

Up-Cycle Makerspace Startup Instructions

Prerequisites

This project uses some tools that are unique to Linux. For developing, we ran everything in Ubuntu 16.04 LTS running in a VirtualBox VM. You will need VirtualBox installed to work on this project. You will also need to enable virtualization on your laptops. You can have them do this at the IT helpdesk.

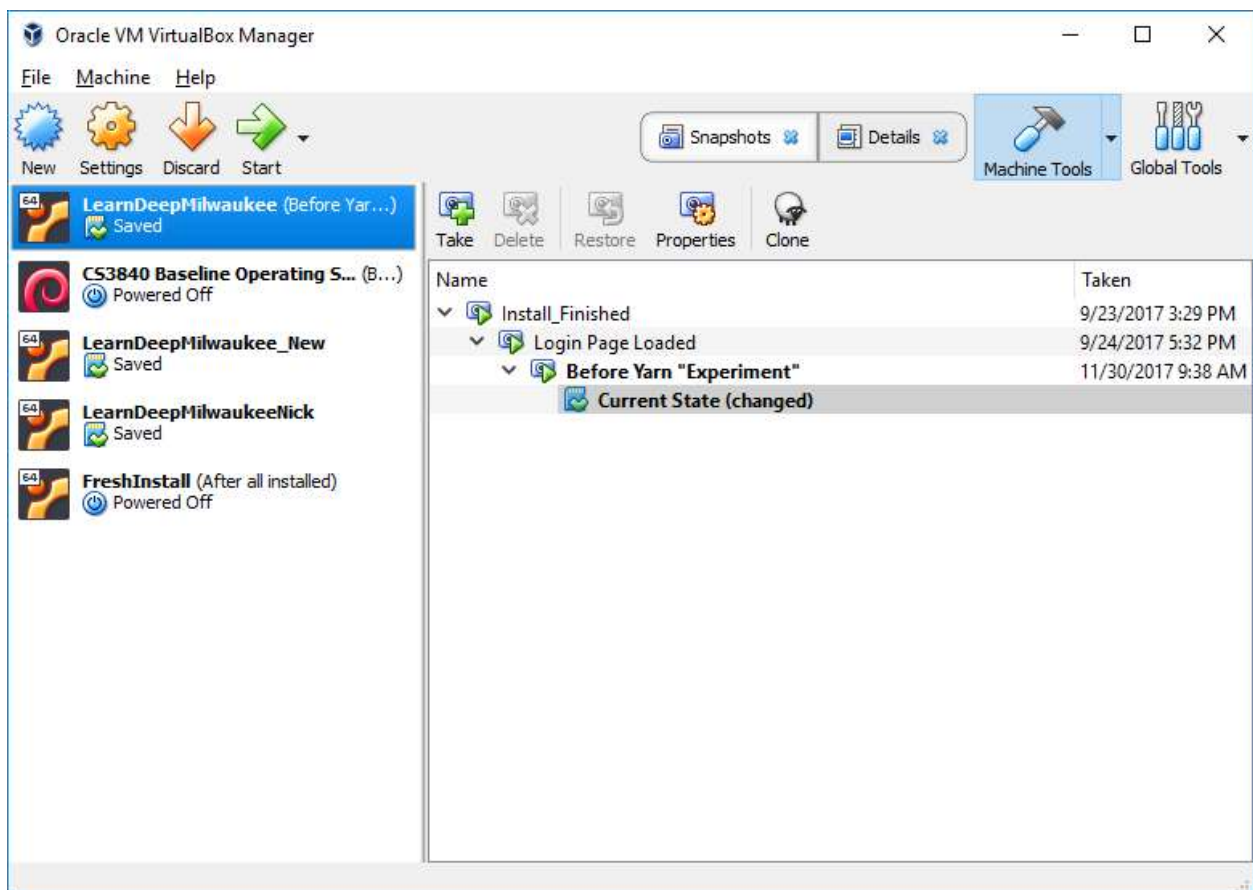
Download Ubuntu 16.04 LTS ISO from Ubuntu's site

Or just follow this link for the correct download <http://releases.ubuntu.com/16.04/ubuntu-16.04.4-desktop-amd64.iso>

