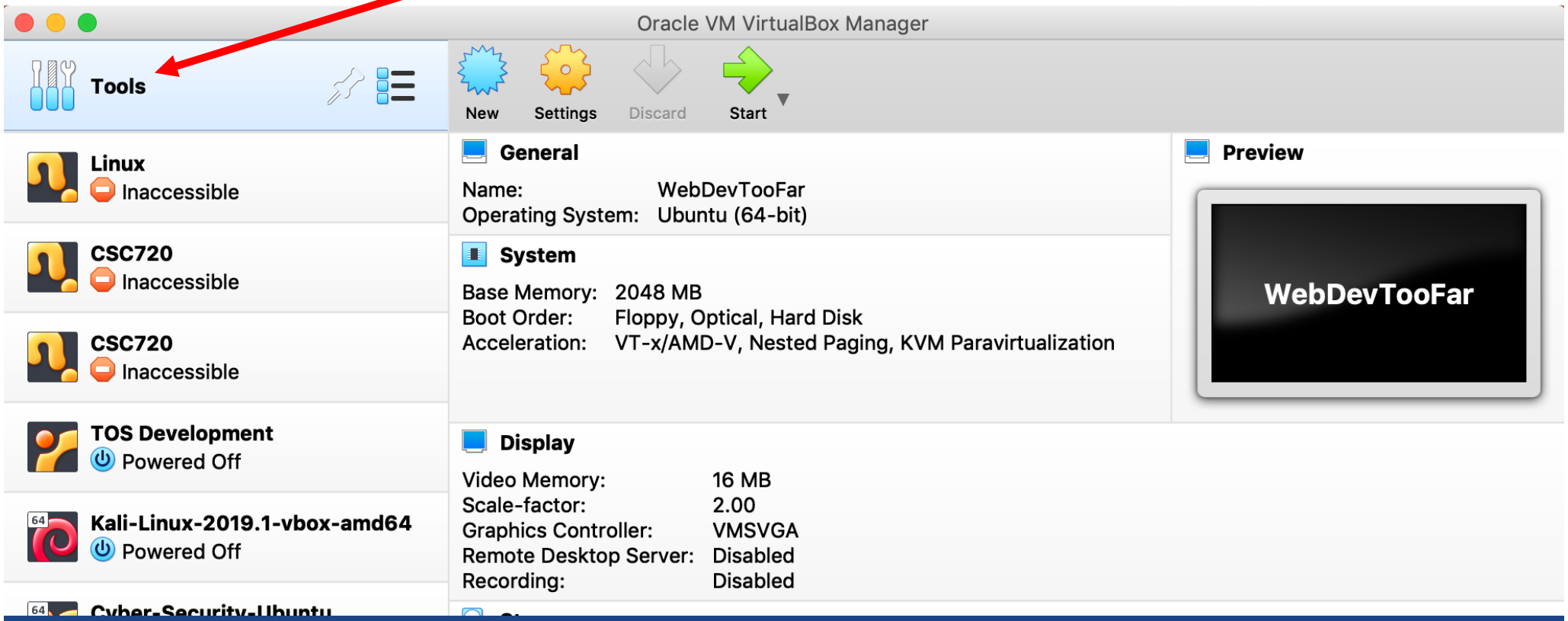
Once you have Virtual Box Installed…

- Do NOT forget to now download and install the **Oracle VM VirtualBox Extension Pack**
- That must be installed next.

- Now you need the OVA file. This is the large 4.2GB file that is on the iLearn site.
- You can also download it here: https://sfsu.box.com/v/csc317webdevova
- It does not have a preview, just click the "Download" button. Depending on your internet speed this can take between 2 and 45 minutes.
- Click "Save" and not open if your browser asks.

