

Scotti-BYTE Industries

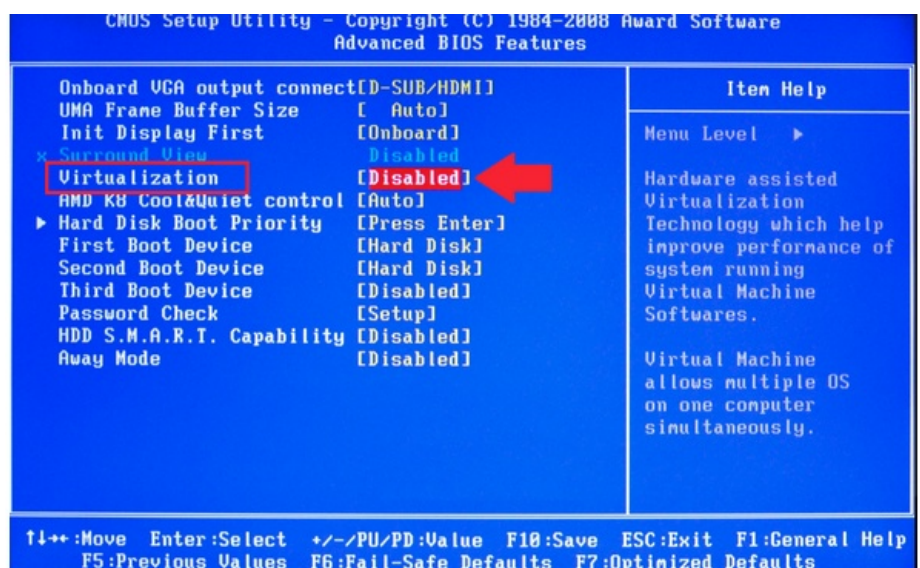
Virtualization Tutorial: Install VirtualBox on a Windows 10 system and Install Ubuntu 18.04 LTS Desktop as a guest Virtual Machine

Once the file saves, choose "Open in Folder". Once the file saves, choose "Open in Folder". Virtualization is a broad topic that includes both container virtualization and machine virtualization.

Containers are either Docker or Linux Containers (LXC). Docker or LXC are very lightweight and use very few resources since they are designed to virtualize an application only and not the entire operating system or the underlying hardware. For that reason, Docker and LXC containers must be Linux applications. Until recently, containers had to be run on Linux only since a key requirement is that the underlying operating system must be Linux. Very recently, the Windows Subsystem for Linux (WSL) was added as a compatibility layer for running Linux binary executables natively on Windows 10 and Windows Server 2019. As of May 2019, WSL 2 was announced which allows an entire real Linux kernel to run under Windows. Container virtualization is a topic all by itself. That's not the subject of this blog.

The other type of virtualization which this blog is focused on is machine virtualization. A virtual machine (VM) is an emulation of an entire computer system hardware including the operating system. VMs provide the complete functionality of a physical computer and are not limited to running just Linux. VMs execute entire operating systems such as Windows, Mac OSX, Android, and all sorts of distros of Linux. A hypervisor uses the underlying virtualization hardware of a CPU chip to allow for native execution of instructions and management of hardware.

It is possible to run a main operating system such as Windows 10 and then to have a Hypervisor running that hosts one or more VMs at the same time. The limitations are only that the CPU must support hardware virtualization and there must be adequate memory and disk space resources to host all of the VMs that you want to run at once. Hardware virtualization must also be turned on at the BIOS level. Every BIOS is different and hardware virtualization is generally turned on by default. You should be sure that virtualization is turned on (Enabled) in your BIOS.



Hardware virtualization has been supported on both Intel VT-x and AMD-V CPUs since 2005 and 2006 respectively.

Hypervisors are of Type 1 and Type 2. A Type 1 hypervisor runs directly on the hosts hardware to control and manage guest operating systems. For this reason, they are often referred to as bare metal hypervisors. VMware ESXi is probably the best known Type 1 Hypervisor. It would be loaded directly on a system without a host operating system. So, the hardware would be dedicated to running only VMs and nothing else.

A type 2 Hypervisor is more common. Examples of Type 2 Hypervisors are:

VirtualBox - This product is free and is distributed by Oracle. This is the product we will discuss today. It is a very full featured product and allows running multiple VMs:
<https://www.virtualbox.org/>

Hyper-V - This product is part of Windows 10 Professional and Windows Server 2012 and later.

VMware Workstation Player - This is a free Product that allows running one virtual machine on a single host PC. <https://www.vmware.com/products/workstation-player.html>

VMware Workstation Pro - This is a full blown commercial Type 2 Hypervisor for running multiple VMs. <https://www.vmware.com/products/workstation-pro.html>

QEMU - QEMU is free and is a very fast and lean hypervisor which allows you to run multiple VMs. <https://www.qemu.org/>

The purpose of this paper is to document how to install and use Virtualbox.

Go to the virtualbox webpage and download virtualbox.

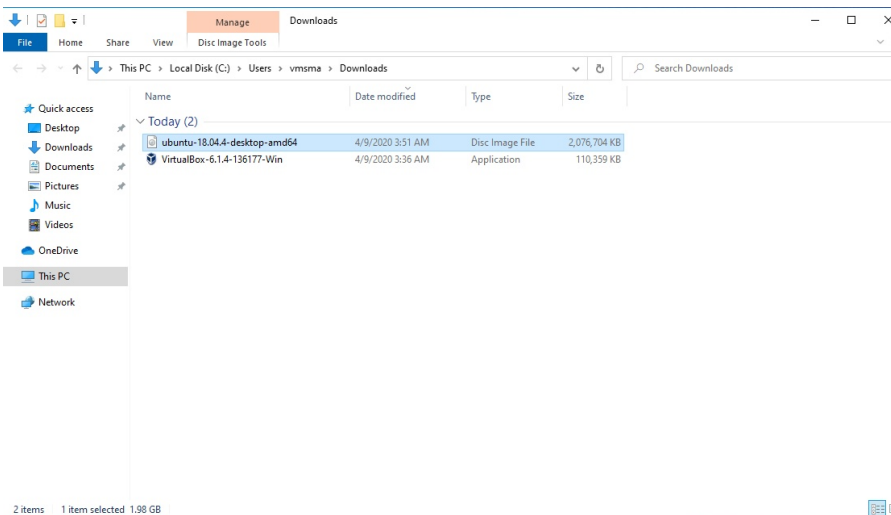
<https://www.virtualbox.org/wiki/Downloads>



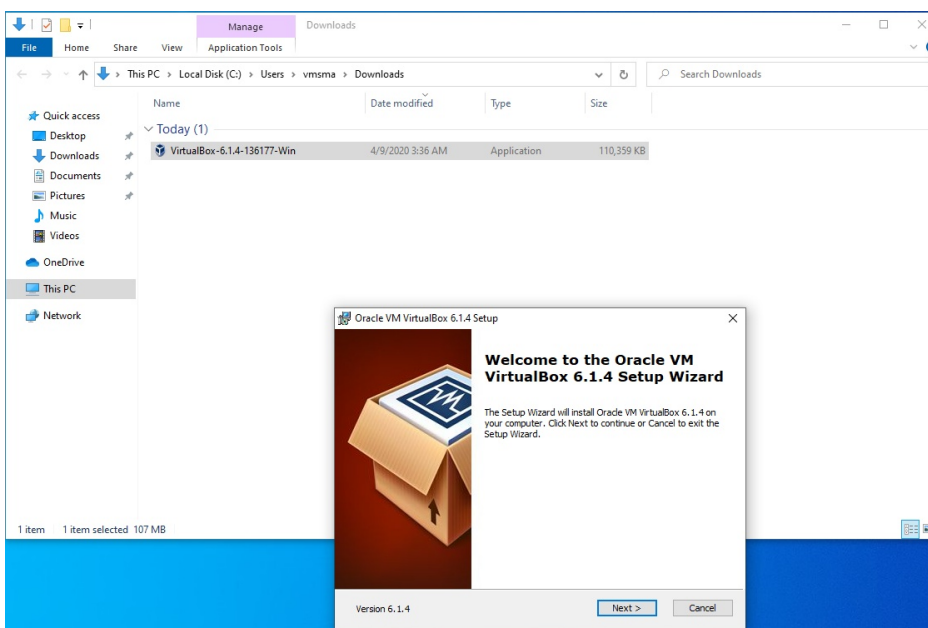
Be sure to click "Save" when you download Virtualbox.



Once the file saves, choose "Open in Folder".



Double click to install and follow the defaults.

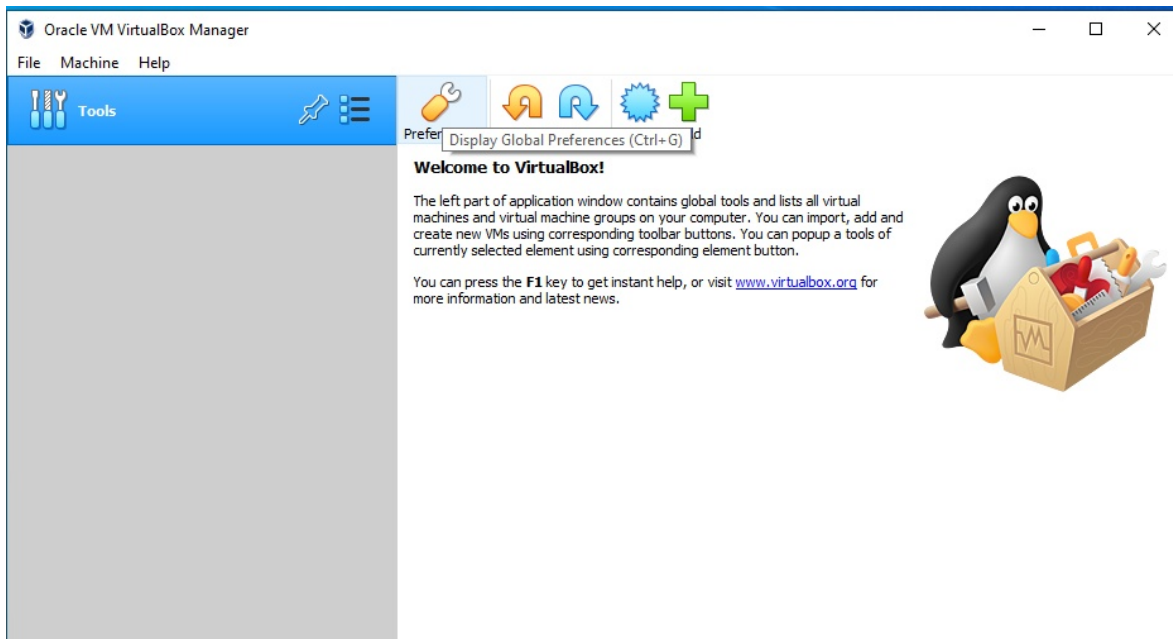


Restart Windows after the installation completes.

Launch VirtualBox from the "Oracle VM VirtualBox" icon created on your desktop.



The program window looks like this:



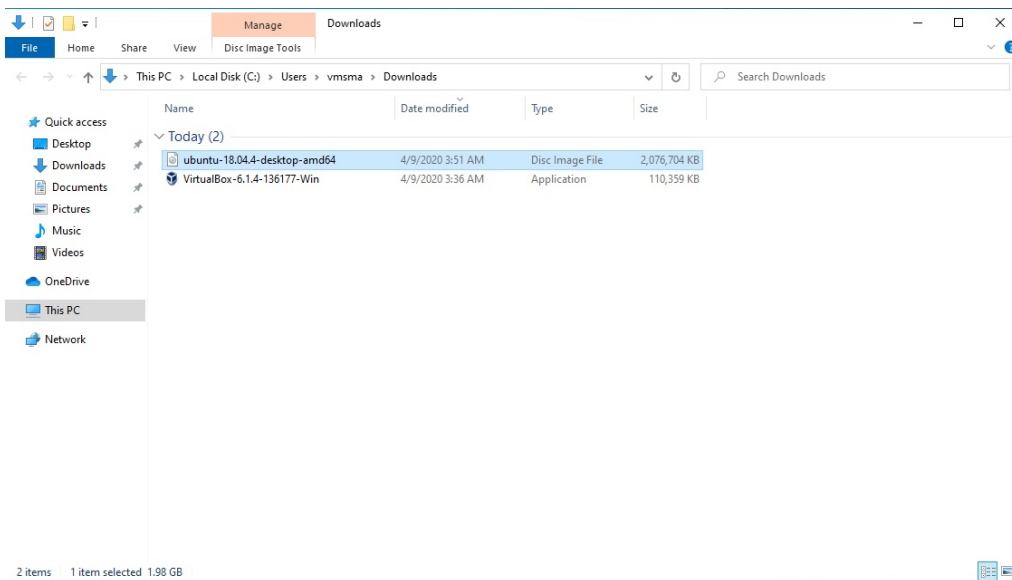
To create your first virtual machine, we will first need to download an operating system. Let's download Ubuntu 18.04 LTS desktop.