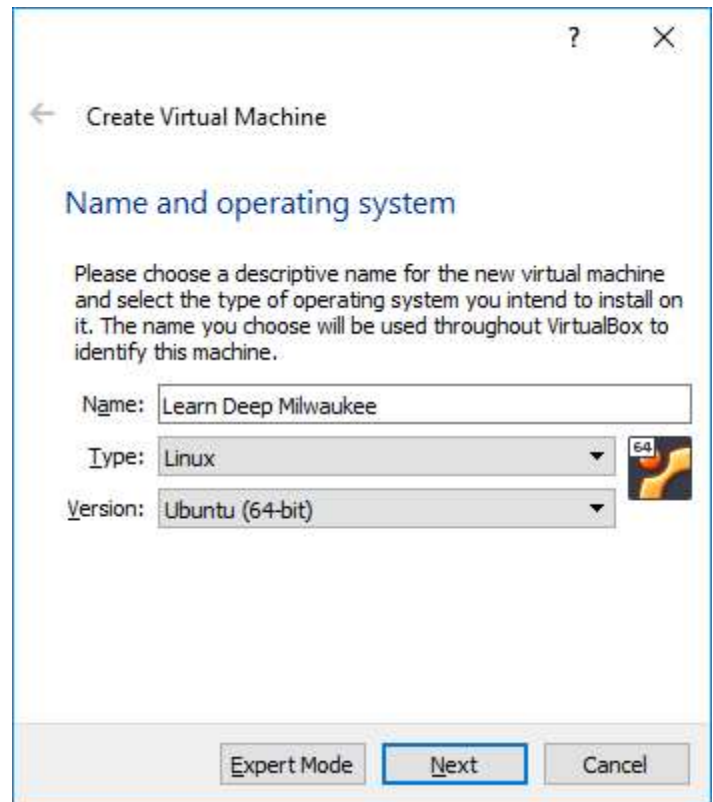
You might want to keep this download handy in case you ever need to restart the VM, but you shouldn't need the ISO once you install the OS onto your VM.

Setting up the VM

Startup VirtualBox. You likely won't have any VMs created yet.



Click on New to start making the VM. Give it a name, and set it to use Ubuntu (64-bit). (right)



The screenshot shows the 'Create Virtual Machine' window with the 'Name and operating system' tab selected. The window has a title bar with a question mark and a close button. Below the title bar is a back arrow and the text 'Create Virtual Machine'. The main heading is 'Name and operating system'. Below this is a paragraph: '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.' There are three input fields: 'Name' with the text 'Learn Deep Milwaukee', 'Type' with a dropdown menu showing 'Linux', and 'Version' with a dropdown menu showing 'Ubuntu (64-bit)'. To the right of the 'Type' and 'Version' dropdowns is a small icon of a person. At the bottom of the window are three buttons: 'Expert Mode', 'Next' (which is highlighted with a blue border), and 'Cancel'.

Create Virtual Machine

Name and operating system

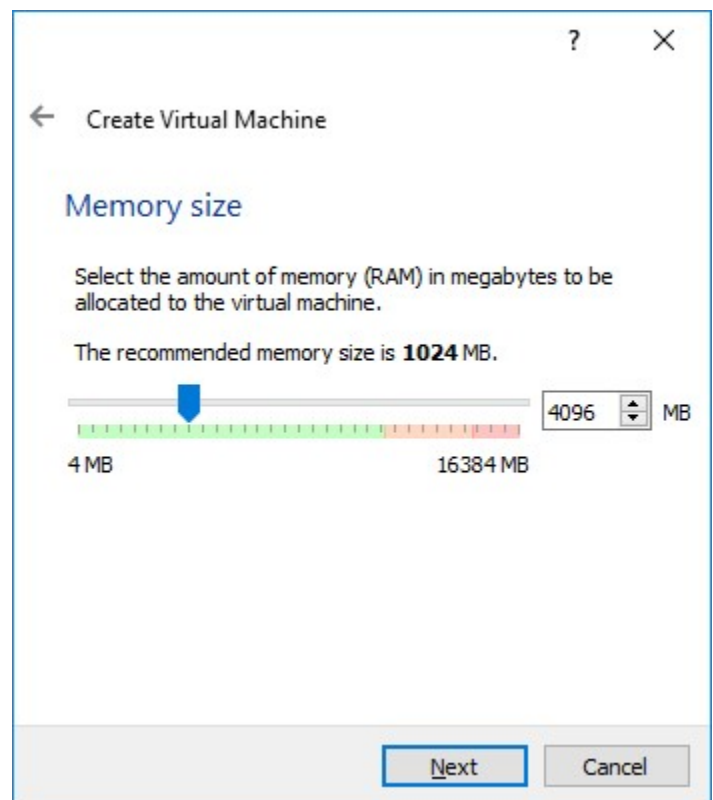
Please choose a descriptive name for the new virtual machine and select the type of operating system you intend to install on it. The name you choose will be used throughout VirtualBox to identify this machine.