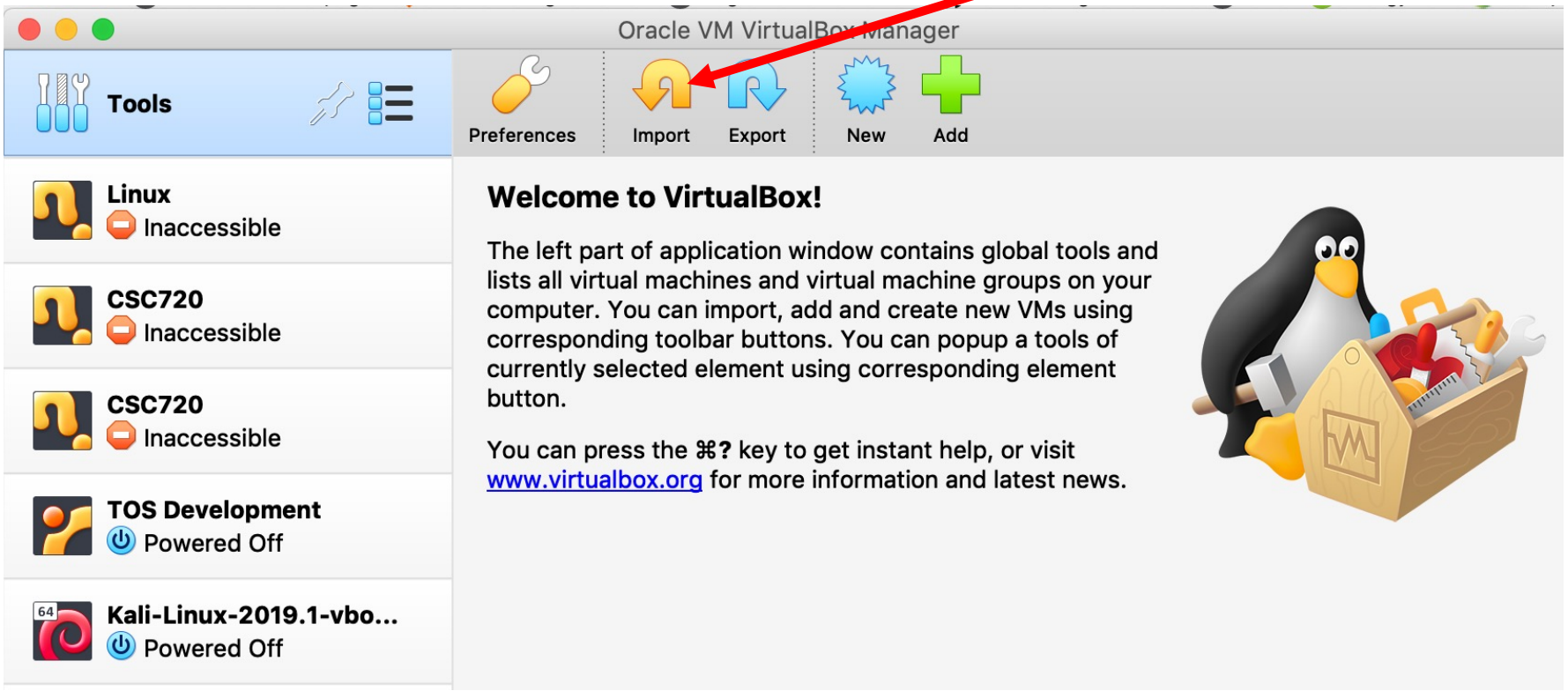
Launch Virtual Box

- The click the Tools Icon

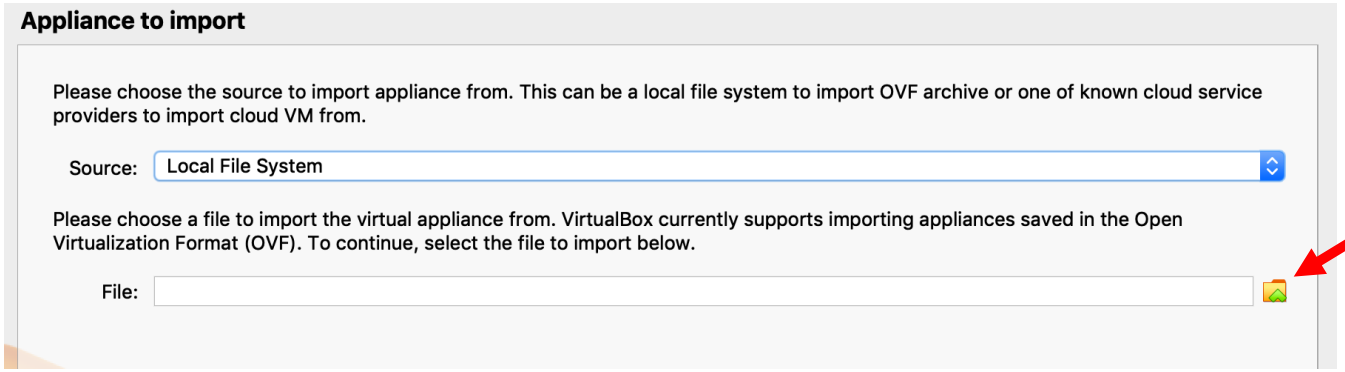


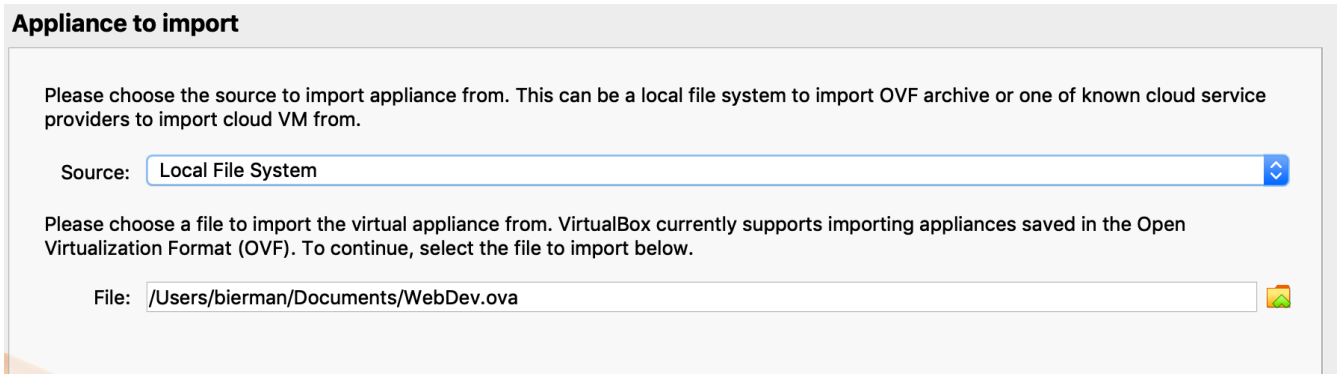
We are IMPORTING an image

- We are not doing an add, but rather an import, so select import