<https://ubuntu.com/download/desktop/thank-you?version=18.04.4&architecture=amd64>

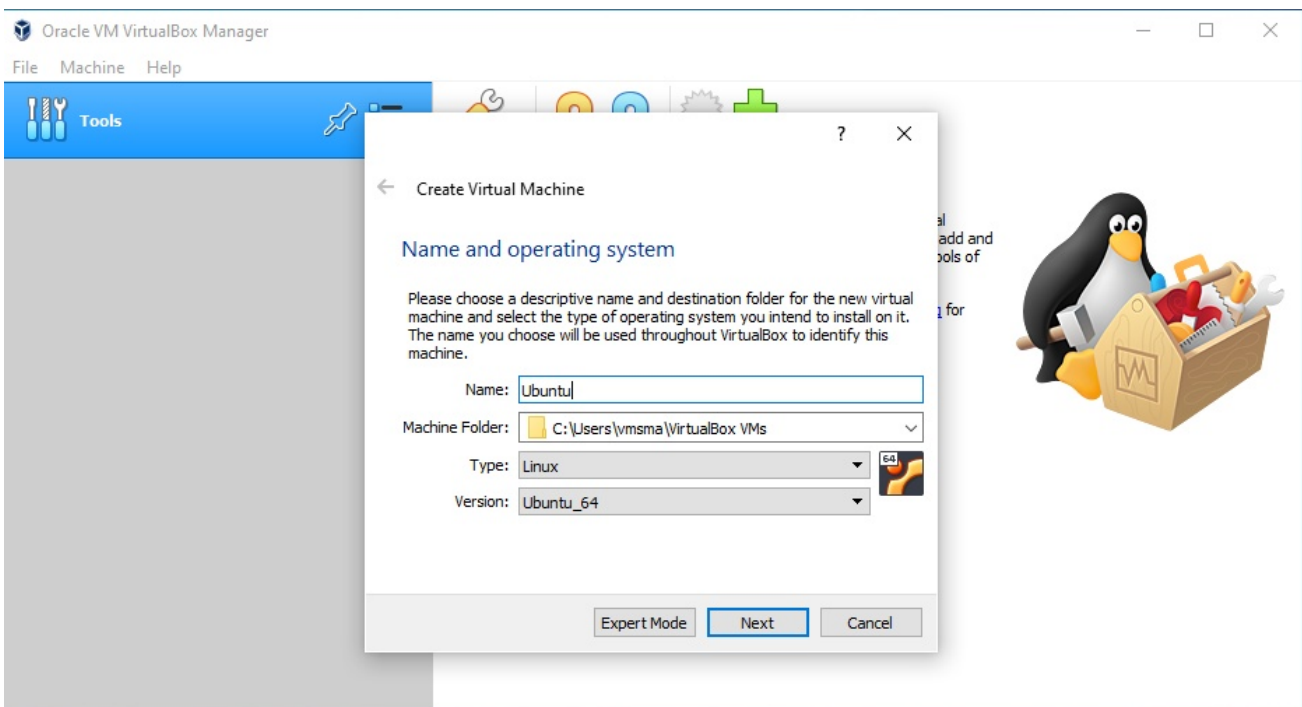
Be sure to click "Save" and wait for the download to complete. You are downloading Ubuntu 18.04 LTS Desktop which is probably the most popular Linux today. The download is an "ISO" file which is a digital image of a DVD disk.

Virtual machines can run Windows, Android, different Linux Distros, and even Mac OSX.

When the download completes, you will have the following in your download folder.

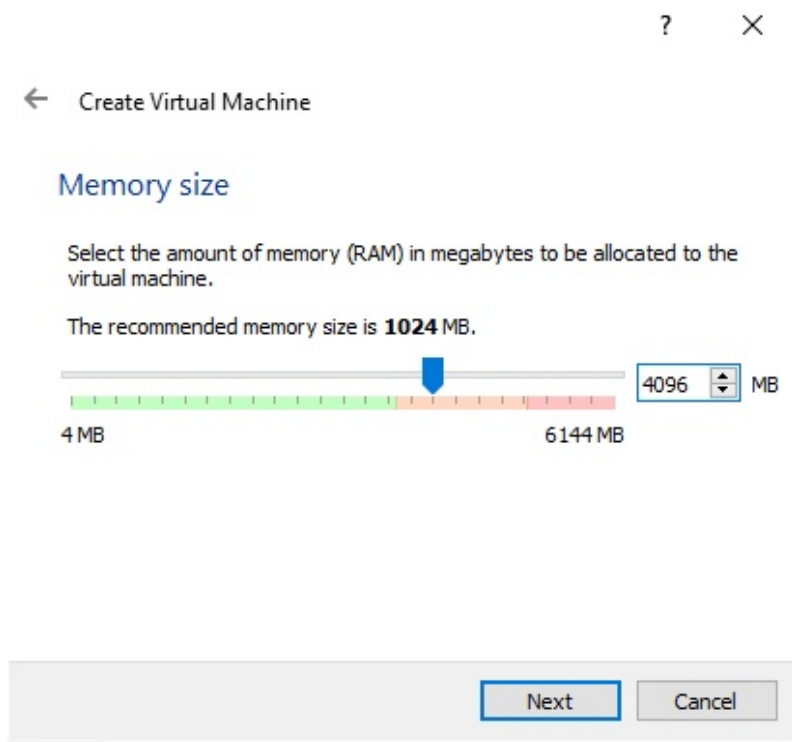


Go to the Oracle VM VirtualBox Manager window and click "New". Name your virtual machine Ubuntu:

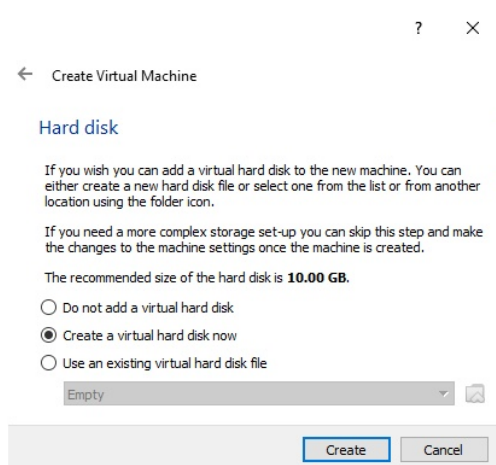


The type and version are automatically filled in for you. Click Next and enter the desired memory for your Ubuntu VM. Keep in mind that this memory is deducted from the memory of your main system while the VM is running. So, if you have a system with 16GB of memory and you allocate 4GB to this VM, the main system only has 12GB of memory when the VM is running.

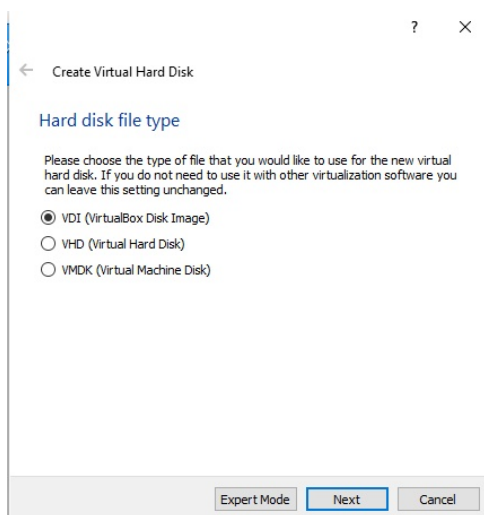
For our purposes, set the memory size to 4096MB since Ubuntu runs very well with 4GB.



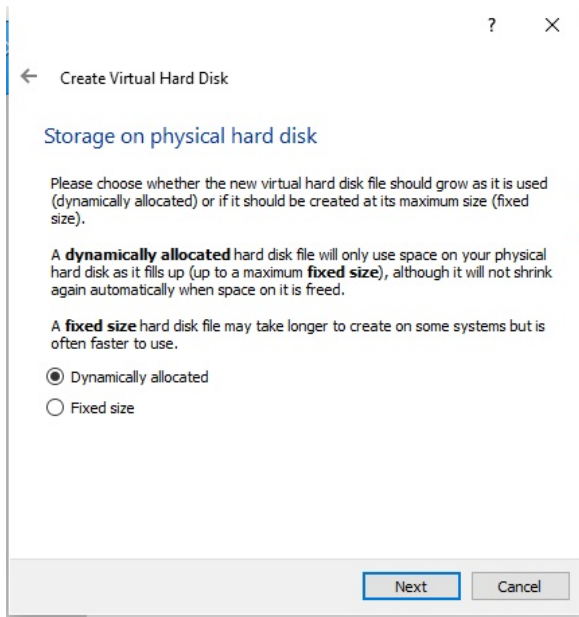
Next you are creating a virtual hard disk. This disk is emulated inside of a single file in the host operating system. For that reason, a VM can never affect any of the files on the main machine ever.



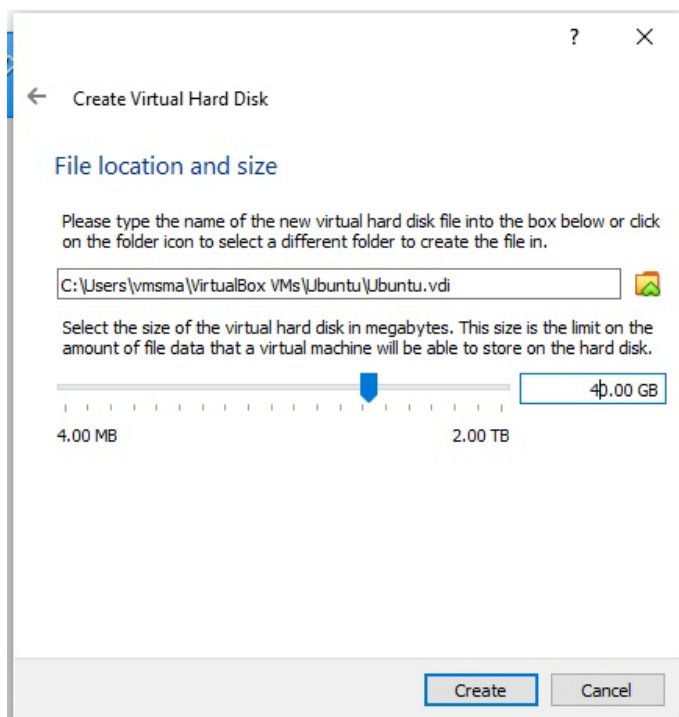
Default the disk to type "VDI".



Allow it to be dynamically allocated. This means that if you create a 40GB disk, that it will only grow to 40GB as it needs the space and not create a 40GB file to begin with.

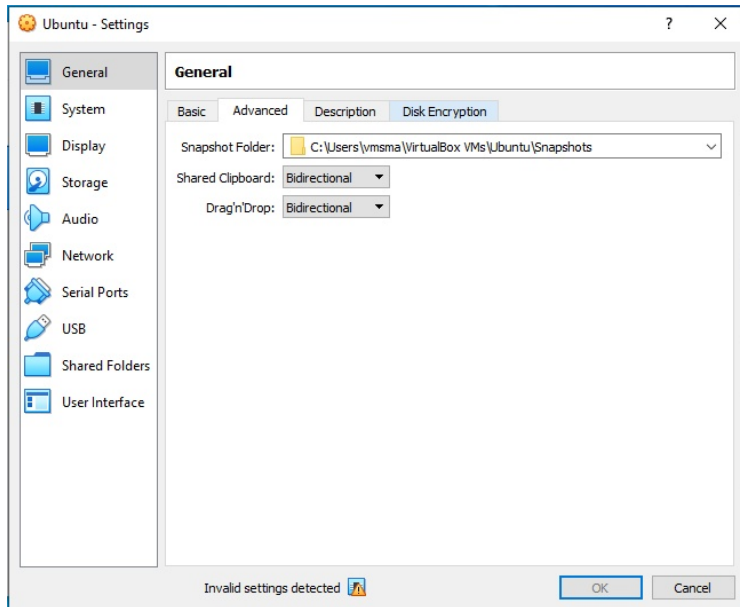


If you have the space on your hard drive, 40GB is a nice amount of space to load Ubuntu and many programs. If not, you can make the drive as small as 10GB.

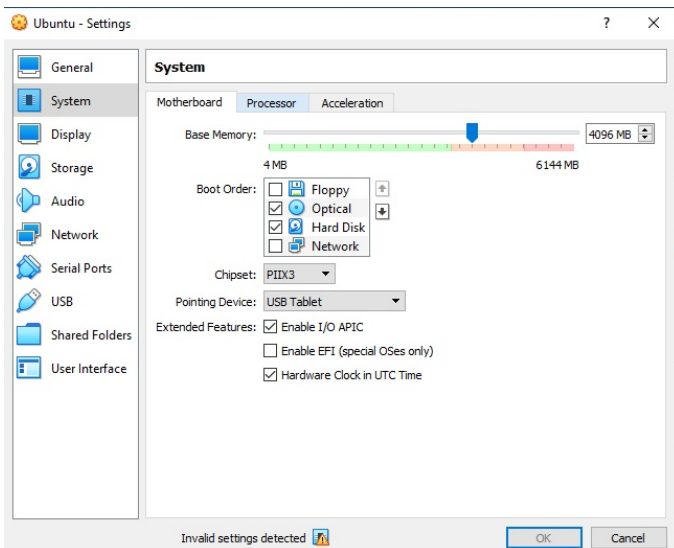


Next, click settings after clicking create in the above screen.

Go to General, and then the advanced tab and change the shared clipboard and drag and drop to bidirectional.

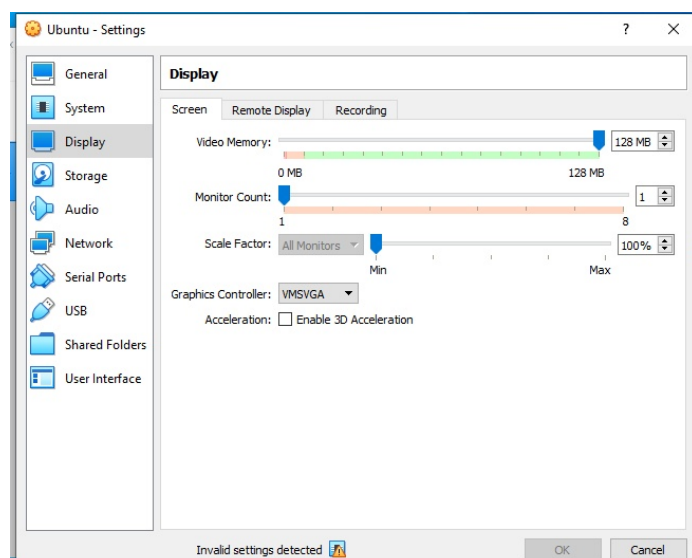


Go to System on the left side and uncheck floppy

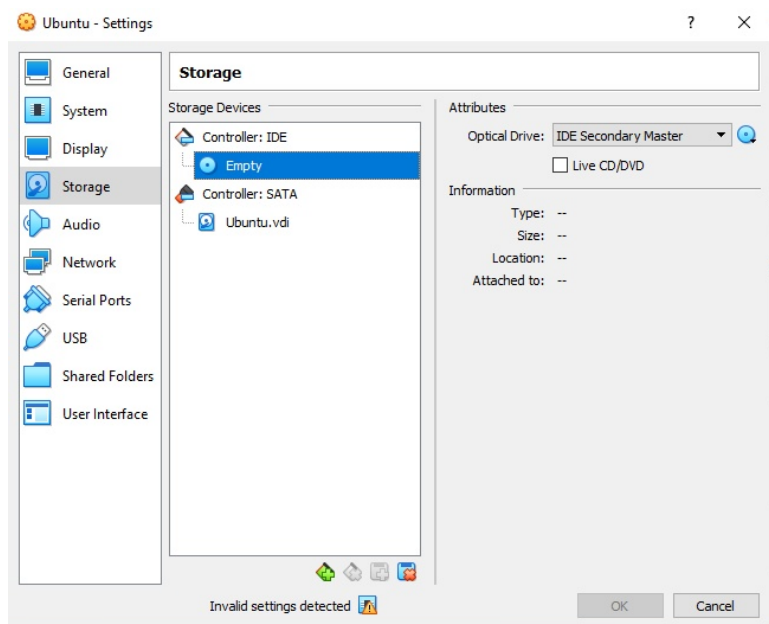


Go to the processor tab and change to 2 CPUs or more (Perhaps 2 is a good choice on a PC with 4 CPU cores)