Name:

Type:

Version:

Allocate memory for the VM. It will not use all the memory right away, this is an upper limit. Although it recommends 1024 MB, I would suggest giving it 4 GB (4096 MB) because you will be running some heavy weight programs. (below)



The screenshot shows the 'Create Virtual Machine' window with the 'Memory size' tab selected. The window has a title bar with a question mark and a close button. Below the title bar is a back arrow and the text 'Create Virtual Machine'. The main heading is 'Memory size'. Below this is a paragraph: 'Select the amount of memory (RAM) in megabytes to be allocated to the virtual machine.' Below that is another paragraph: 'The recommended memory size is 1024 MB.' There is a slider bar with a blue arrow pointing to the right. The slider has a green section on the left and a red section on the right. Below the slider are the labels '4 MB' and '16384 MB'. To the right of the slider is a text box containing '4096' and a unit 'MB'. At the bottom of the window are two buttons: 'Next' (highlighted with a blue border) and 'Cancel'.

Create Virtual Machine

Memory size

Select the amount of memory (RAM) in megabytes to be allocated to the virtual machine.

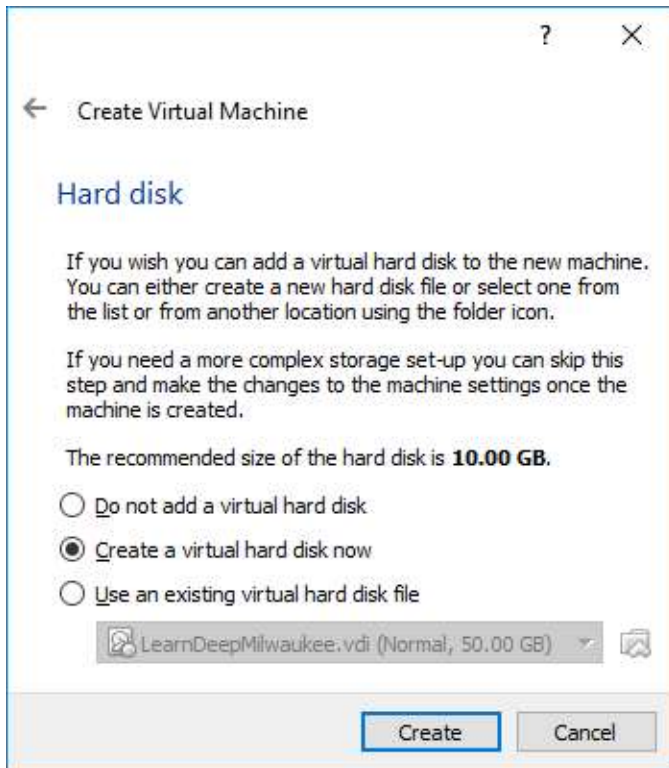
The recommended memory size is **1024 MB**.