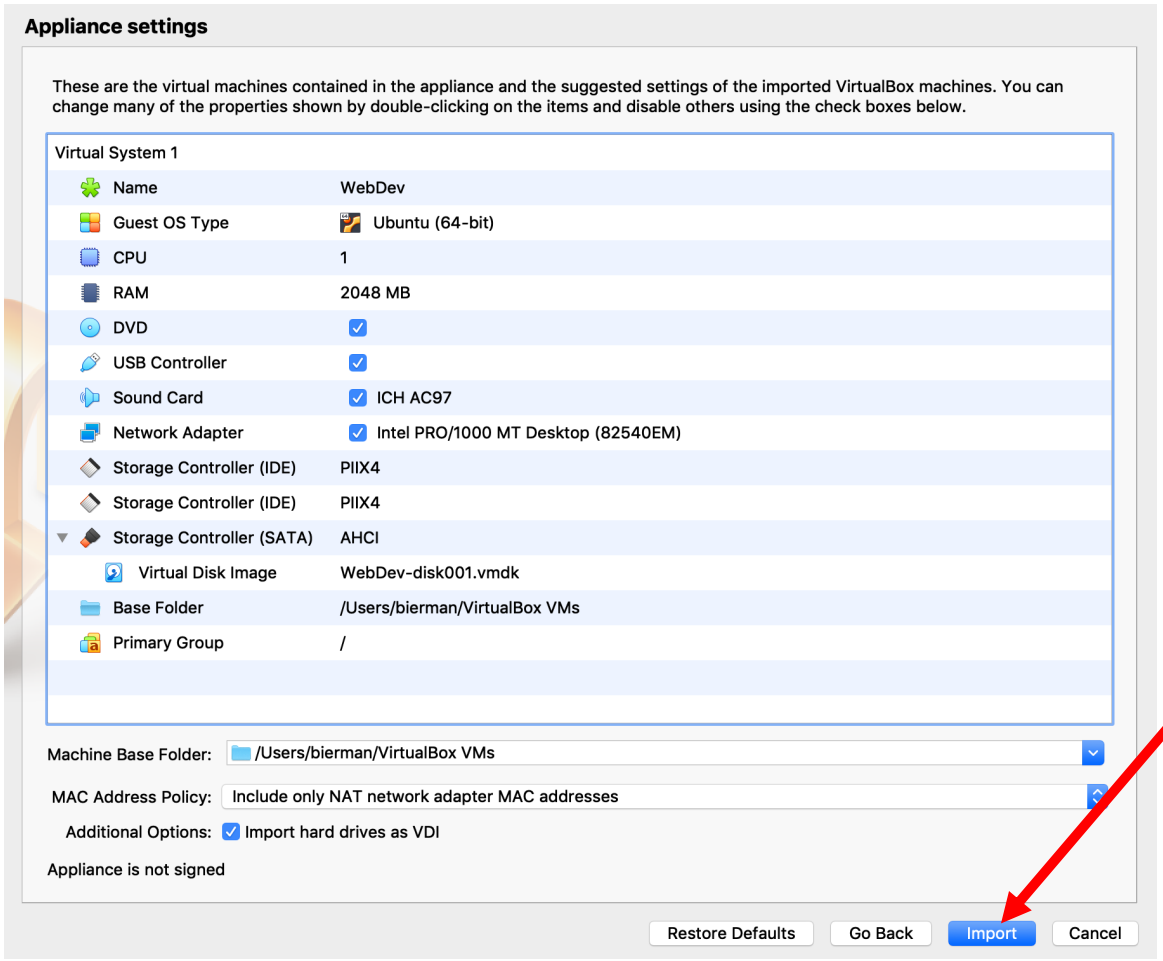
Now it will ask for the file to import





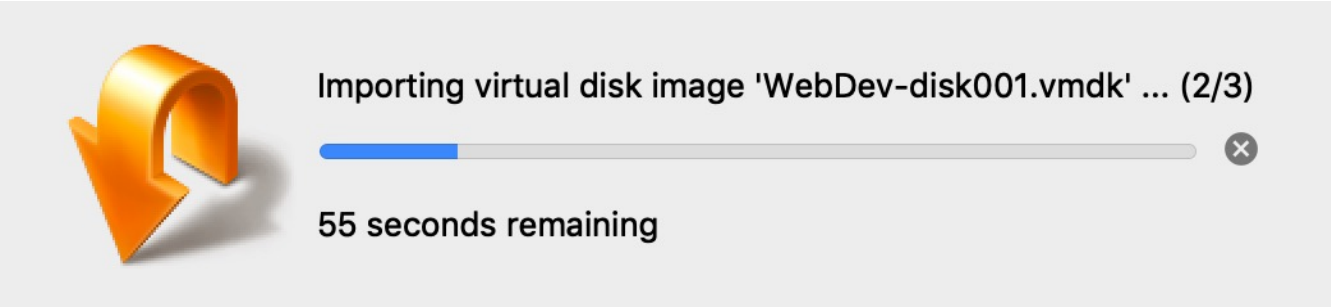
- Click the folder icon
- Then choose the WebDev.ova file you downloaded
- See iLearn on how to get WebDev.ova