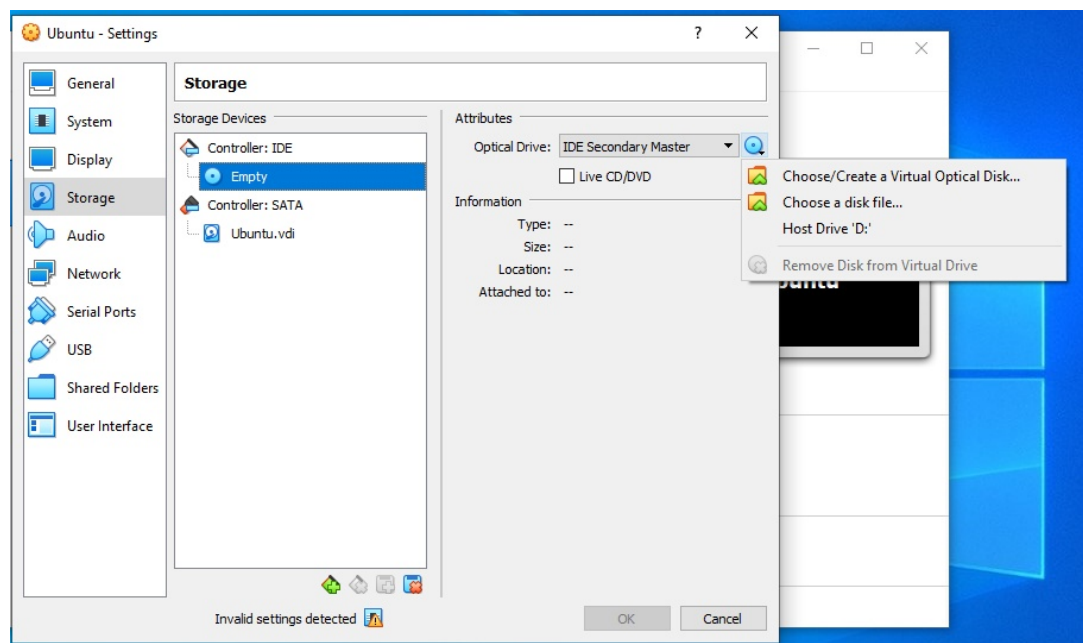
Go to Display on the left side and max the Video memory to 128MB



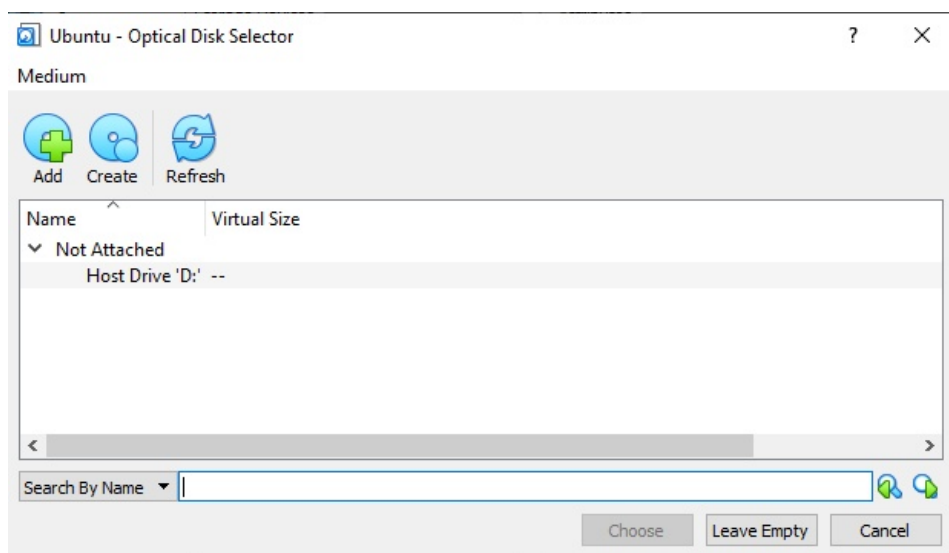
Go to storage and highlight the empty CD Rom Drive.



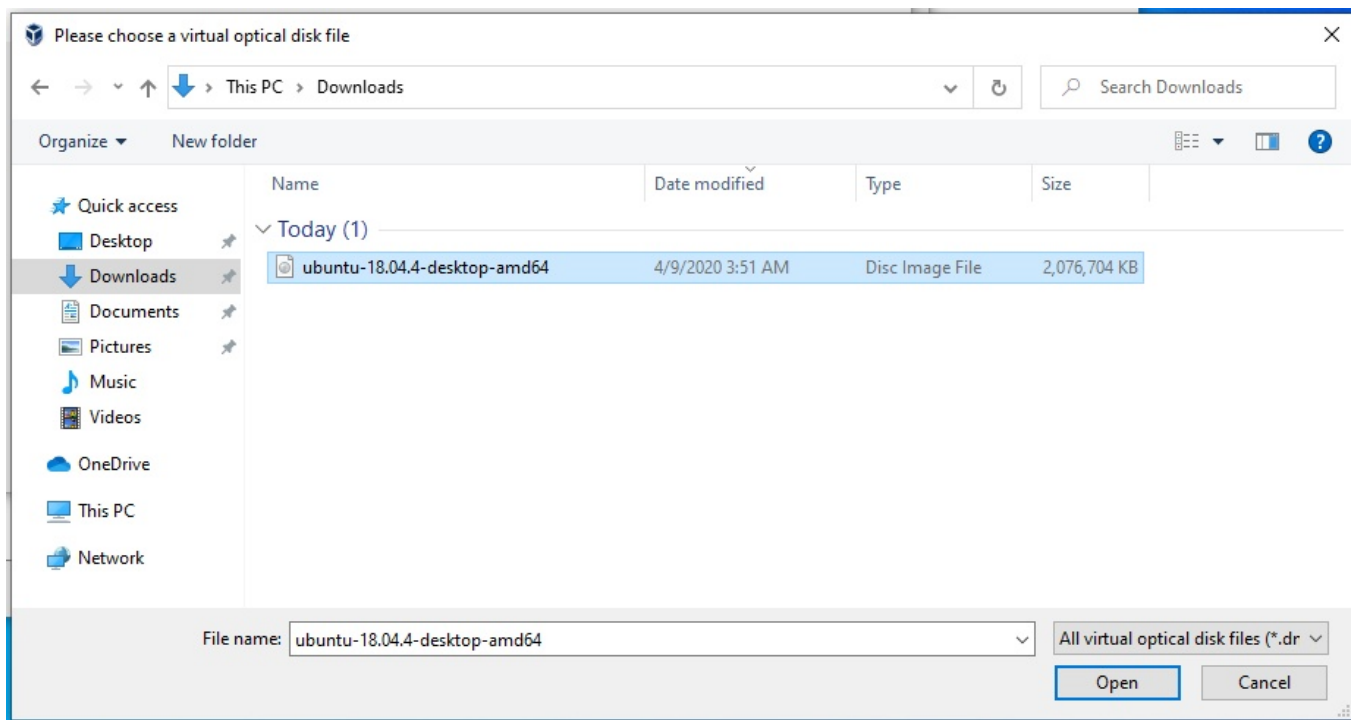
Click on the CD icon to the right of "IDE Secondary Master"



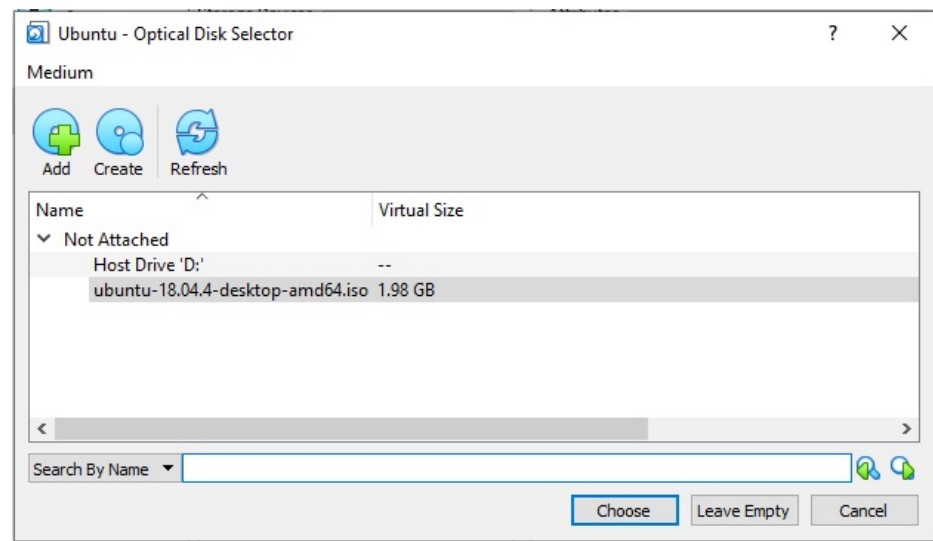
Click Choose/create an optical disk.



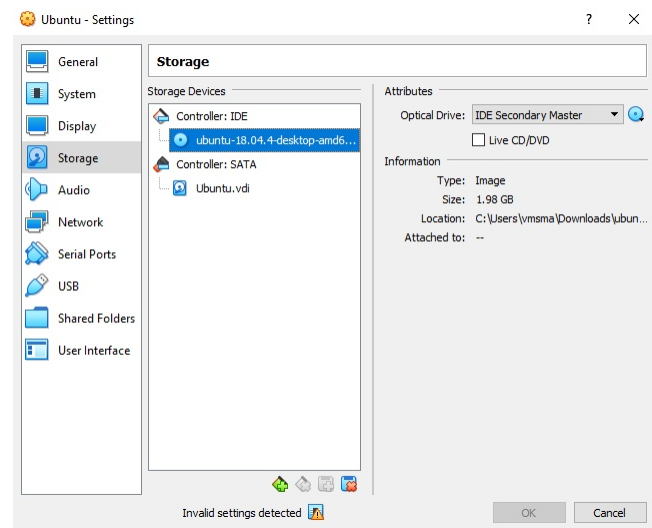
Click "Add". Go to the "Downloads" folder and your Ubuntu should be highlighted.



Click Open.

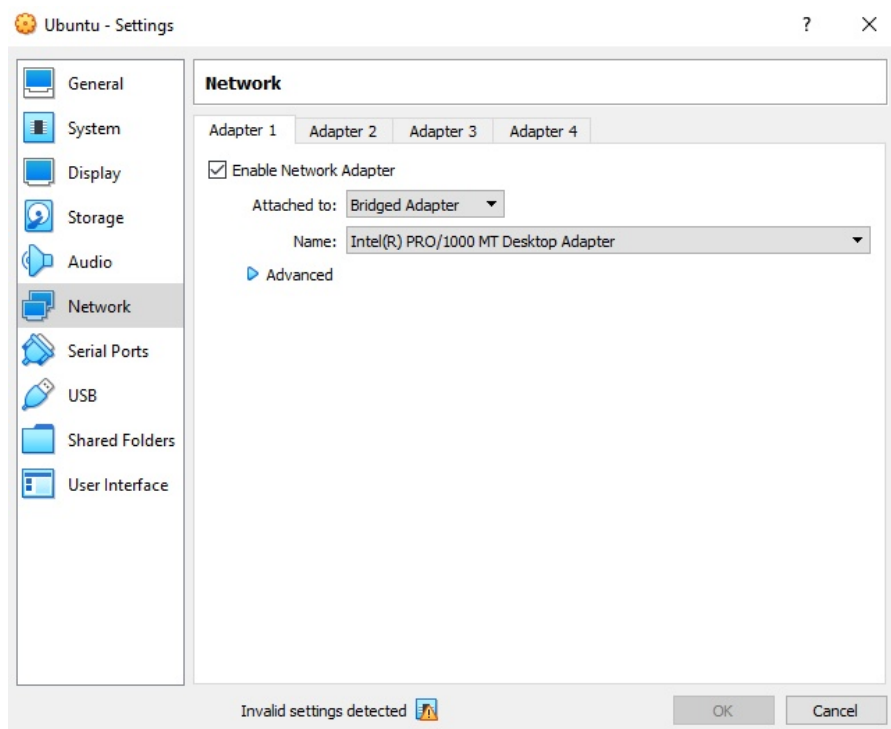


Now Click "Choose".



You can now see above that you have your ubuntu virtual disk image inserted into your virtual DVD disk drive. You can also see that the system knows about your Ubuntu.vdi virtual disk which is an empty file on your windows hard drive in which we will be installing the Ubuntu Linux operating system.

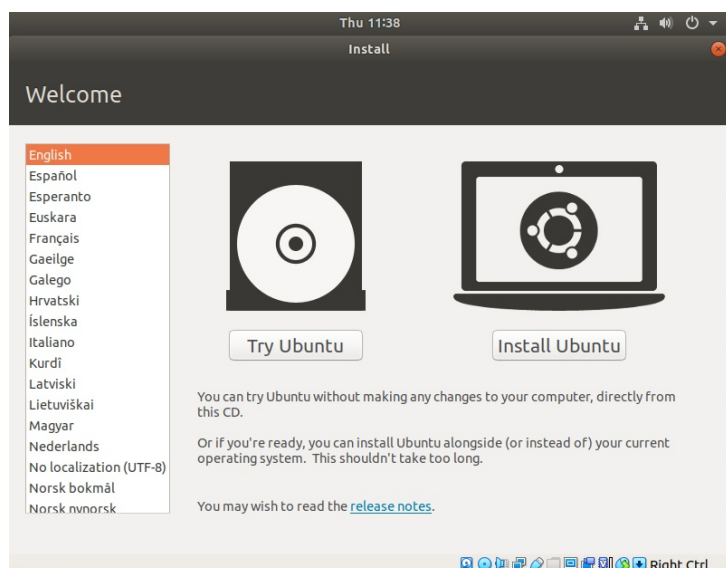
Go to "Network" on the left side and change to "Bridged Adapter" and choose your network card. Note that you might have a choice of your wireless or wired card on your Windows host operating system. Be sure to choose the one that is actually connected to the Internet.