MB

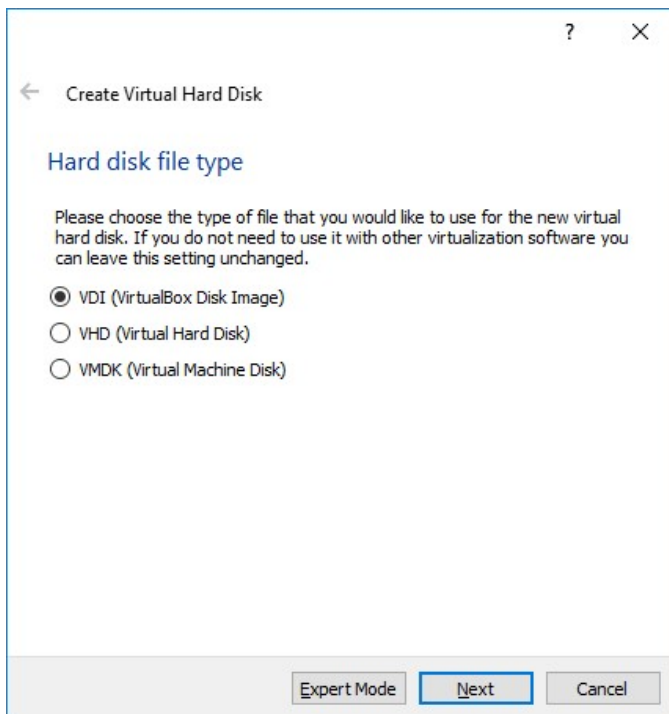
4 MB 16384 MB

Setting up the Hard Disk

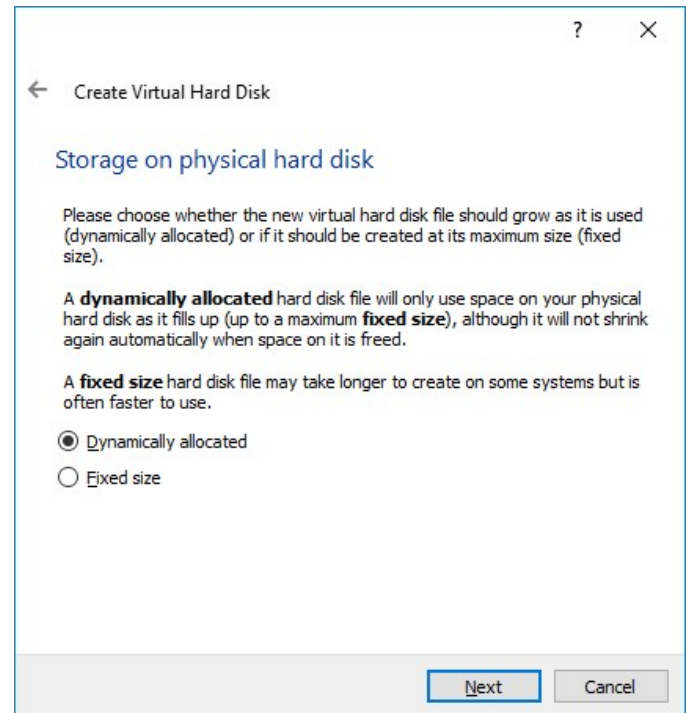
1. Create a virtual hard disk now



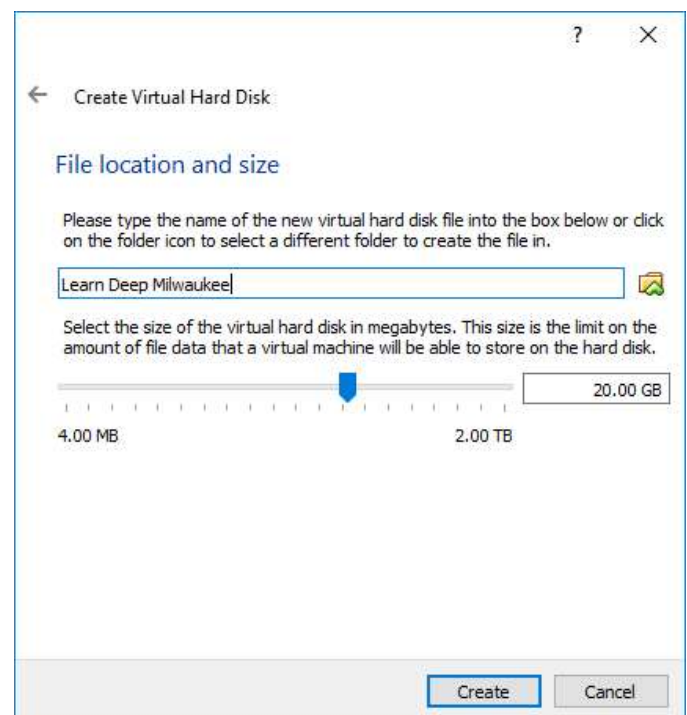
2. VDI (VirtualBox Disk Image)