Info on the image

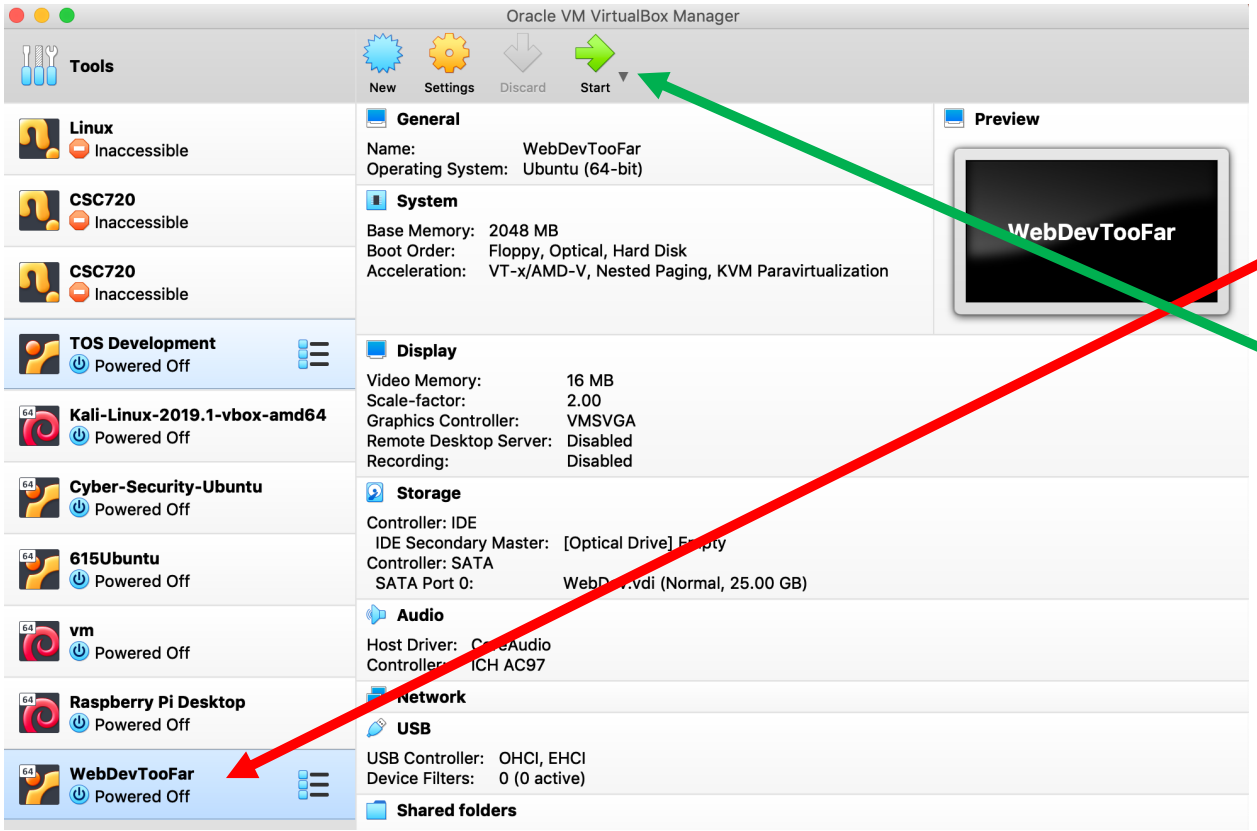


- The Defaults here are fine
- Just click the import button

Now Wait...


Importing virtual disk image 'WebDev-disk001.vmdk' ... (2/3)
55 seconds remaining

Now select WebDev and Start



- Select the WebDev "machine"
- Then click Start

Jump to Slide 18

Installing Parallels for Macintosh M1 and M2 Systems

- This section is only for those with an ARM architecture Macintosh system.
- **If you are on a M1 or M2 Macintosh - you must install Parallels then install Linux**
- You will have to purchase and install Parallels (They offer a 50% discount for students, for $39.99, go to https://www.parallels.com/pd/general/ (Links to an external site.) then click the FAQ, and under "Do you have educational pricing for students or faculty?" open that and click the "verify their eligibility" link).

After Purchasing Parallels

- Install Parallels

- Instead of my video for installing Virtual Box, use this YouTube video for installing Ubuntu on Parallels:
  - https://www.youtube.com/watch?v=EiO_CHfSn2s&t=13s
  - Use the password “student” and then change name to student where he shows you how to change your name.

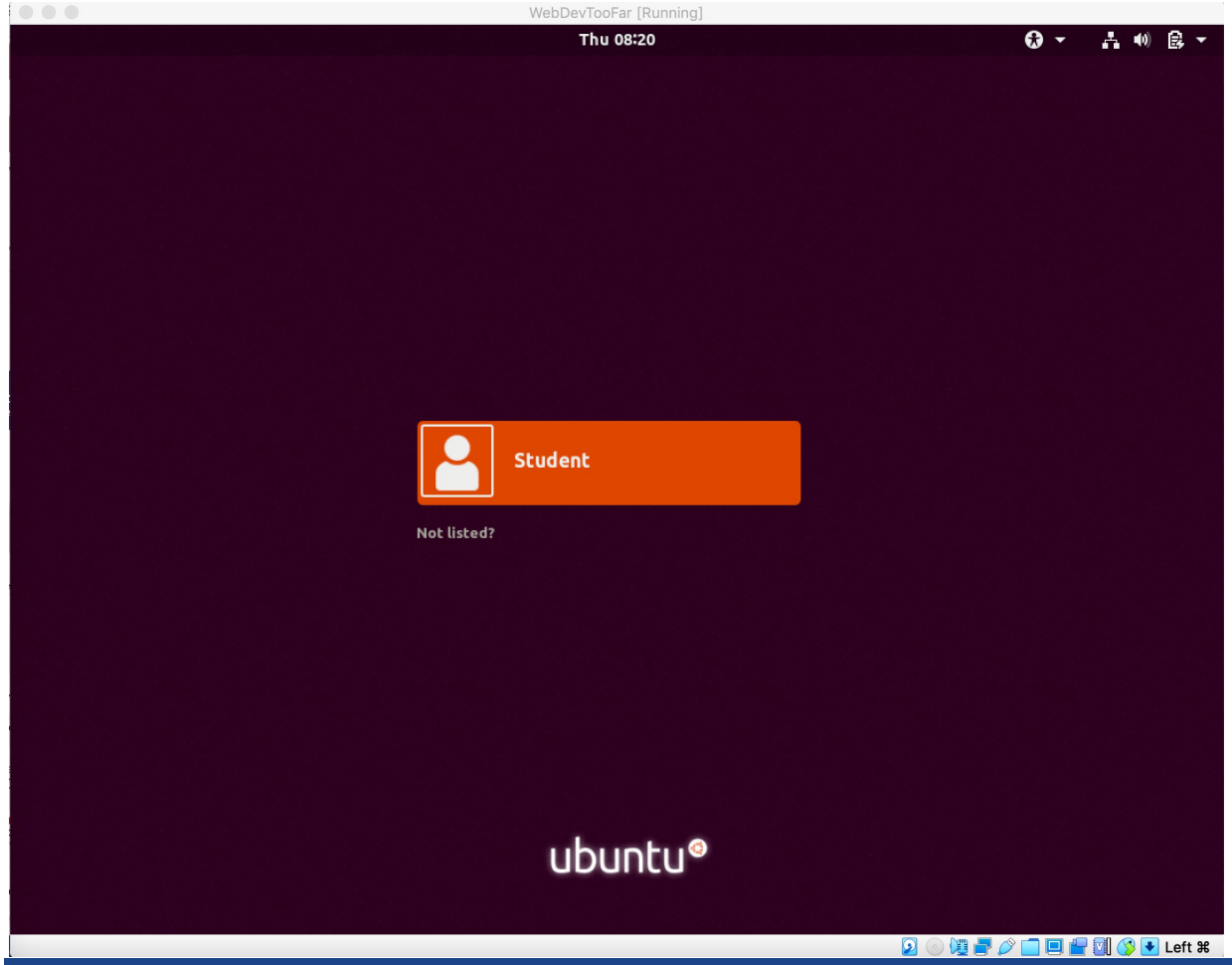
- Don’t forget to take a screen shot of your installed Ubuntu