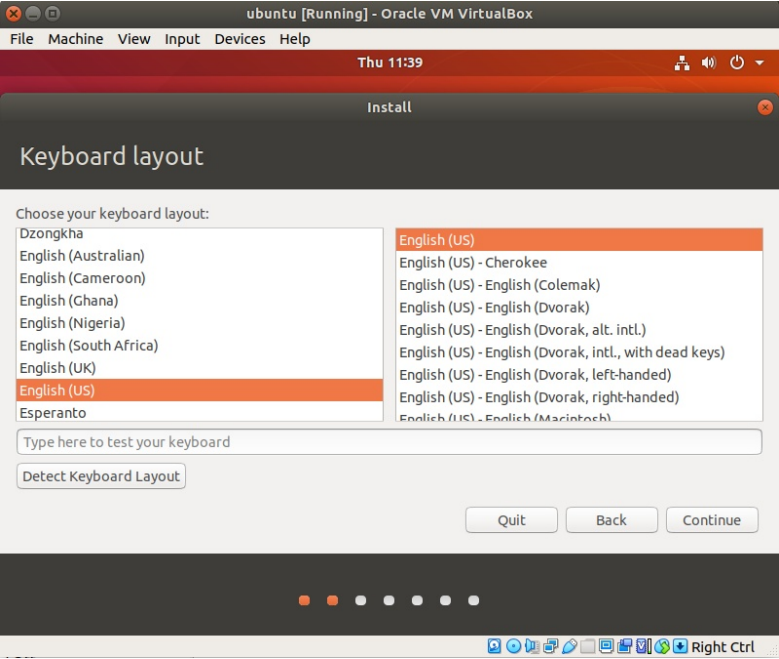
At this point, you should be able to hit ok. Note that my screen above has "OK" grayed out and says that their are invalid settings. I only have that because I wanted this guide to be for configuring VirtualBox on Windows and so I created a virtual machine inside of a virtual machine. Not advised.

Next, you should be back to the virtualbox main screen. Click "Start" at the top and your virtual machine should boot up the Ubuntu installation media.

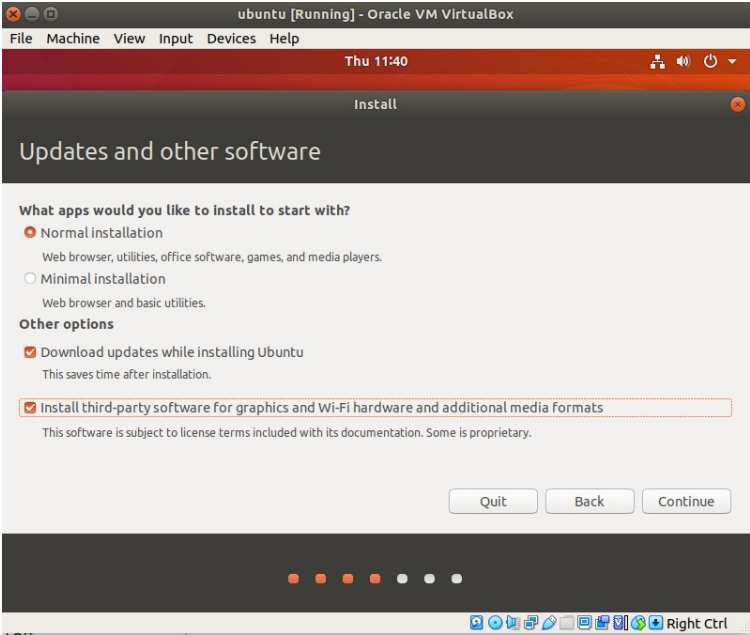
You should get a screen like this after a minute or so.



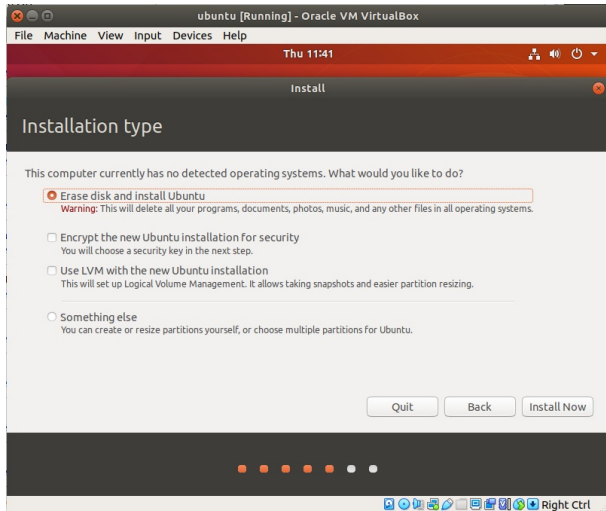
Click on "Install Ubuntu"



Choose English as above and continue

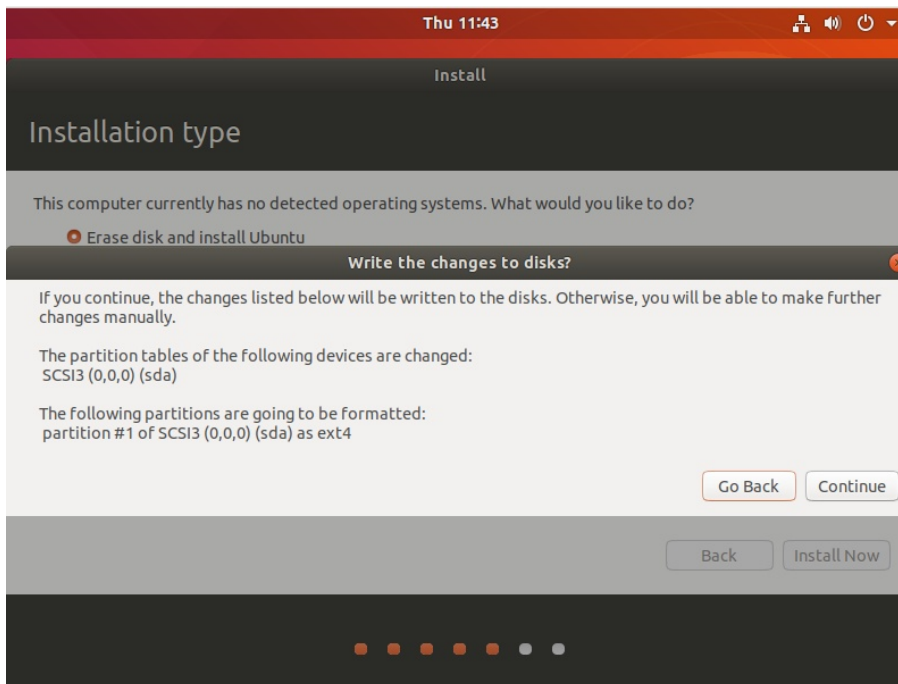


Check the options as above and continue



Choose erase disk and install ubuntu as indicated above. Don't worry, this is erasing the virtual disk ubuntu.vdi which is a single file on your windows system. Remember that the VM cannot see your windows disk directly.

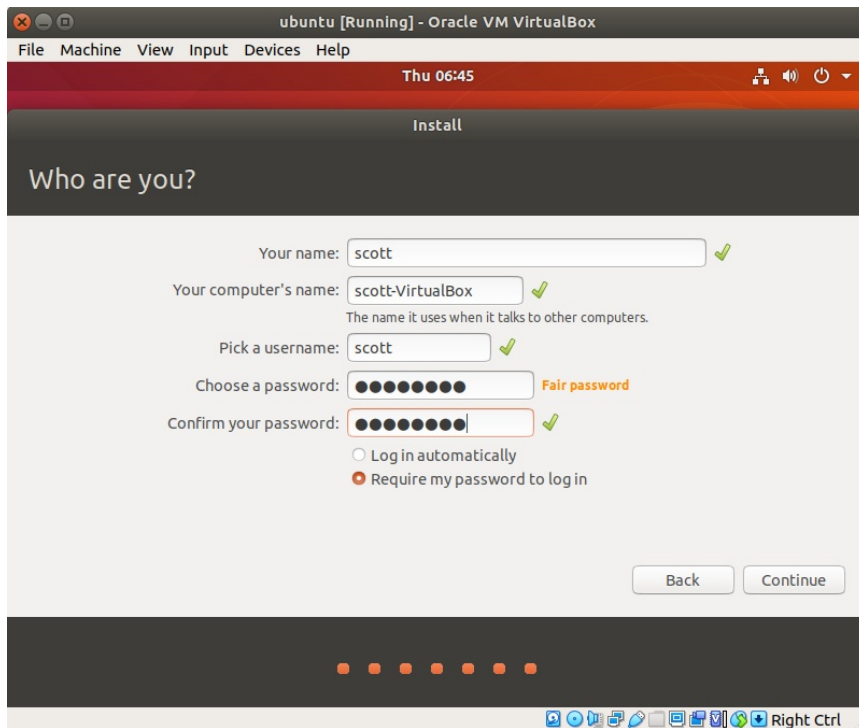
Next, click "Install Now"



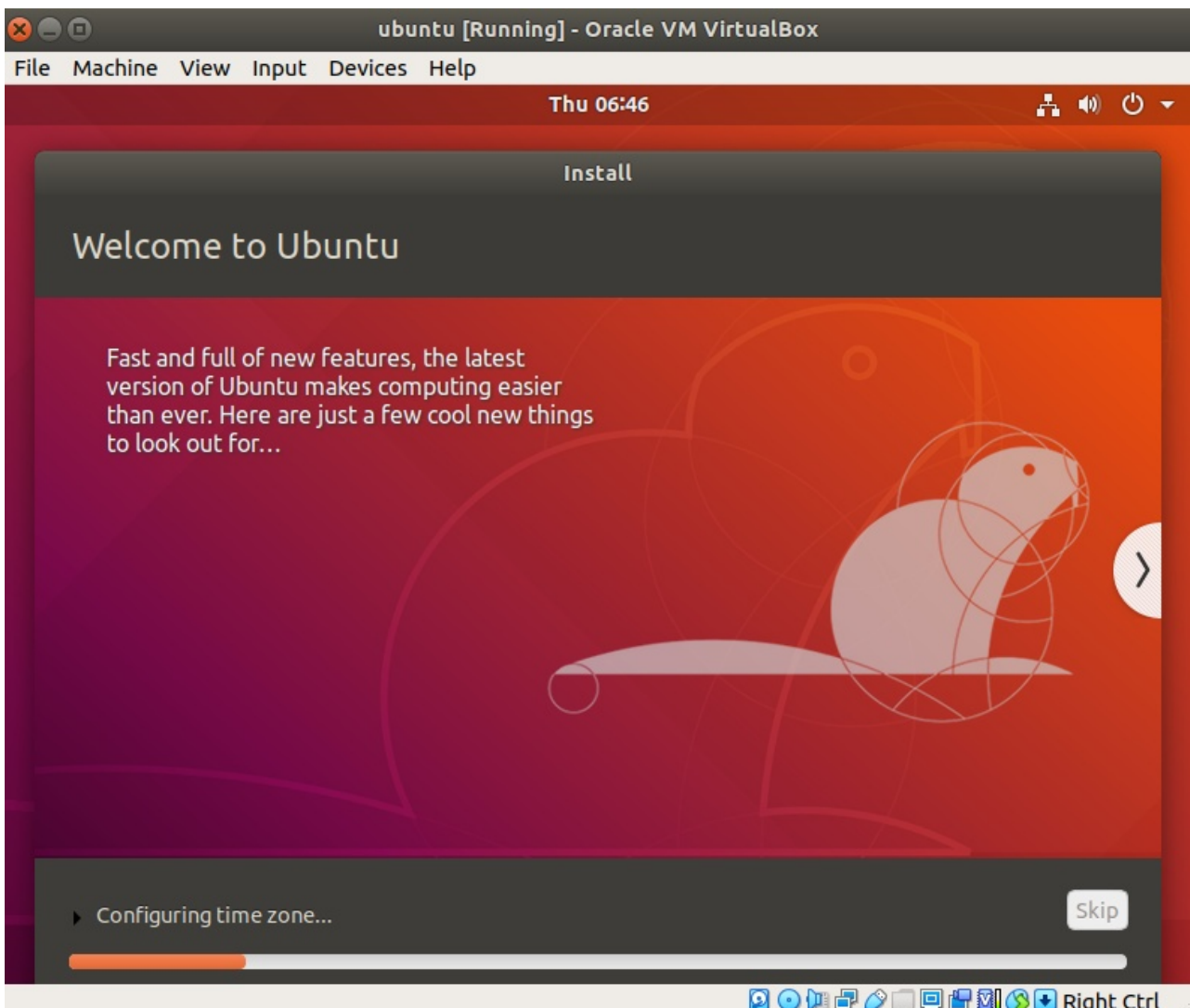
Click Continue in the screen above.



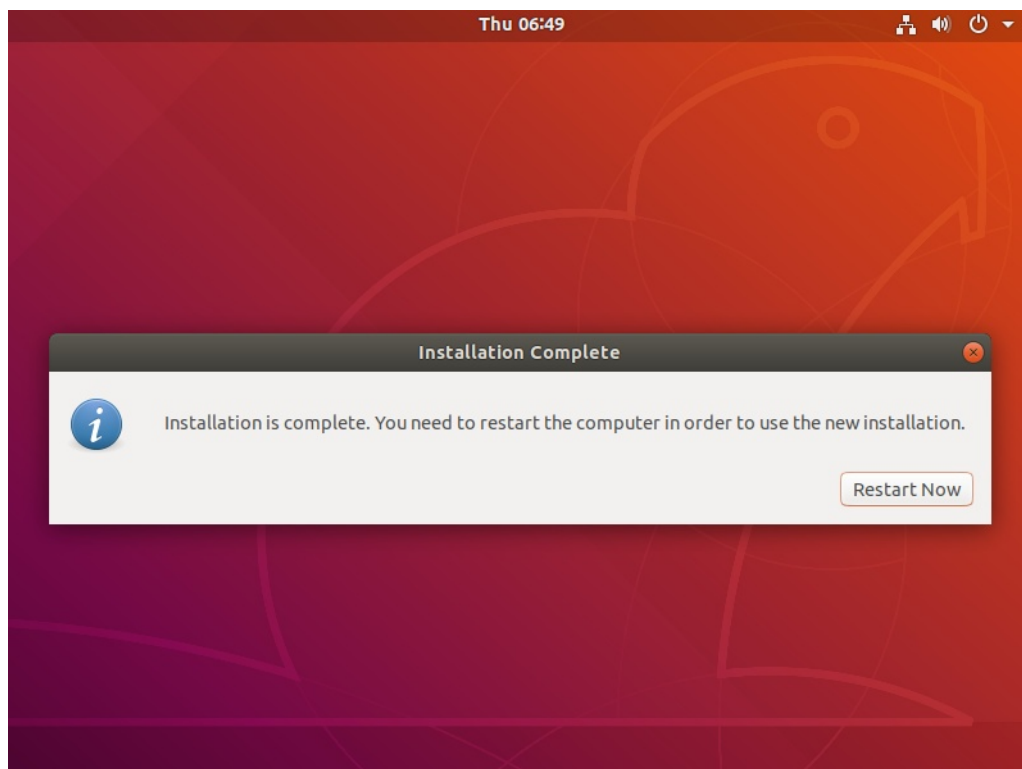
Set your time zone above and press continue.