3. Set it up to dynamically allocated new space



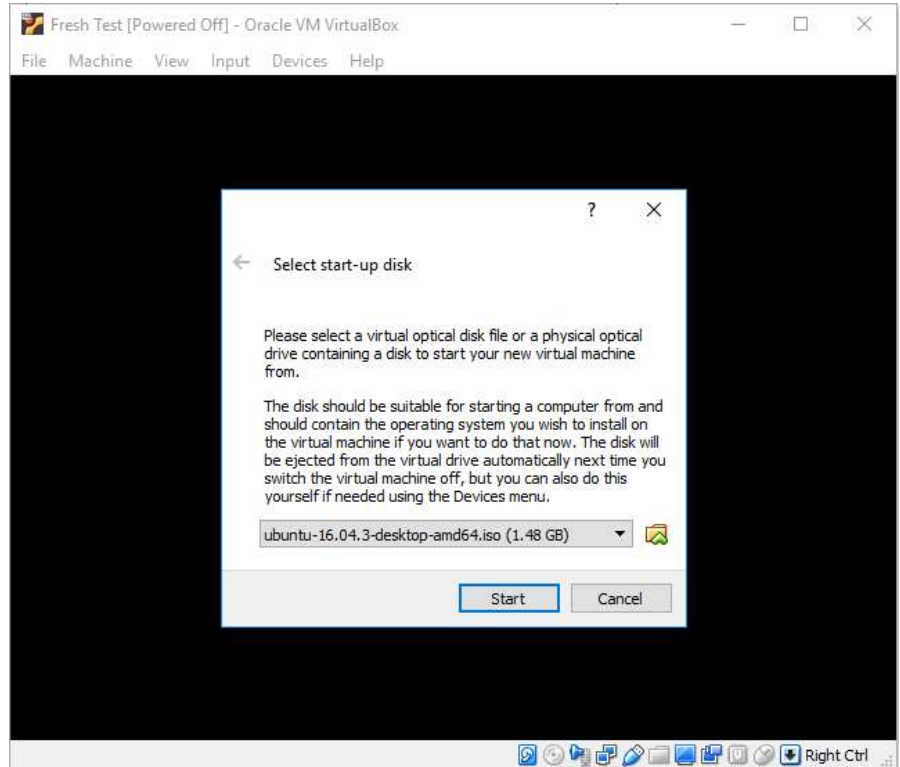
4. Give it 20 GB. It is dynamically allocated, so it will only take space as it is needed. The 20 GB is an upper limit. After some use, we ran out of space with only 10 GB.



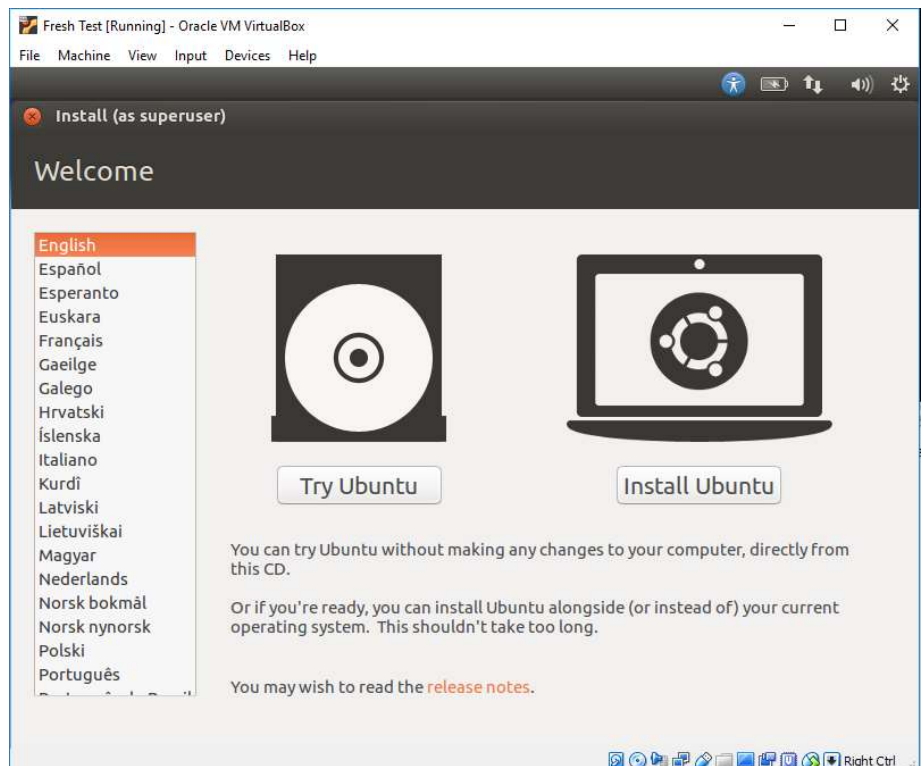
Installing Ubuntu

At this point, the setup wizard should close, and you should see the VM you just created in the main window. Select it, and click “Start”.