- Now continue with the rest of the class…

Linux Installed

Lamp

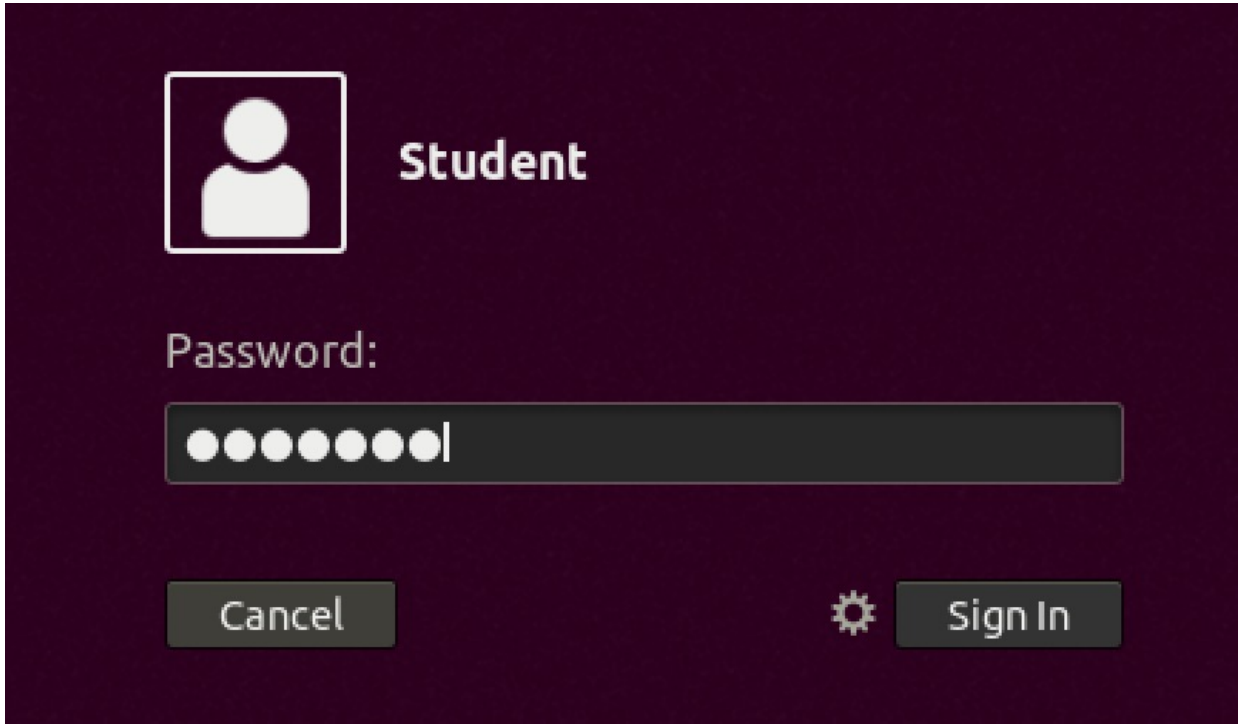
Ubuntu is now running!!