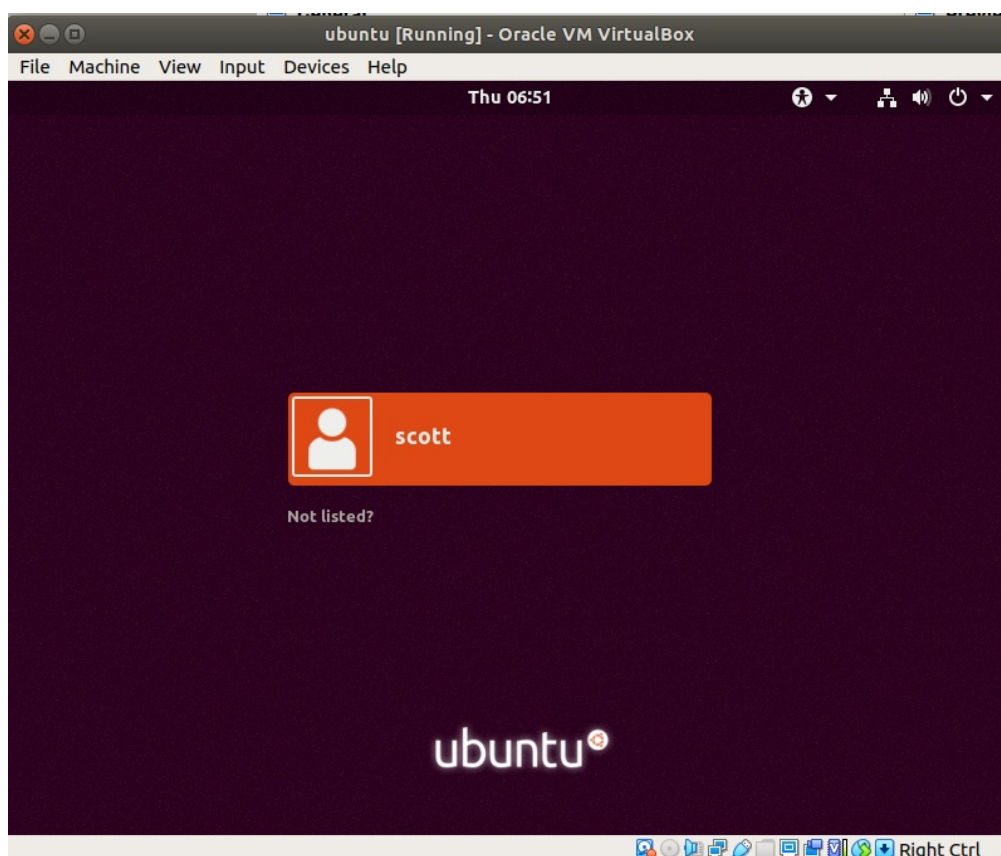
Enter the data above to name your Ubuntu computer name and to create an account. Then press continue.



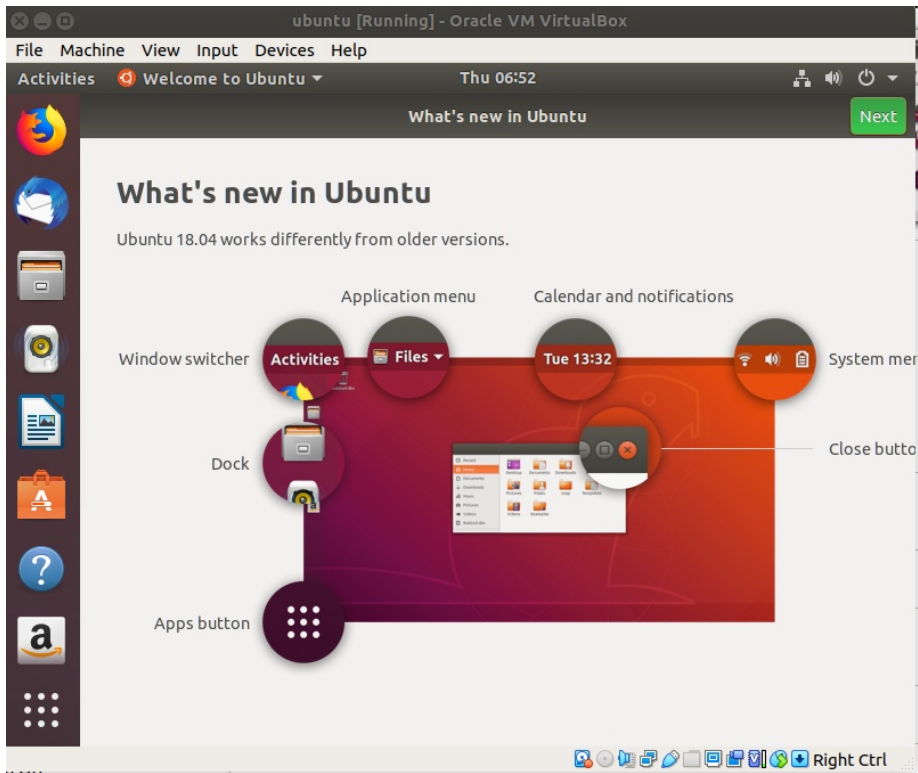
Installation continues as shown above.



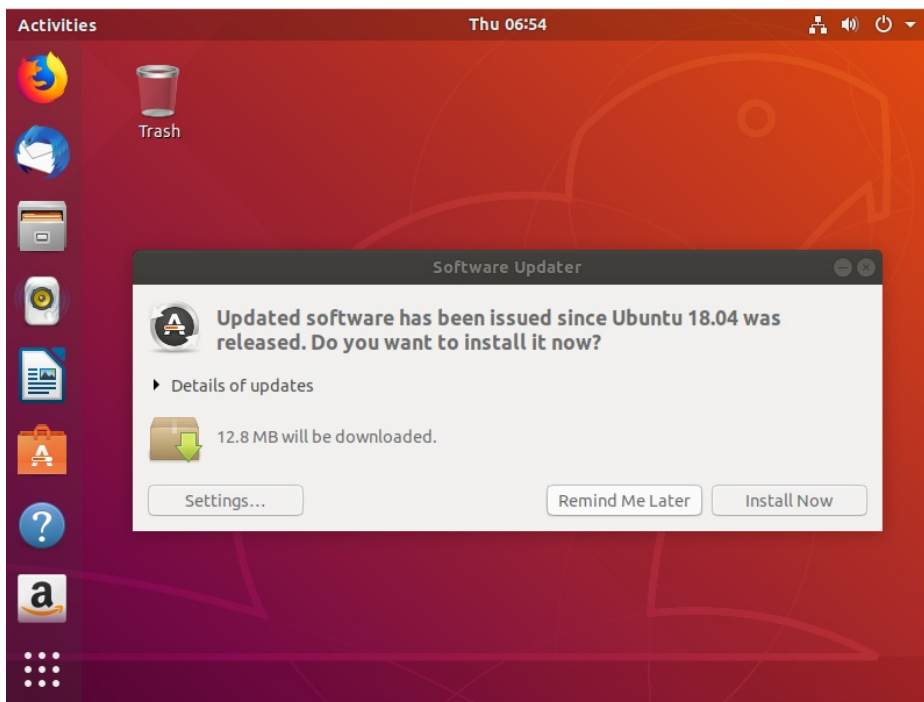
Click Restart Now above to reboot your Ubuntu VM.



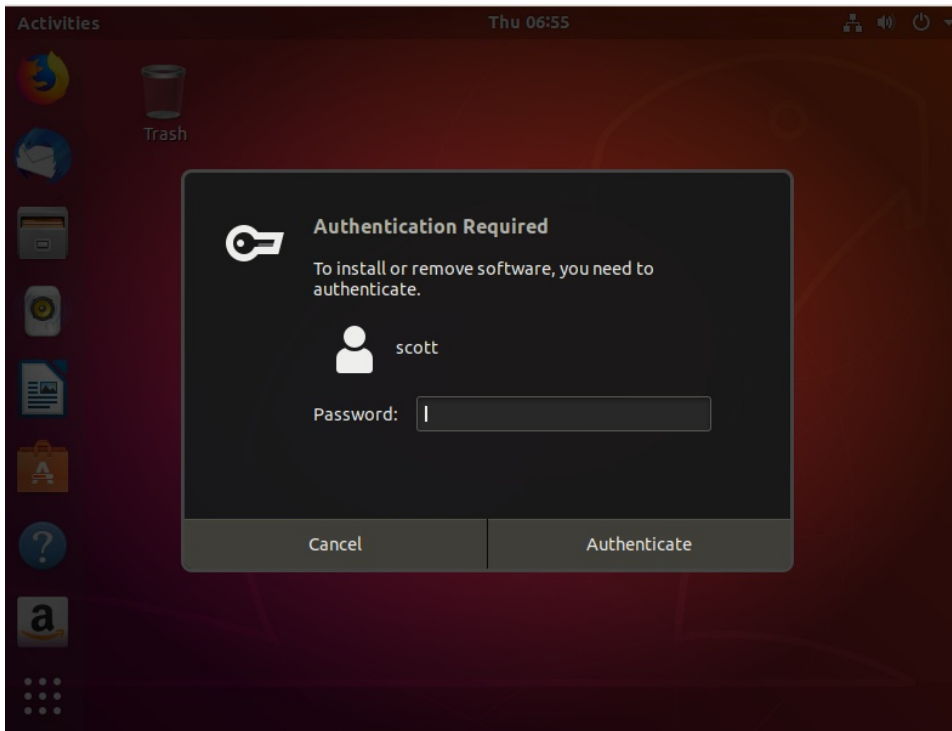
Click on your username above and enter your password.



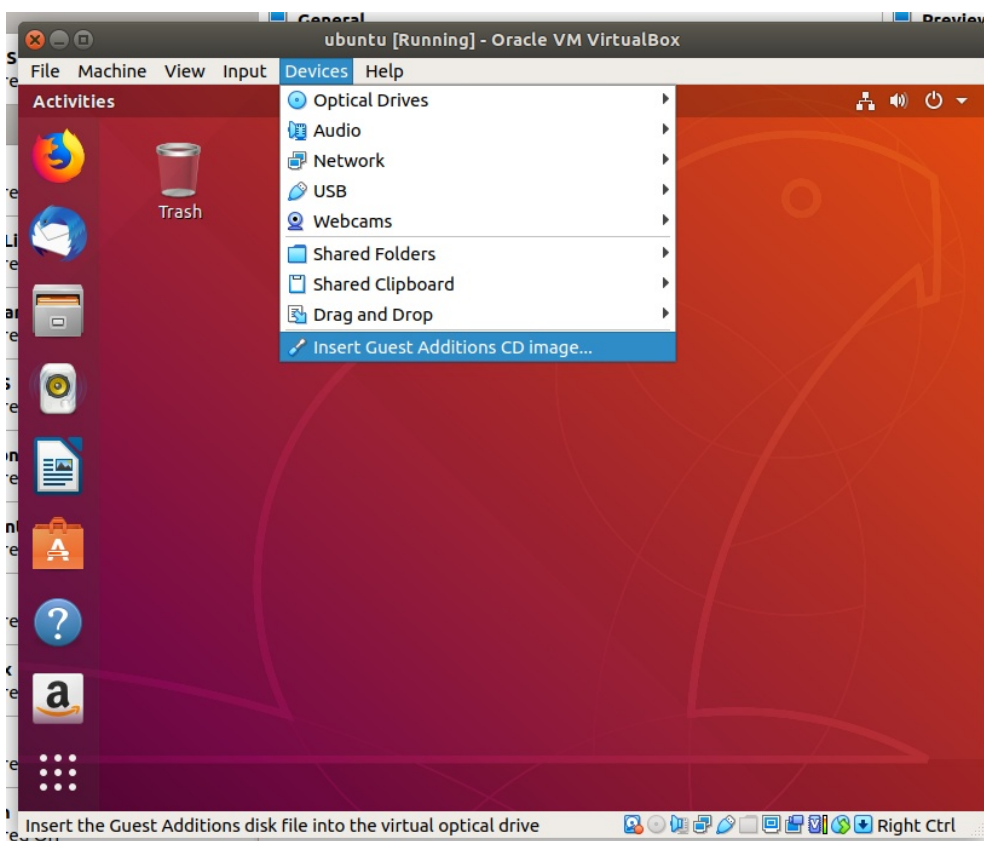
Once signed on, click through the above welcome screens. You can bypass "Livepatch". Once you click "Done" you will probably be presented with updates.