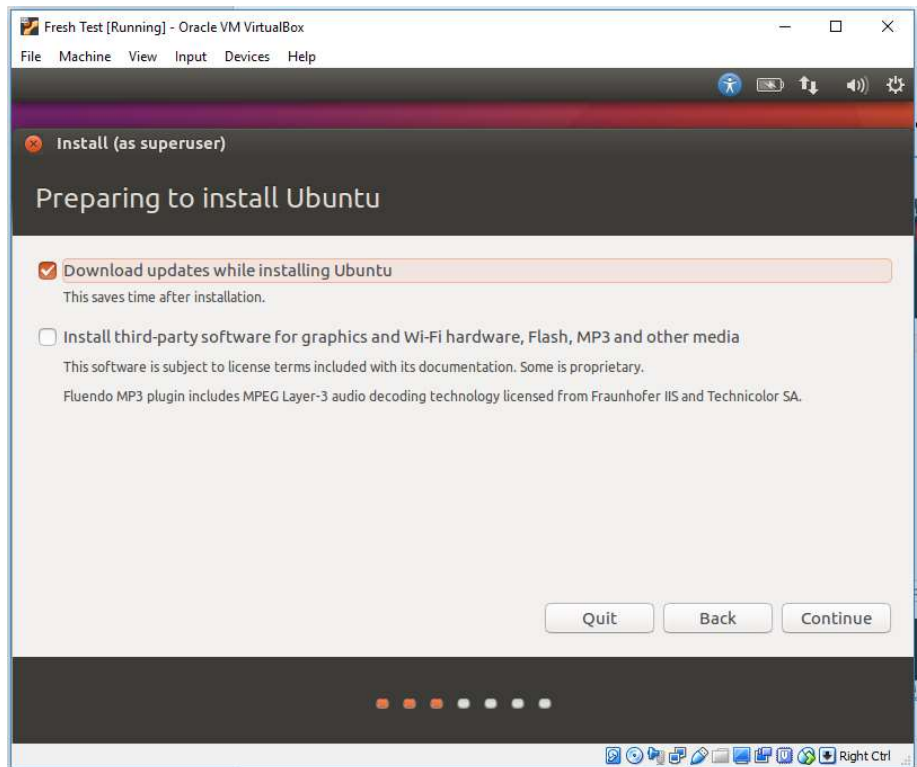
It will ask you for the operating system to use. You will need to browse to where you saved the Ubuntu 16.04 LTS ISO file earlier and select it. Then click Start.



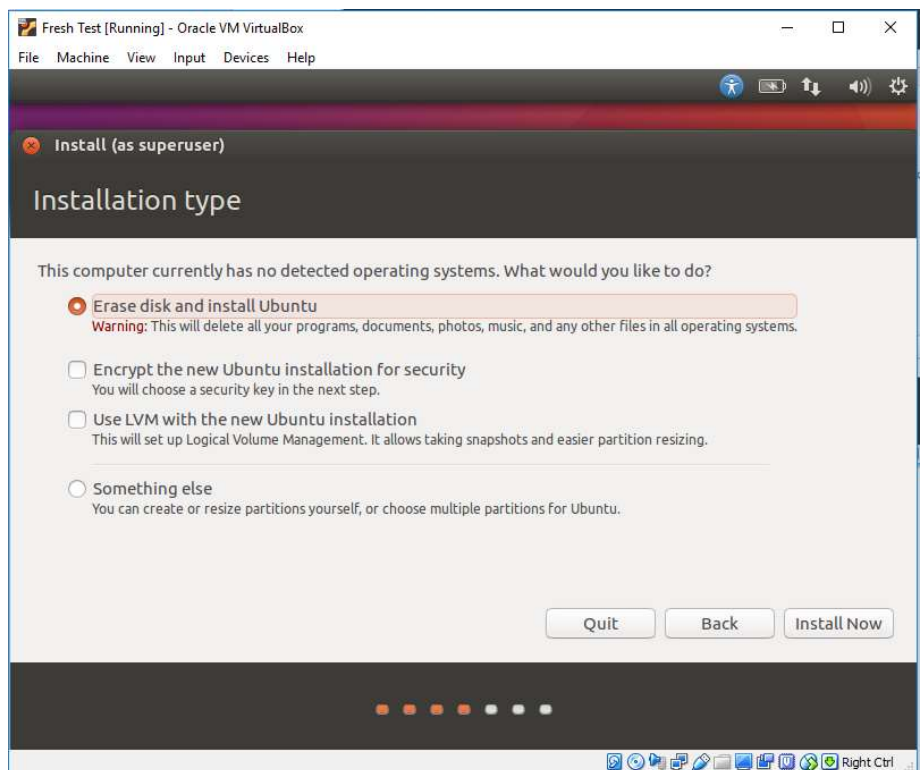
You will now install and set up the operating system. Click “Install Ubuntu”



Select Download updates while installing Ubuntu. Don't worry about the third-party software. Everything additional you will need is already taken care of in our setup script.