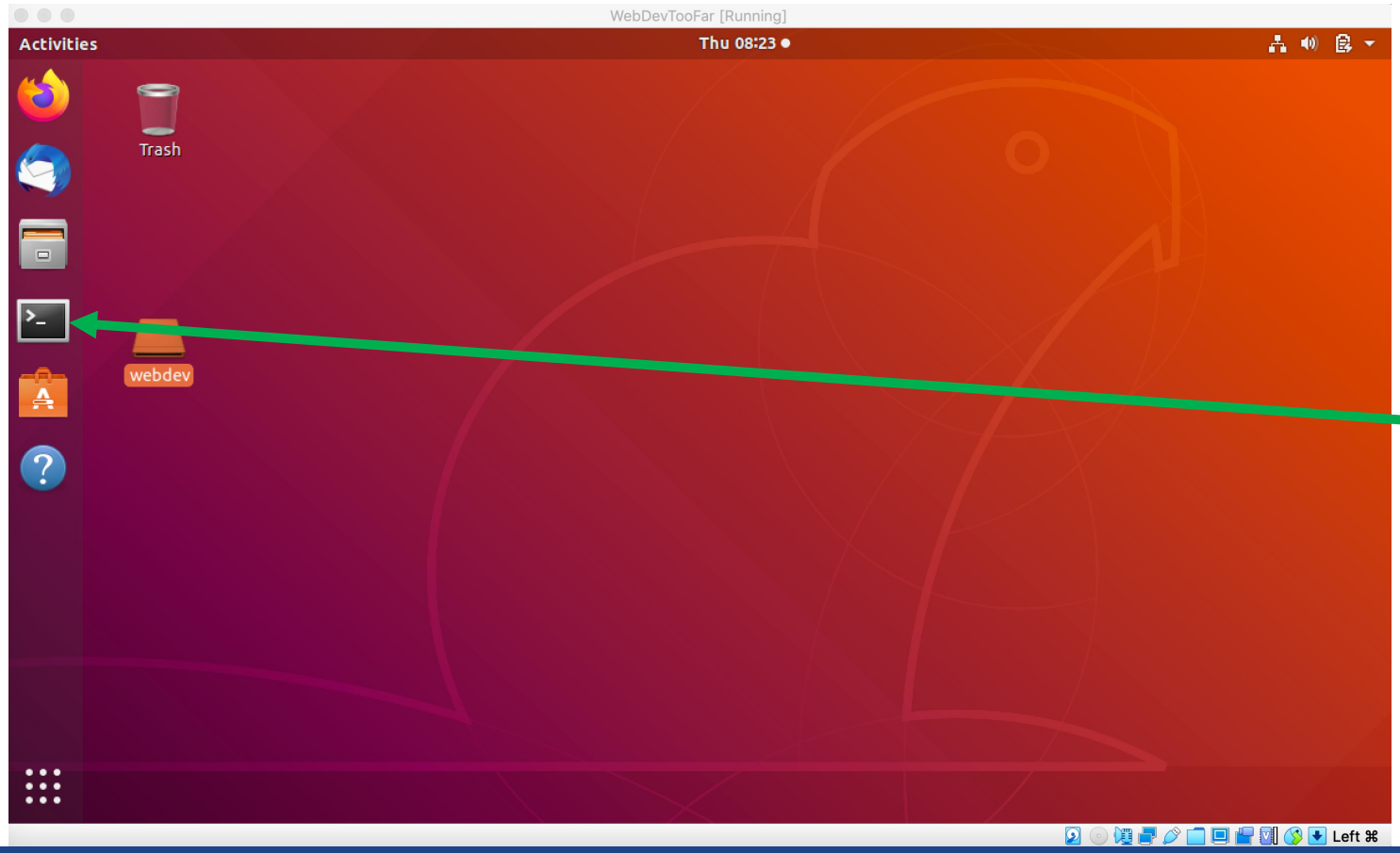
- Select student (this is your username)