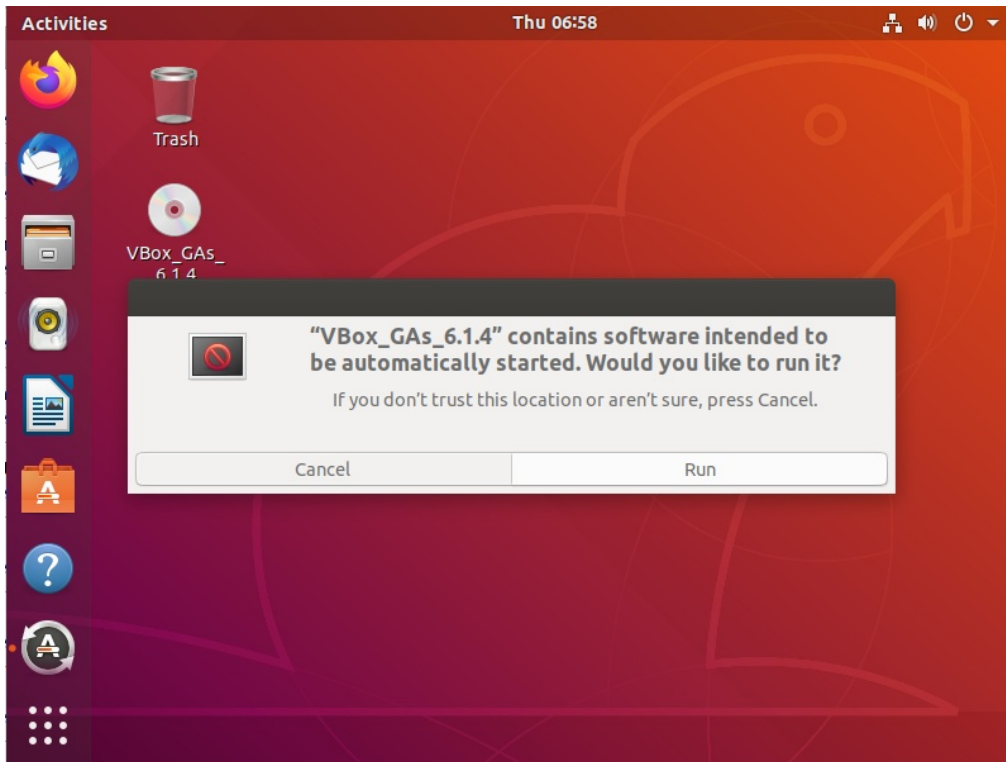
Click install now.



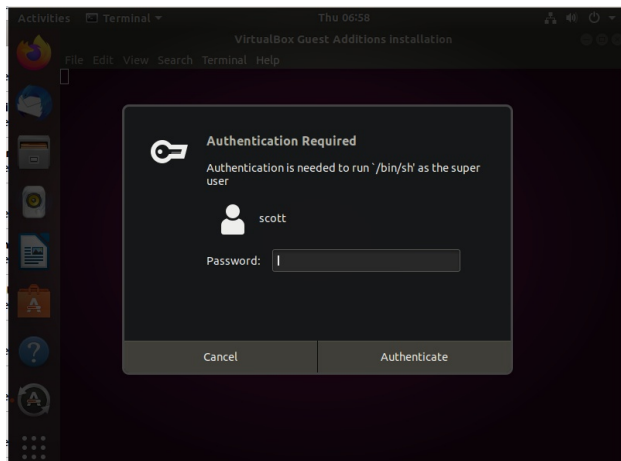
Enter your password and click "Authenticate".



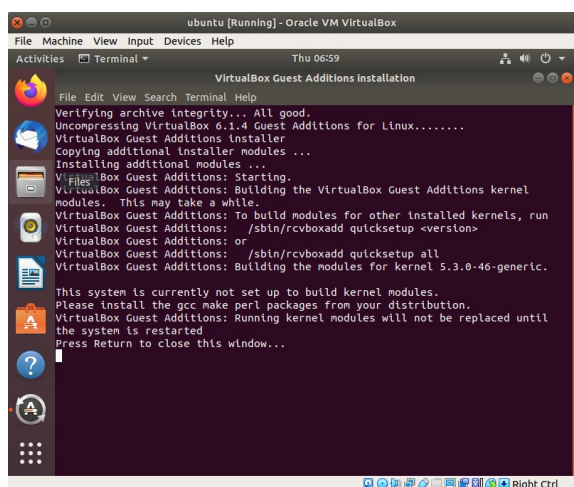
Go to the "Devices" pulldown menu and choose "Insert Guest Additions CD Image".



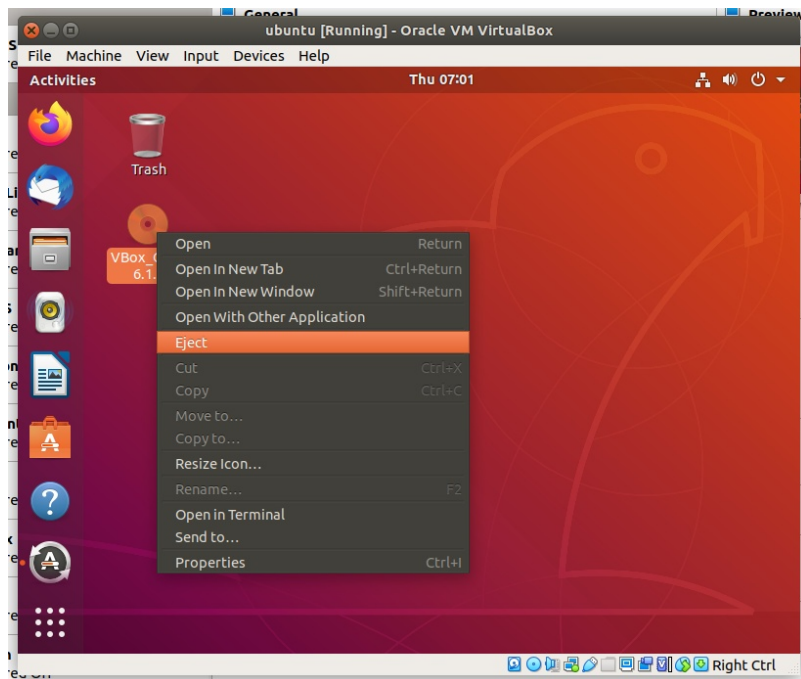
Choose "Run" when the above screen comes up.



Type your password again and click authenticate.

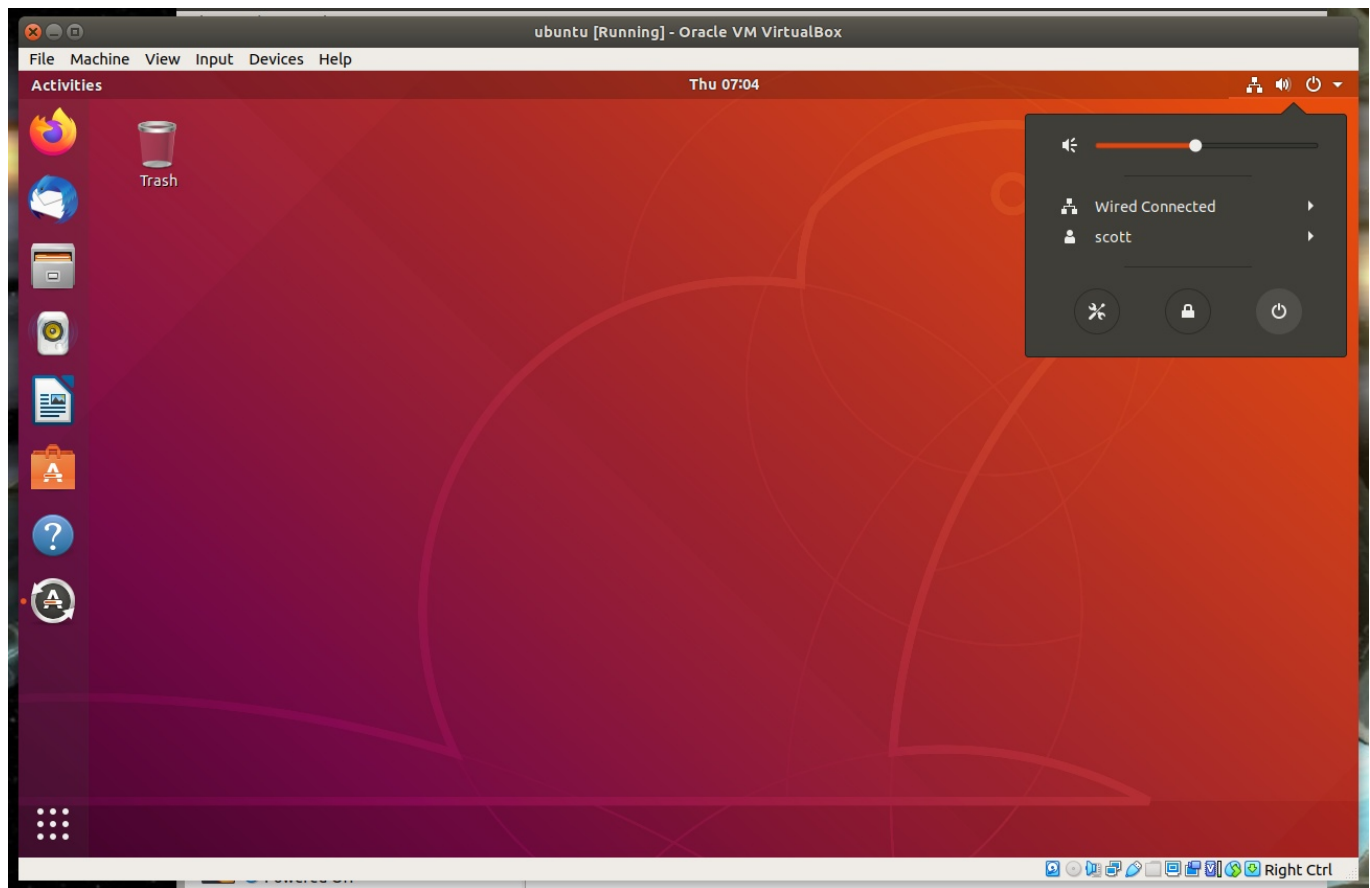


The script to run the guest additions runs above. Guest additions add proper screen control, resolution, and resizing and also gives you mouse and pointer control and drag and drop features from the host to the VM. Press Return and the terminal window will exit.

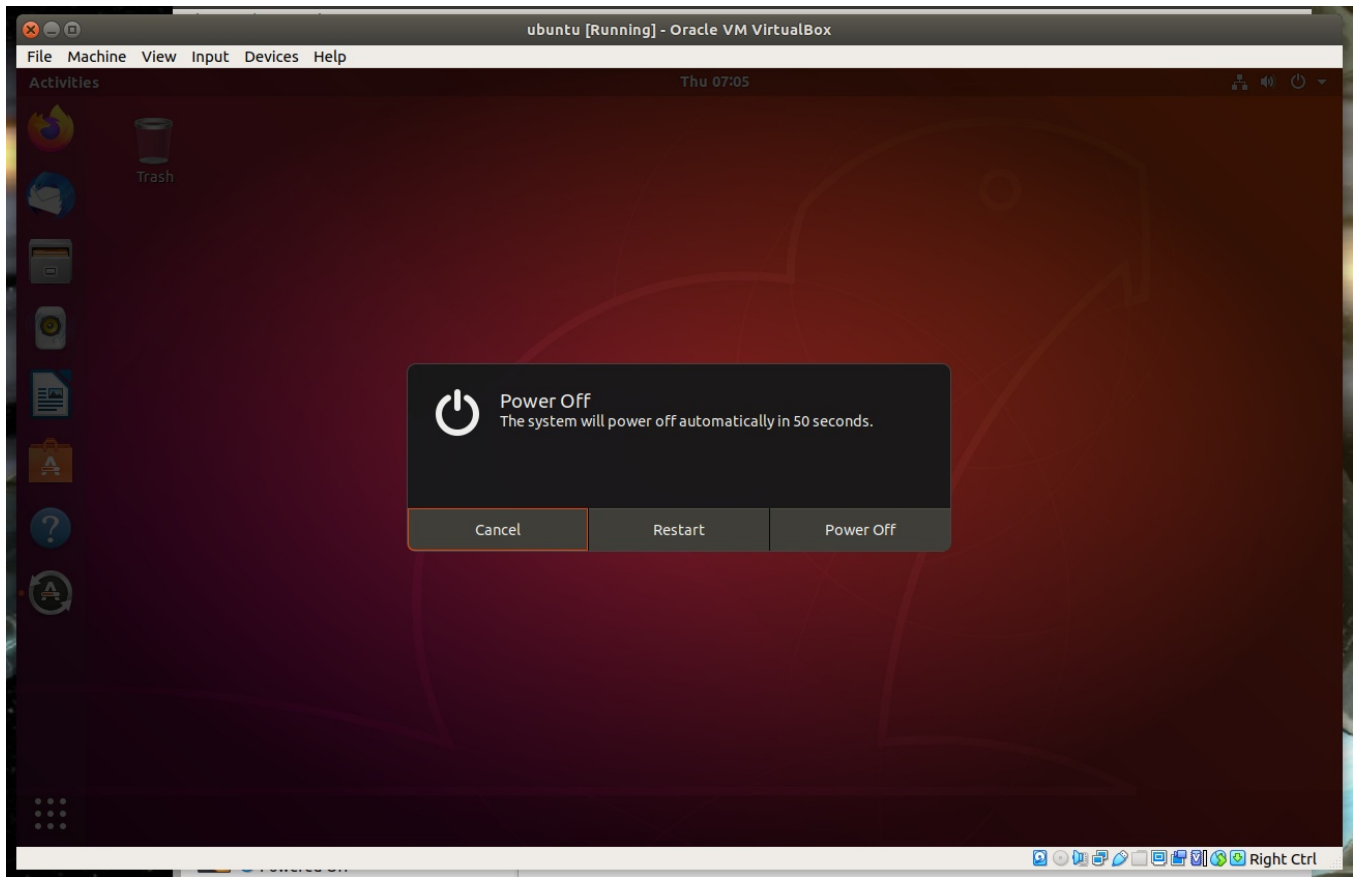


Right click on the Guest additions CD above and eject it.