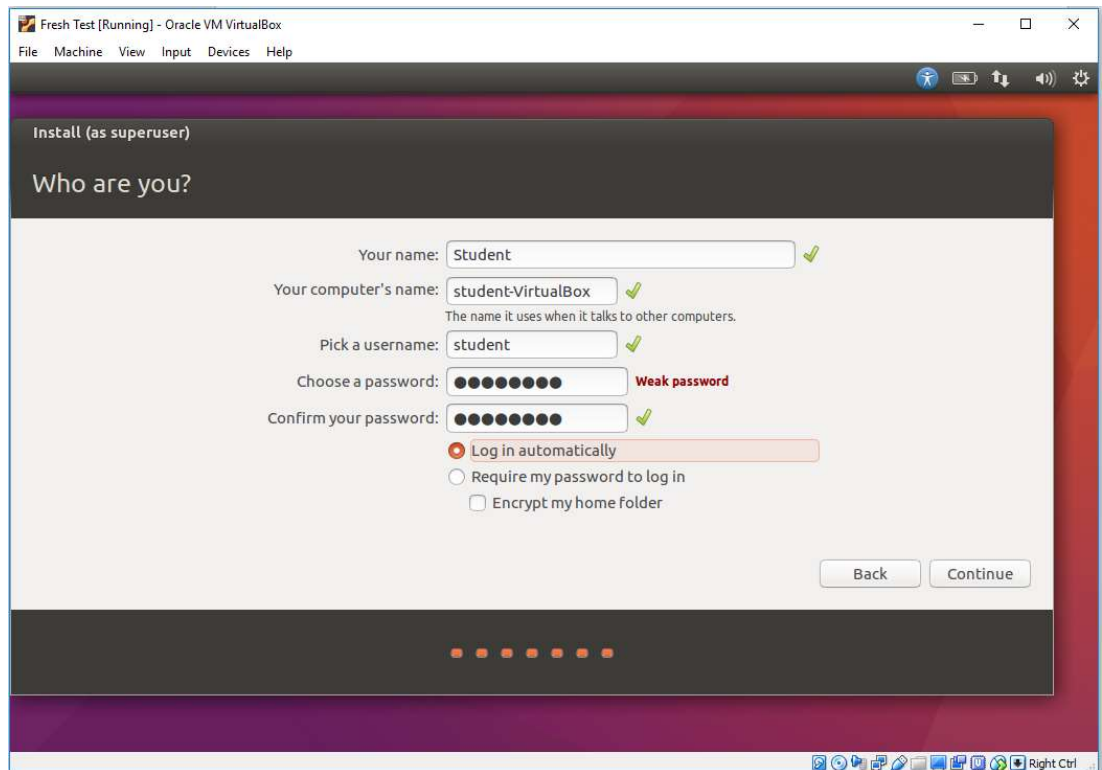


Keep the default of erase disk and install Ubuntu. It will erase the virtual disk you created in the previous steps (already empty) and install Ubuntu in that. Windows will not be affected.



Press “Install Now” and then “Continue” when asked about partitions. Select your location as Chicago to set the clock. Leave the keyboard defaults for English (US).