The password is…

- … student

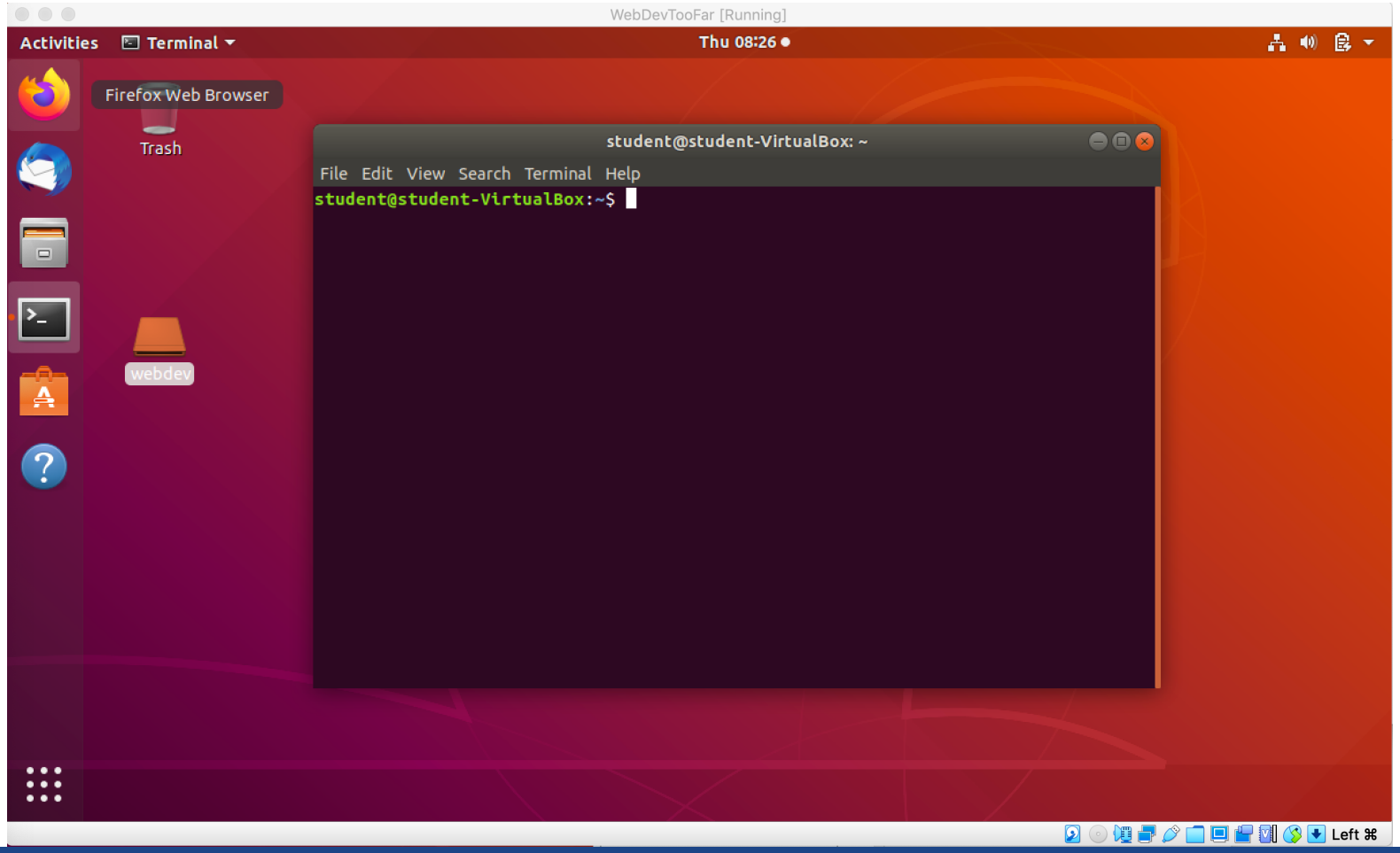


We are up



- We are up and running
- Click the Terminal Icon

Welcome to the Linux command line interface



Some command we will see today

- `sudo` – superuser do – allows normal users (with sudo privileges) to do administrative tasks
- `ls` – lists the files in a directory (we will use `ls -la` a lot as well)
- `cd` – change directory
- `mkdir` – make directory
- `apt` – a tool to help install programs

More commands

- `curl` – a way to retrieve the actual HTML of a URL – no rendering is done
- `apache2ctl` – part of the apache web services (control program)
- `nano` – a simple editor
- `echo` – plays back what is on the line to stdout
- `ufw` – uncomplicated fire wall
- `rm` – remove a file (or directory)

Let's start to install our software

- The `apt` program will be how we install the next set of components we want, but first we need to make sure the `apt` database is up-to-date
  - `sudo apt update`

- Note, when we use `sudo`, we may be prompted to enter our password

- Next, we will install and use `curl`
  - `sudo apt install curl`
  - `curl sfsu.edu`
    - You should see the following: (next screen)

Curl at work

- This shows us that `curl` worked and that sfsu.edu now is an https site.

```
student@student-VirtualBox: ~
File  Edit  View  Search  Terminal  Help
student@student-VirtualBox:~$
student@student-VirtualBox:~$ curl sfsu.edu
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>301 Moved Permanently</title>
</head><body>
<h1>Moved Permanently</h1>
<p>The document has moved <a href="https://sfsu.edu/">here</a>.</p>
</body></html>
student@student-VirtualBox:~$
```

Time for Apache

lAmp

Install Apache

- We are now going to start the Apache installation
  - `sudo apt install apache2`

- Now let's use curl to get our IP address https://www.showmyip.com/
  - `curl https://www.showmyip.com/`
    - Now scroll back up to where the command started and then look down a few lines and you will find your IP address. Make a note of it.

Configuration

- So now we need to edit some files

- `sudo nano /etc/apache2/mods-enabled/dir.conf`
  - We are going to enter at the bottom of the file (replace IP as needed):
    - `ServerName 130.212.92.59`
  - Then to save and exit we to the following:
    - `Ctrl-X` (for exit)
    - `Y` (for yes we want to save our changes)
    - `Return` (to write it to the file and exit)

Verify the configuration and start Apache

- # Verify Configuration
  - `sudo apache2ctl configtest`
- # Restart Apache
  - `sudo service apache2 restart`
- # Ensure the server starts on boot
  - `sudo update-rc.d apache2 defaults`

Setting up the firewall

- Setting up Uncomplicated Firewall (ufw)

- First, show the apps that ufw understands
  - `sudo ufw app list`
- You should see Apache Full as one of those choices, that is what we want
  - `sudo ufw allow 'Apache Full'`

- Check the status of ufw
  - `sudo ufw status`
- Enable ufw
  - `sudo ufw enable`

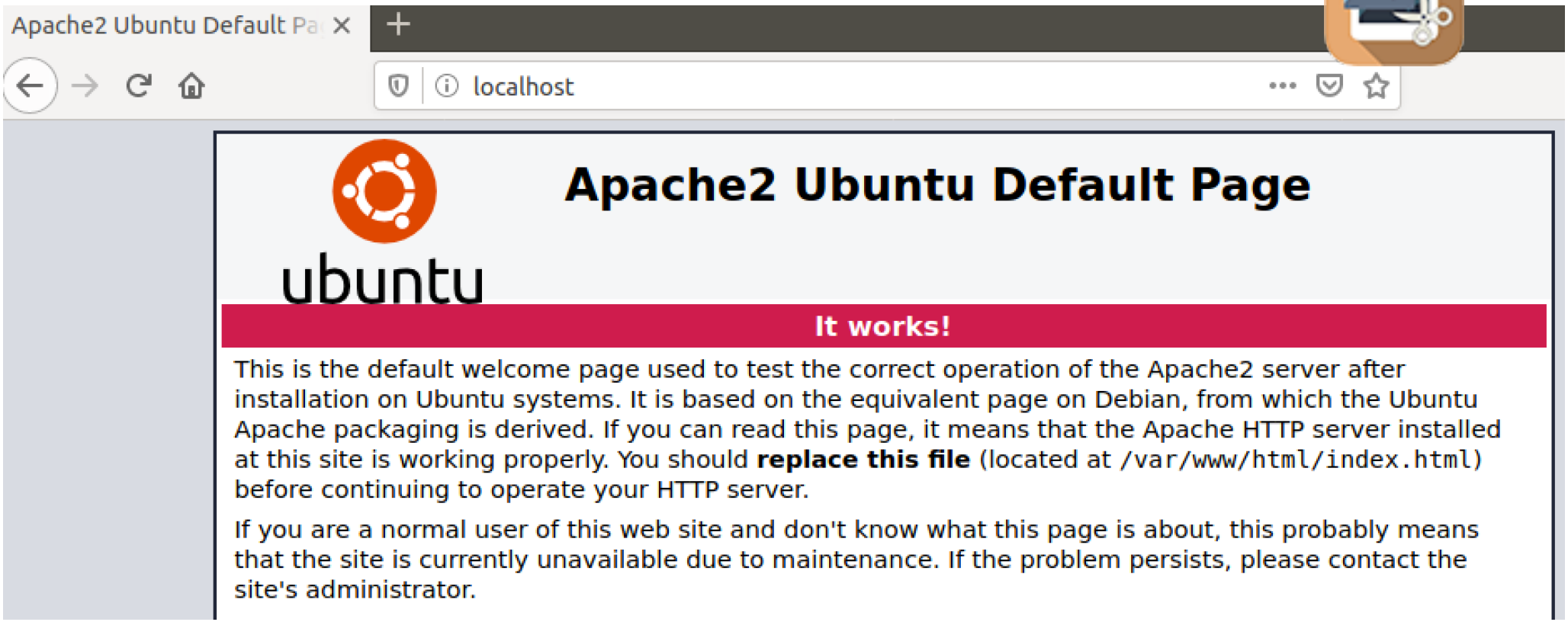
Let's see if that worked...