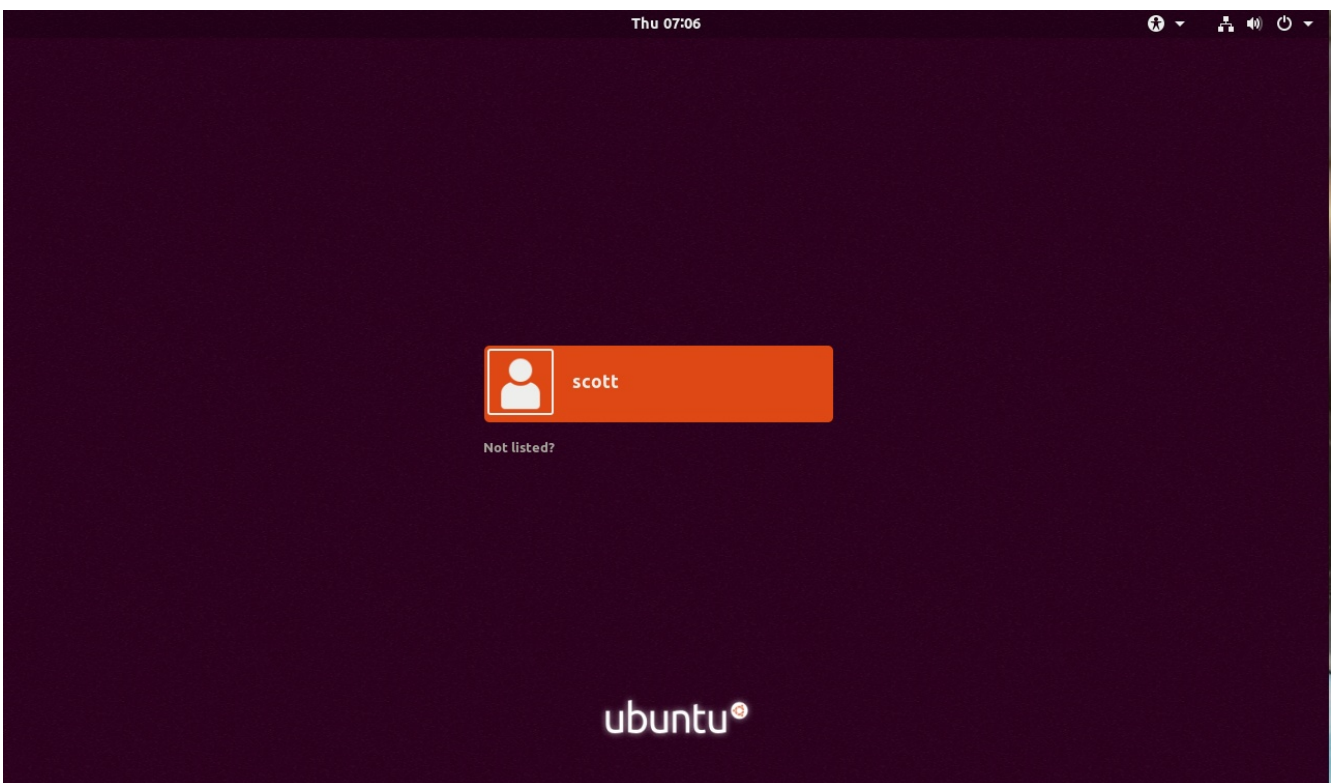
You will notice that you can resize your Ubuntu VM now that guest additions are installed.



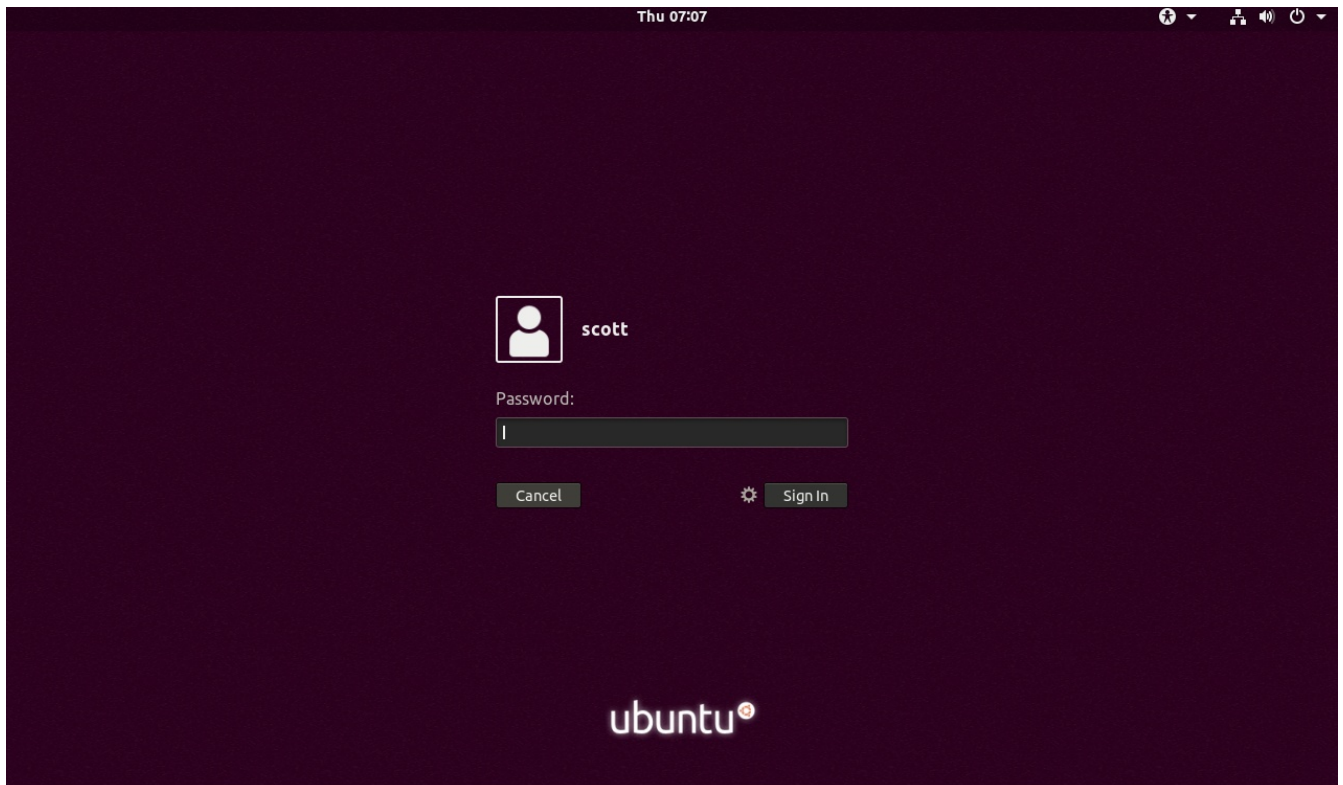
Click the power button in the upper right and reboot your VM.



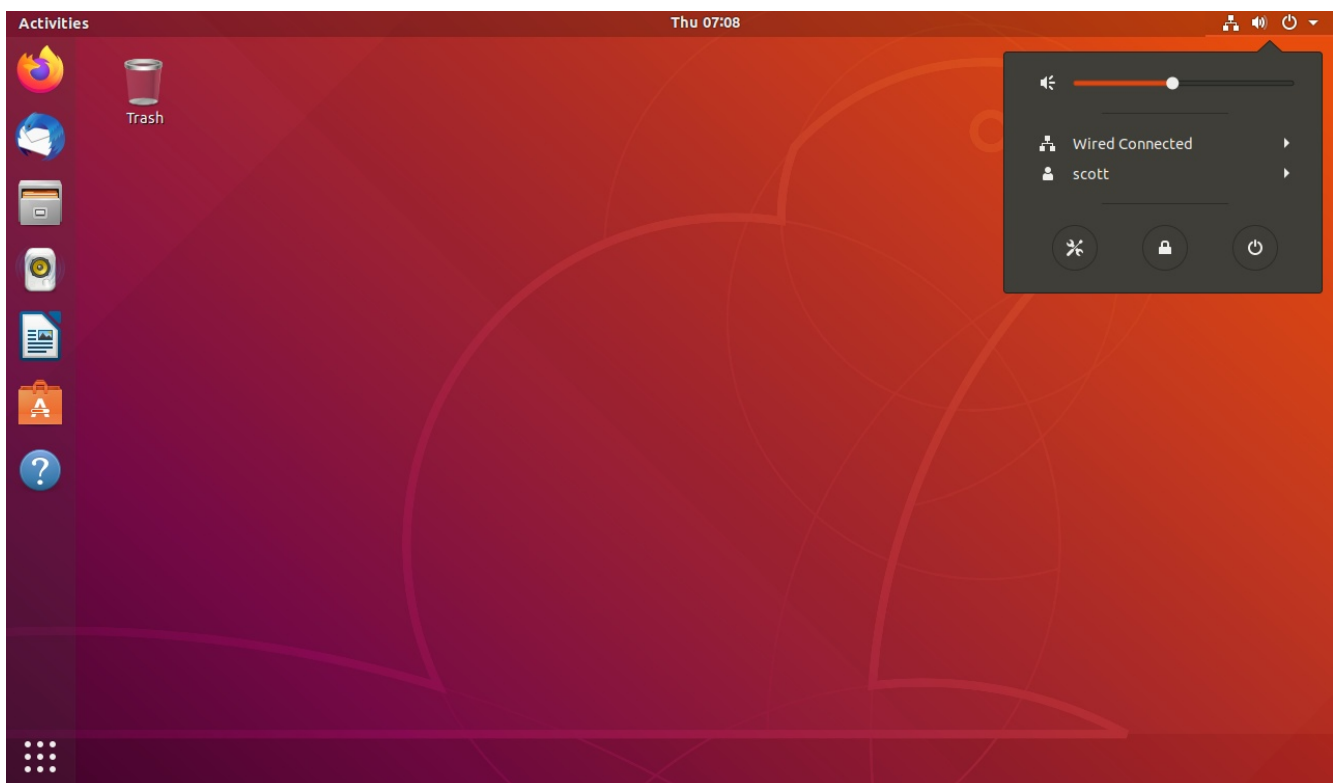
The system reboots



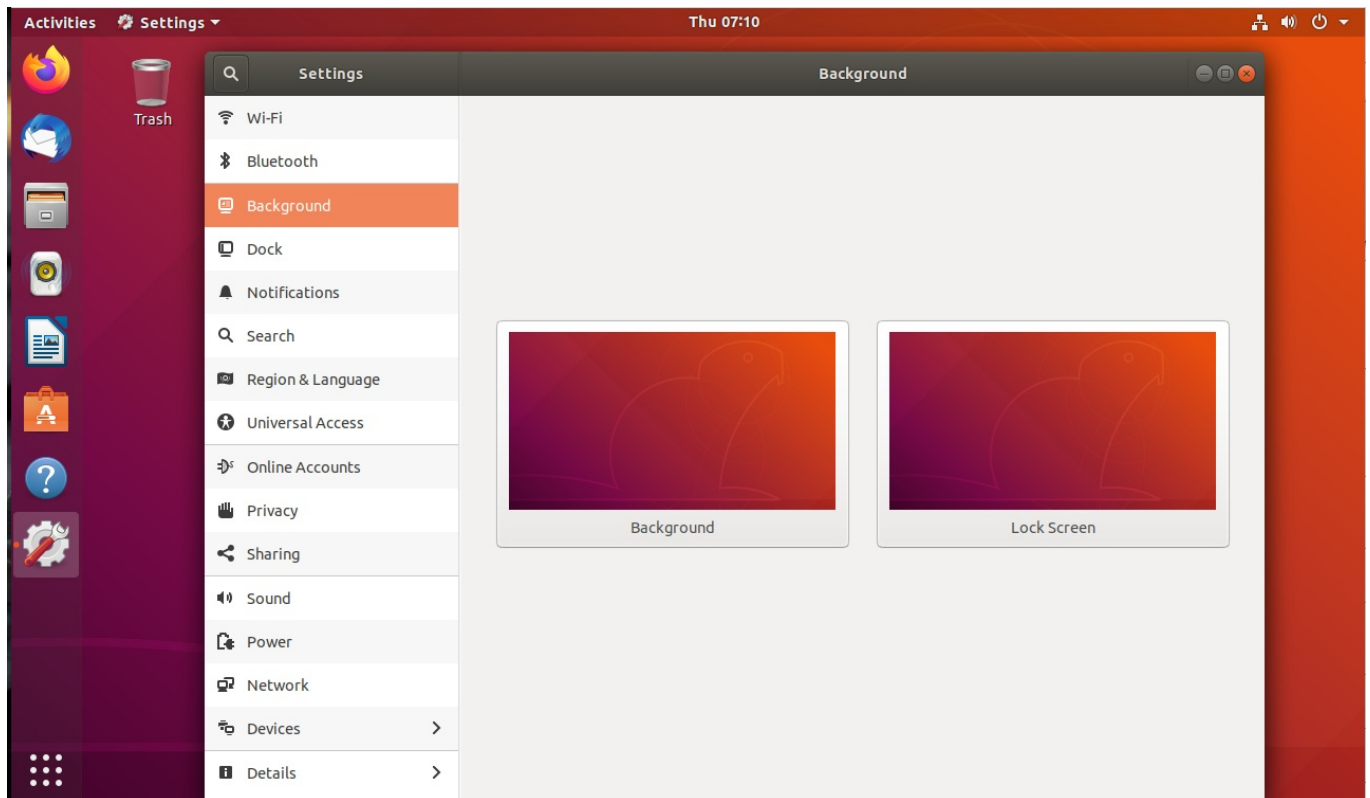
Click on your username



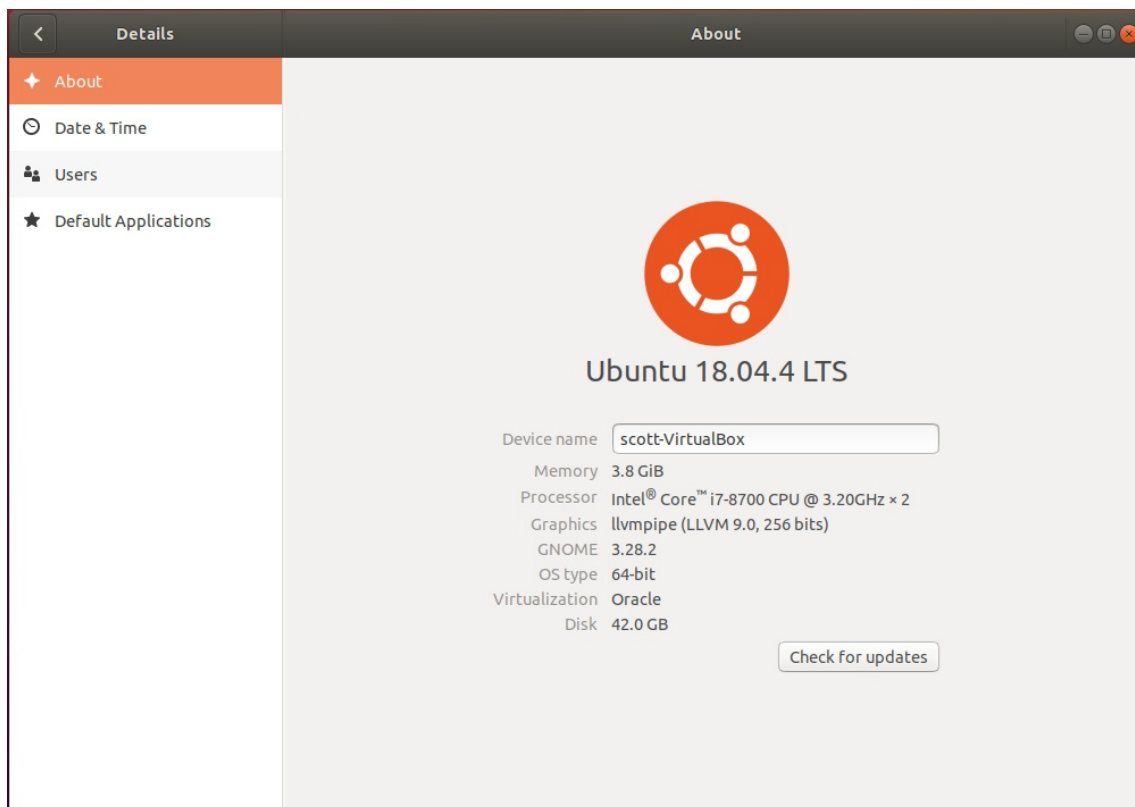
Enter your password



Click the upper right again. This time choose the wrench/screw driver icon.