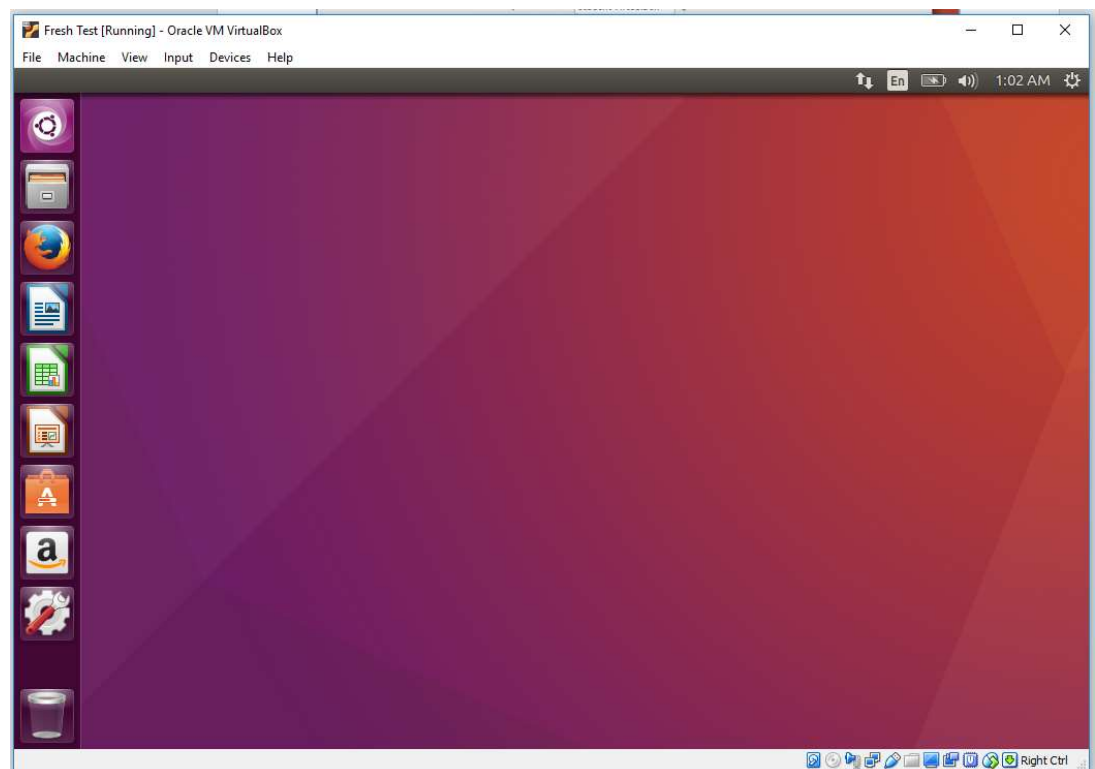
Fill in your info for what you would like your login to be. Optionally check Log in automatically to save a step when starting up the VM. Then hit Continue.



It will take some time to install. Just wait patiently. When prompted, click Restart Now. It will only restart the VM, not Windows.

It will ask you to remove the installation media, then press Enter. VirtualBox has already removed the ISO, so just press Enter.

You should now see a fresh Ubuntu install.



Setting up the Project

For development work, we recommend installing Google Chrome, Webstorm, and WireShark. There are a handful of other development programs that will be needed. To simplify things, we have created a script that you can run which should install everything in its default location. To use this, simply follow the instructions below.

1. Right click on the desktop and select "Open terminal"
2. Enter the command
`wget -O - https://pastebin.com/raw/fpVP5est | tr -d '\r' | sudo bash`
3. It will take a while, and at some point might ask for your sudo password.
4. When asked if non-superusers should be able to capture packets, select <No> by pressing Enter
5. It will warn you that the NodeJS version is out of date. It's fine. It is setup to use the older version. Let it wait for 20 seconds and it will automatically continue.
6. At some point, during the yarn / lerna install, it will pause for a very long time. (Install dependencies: pruning tree) or something like that. It can take up to 15 minutes. Just let it sit. It's not frozen, it just takes a while
7. It should finish and the last output should be Congrats! You are all set.

Google Chrome is installed, and can be launched from the Ubuntu menu.

Wireshark can be launched from the terminal (must be su) by typing `sudo wireshark`

Webstorm can also be launched from the terminal by simply typing `webstorm`.

Webstorm will need you to set it up.

1. Do not import settings
2. Scroll down and Accept
3. Don't Send
4. You can activate it with your JetBrains account info you likely already have, or evaluate for free (but only for 30 days)
5. Once it is configured, you can right click on the icon in the launched and select "Lock to Launcher" to make it easier to launch in the future.

Launching the Project

The project consists of two different repositories. The part you will be working on it called `rea-app` and is located at `$/rea-app`. It communicates with the backend and database from the `valuenetwork` project (located at `$/valuenetwork`). To start up the project,

1. Open two terminal windows
2. Launch the `valuenetwork` project in the first one by executing
`cd valuenetwork`
`sudo ./manage.py runserver`
3. Launch the `rea-app` project in the second one by executing
`sudo npm run dev`
4. Check that the backend is running by opening Chrome and going to `localhost:8000`.
5. Check that the front end is running by going to `0.0.0.0:3000`.
6. You are all set to work on the project