- We can now check our work
  - `curl localhost`

- localhost is the "name" for our local computer (in this case the ubuntu image)

- You should see a bunch of HTML displayed
- Now open Firefox from the Ubuntu GUI – you should see the image on the next slide

It Works!


Apache2 Ubuntu Default Pa
localhost
Apache2 Ubuntu Default Page
ubuntu
It works!
This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should replace this file (located at /var/www/html/index.html) before continuing to operate your HTTP server.
If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Now for PHP

lamP

Now let's install PHP

- First run the install
  - `sudo apt-get install php libapache2-mod-php php-mysql`

- Now we need to dit the DirectoryIndex line so `index.php` is before `index.html` in this configuration file
  - `sudo nano /etc/apache2/mods-enabled/dir.conf`

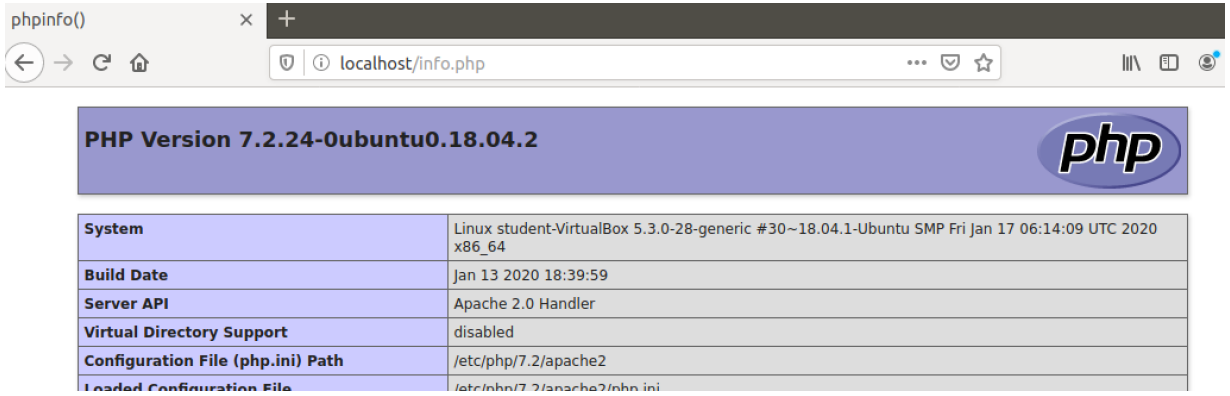
- Modify this line so that index.php is before index.html as shown

```
DirectoryIndex index.php index.html index.cgi index.pl index.xhtml ind
```

Testing PHP

- Create a test file with echo – we are creating the file info.php in the "website" directory structure with a php function call. A built-in function that returns information on your server and configuration
  - `sudo chmod 777 /var/www/html`
  - `sudo echo '<?php phpinfo(); ?>' > /var/www/html/info.php`

- Now navigate to: `http://localhost/info.php` in your Ubuntu browser



Remove that info.php file

- We now want to remove that info.php file. This file displays way too much information about your server and should only be used for debugging your site.
  - `sudo rm /var/www/html/info.php`

Finally MySQL

laMp

Database time

- First install mysql
  - `sudo apt install mysql-server`

- Then run through configuration steps...
  - Make sure to **document** your credentials!
    - `sudo mysql_secure_installation`

Configuration Questions

```
Securing the MySQL server deployment.

Connecting to MySQL using a blank password.

VALIDATE PASSWORD COMPONENT can be used to test passwords
and improve security. It checks the strength of password
and allows the users to set only those passwords which are
secure enough. Would you like to setup VALIDATE PASSWORD component?

Press y|Y for Yes, any other key for No:
```

- Choose N (low strength) then enter a password – this can be anything you want but you MUST REMEMBER IT!
  - I use **`student`** for this image since it is not production

- It will then ask you to reenter the password and if you want to use that password (renter it)

Configuration Questions

- Answers to the rest of the configuration questions
  - Remove anonymous users? Y
  - Disallow root login remotely? Y
  - Remove test database and access to it? N
  - Reload privilege tables now? Y

Congratulations
You now have a Webserver