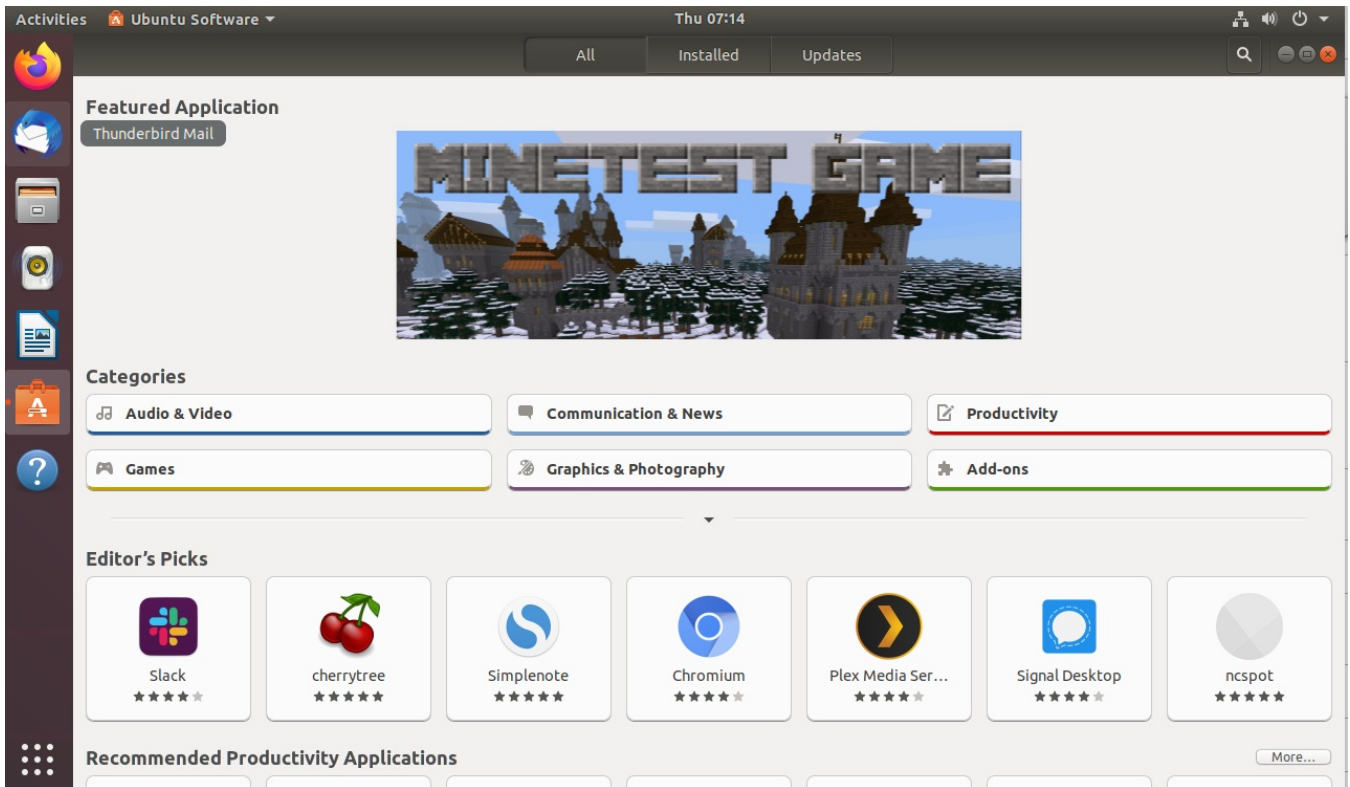


The system settings screen appears. This is where you can change many of the system settings in Ubuntu. Click "Details"

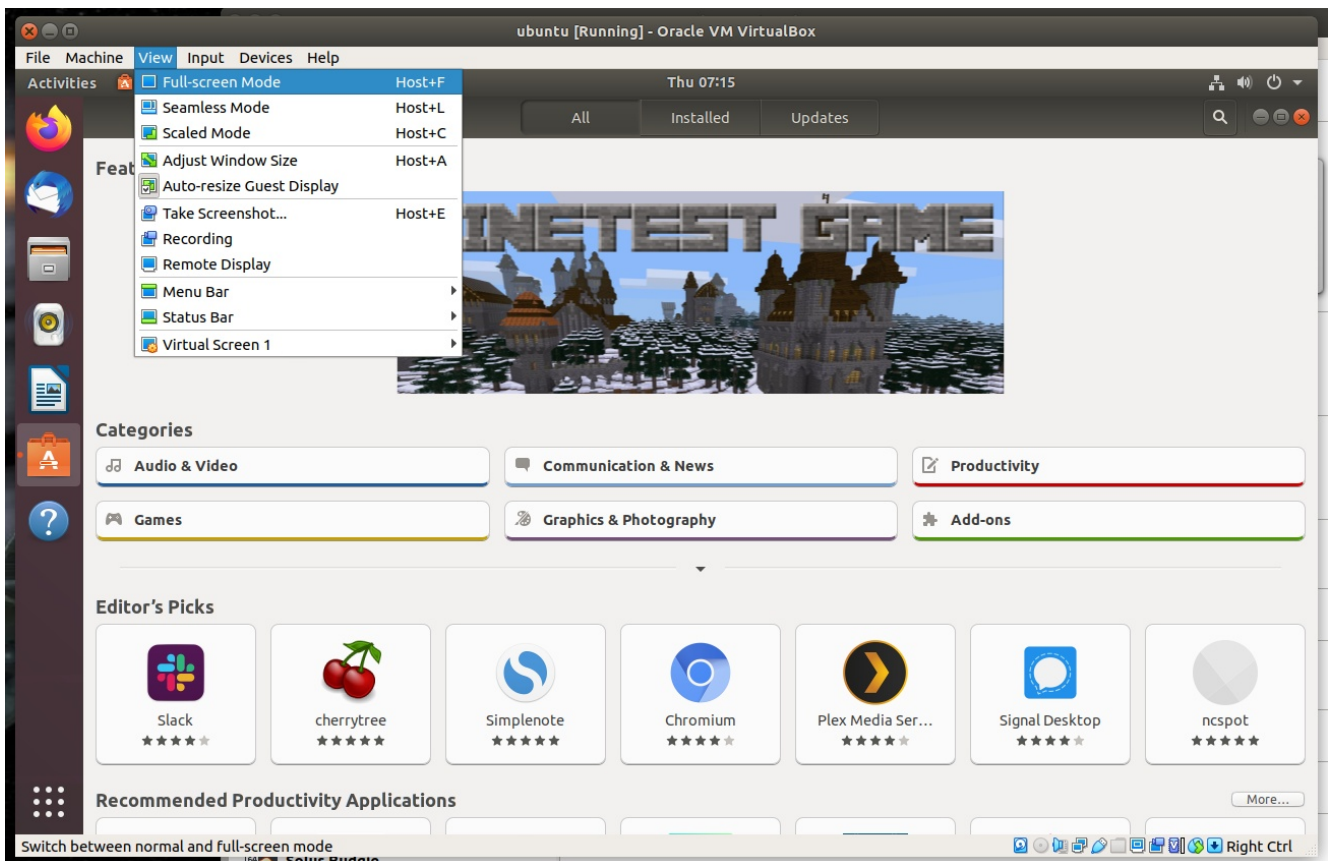


This will show the basic VM stats.

The bar on the left of the screen is called the "Unity menu". Click on the orange suitcase looking icon. This is the "Store".



You can install thousands of applications from here.



Click on "View" and then "Full Screen Mode" as in the screen shot above. This will make your VM full screen instead of in a window. If you move your cursor to the bottom middle of the screen, you will notice a slide up window that will let you get out of full screen mode. Shutting the VM down will also exit the window, of course.

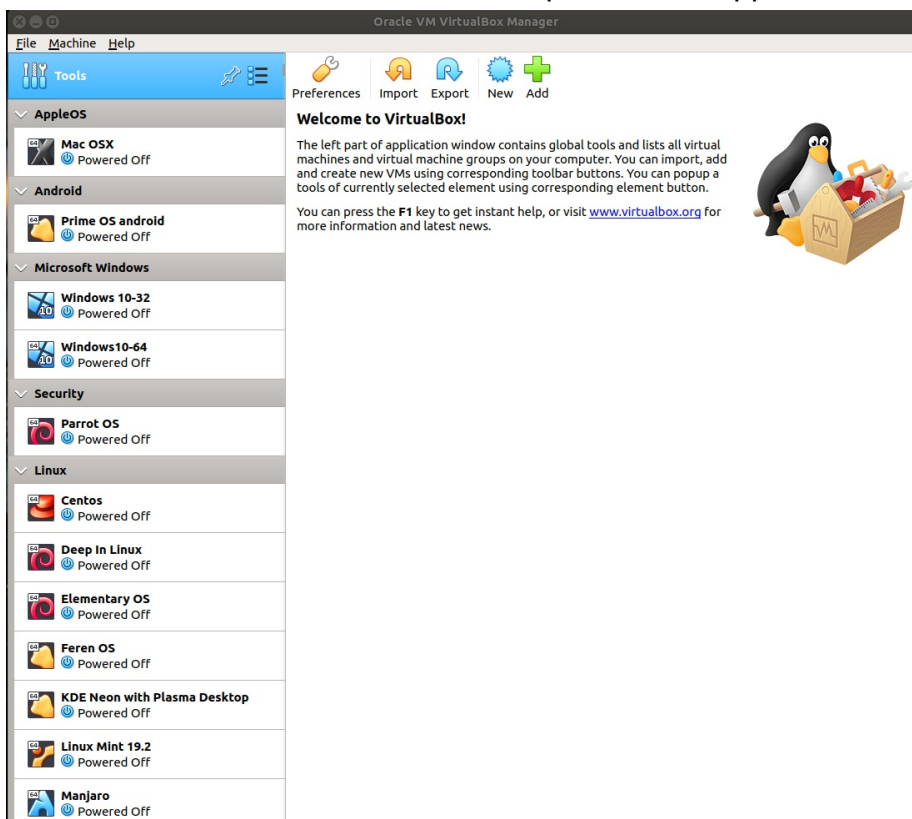
The nine dot icon in the lower left of the screen on the Unity menu is the "App Drawer". Click it to see apps on the system.



You can also change the context of the app drawer from frequently used applications to ALL apps. You can also type in the search field at the top of the app drawer to find an application or a link to install an application that you do not have loaded.

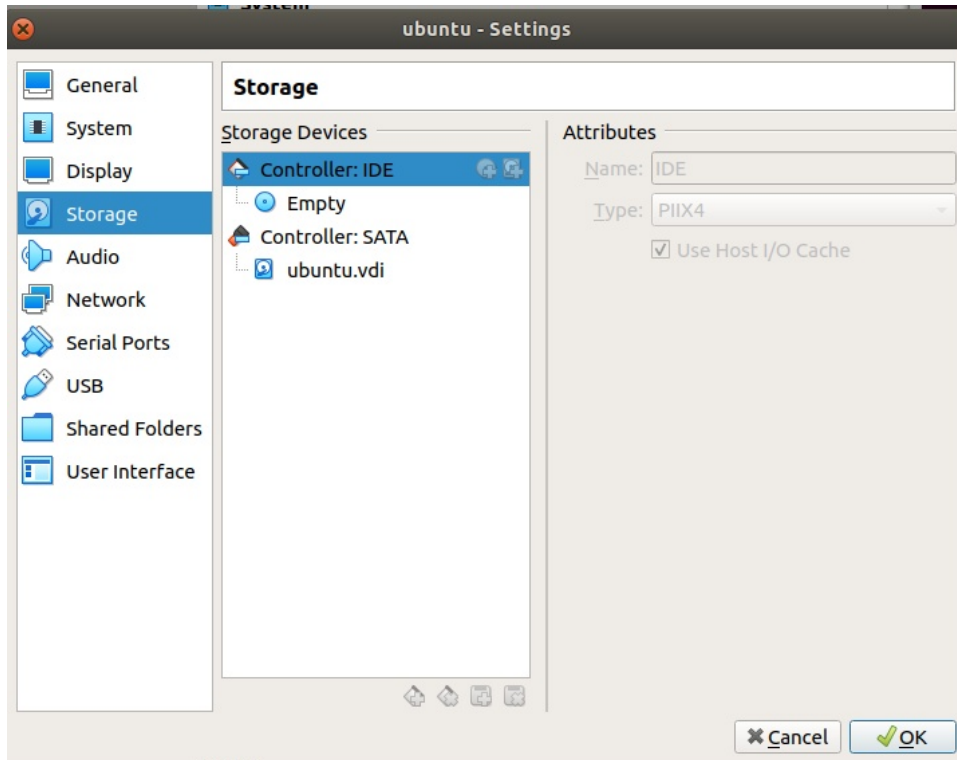
Since this blog was about installing virtual box and a test VM, I will leave you to playing with your Ubuntu VM. Be sure to perform an orderly shutdown of your Ubuntu from the menu as opposed to Xing the window which is like hitting the power button. That's never a good thing to do on a physical computer either.

I have a lot of virtual machines on my VirtualBox. Here is what my menu looks like. You will notice below that I have created "Groups". I have Apple, Android, Windows, and Linux.

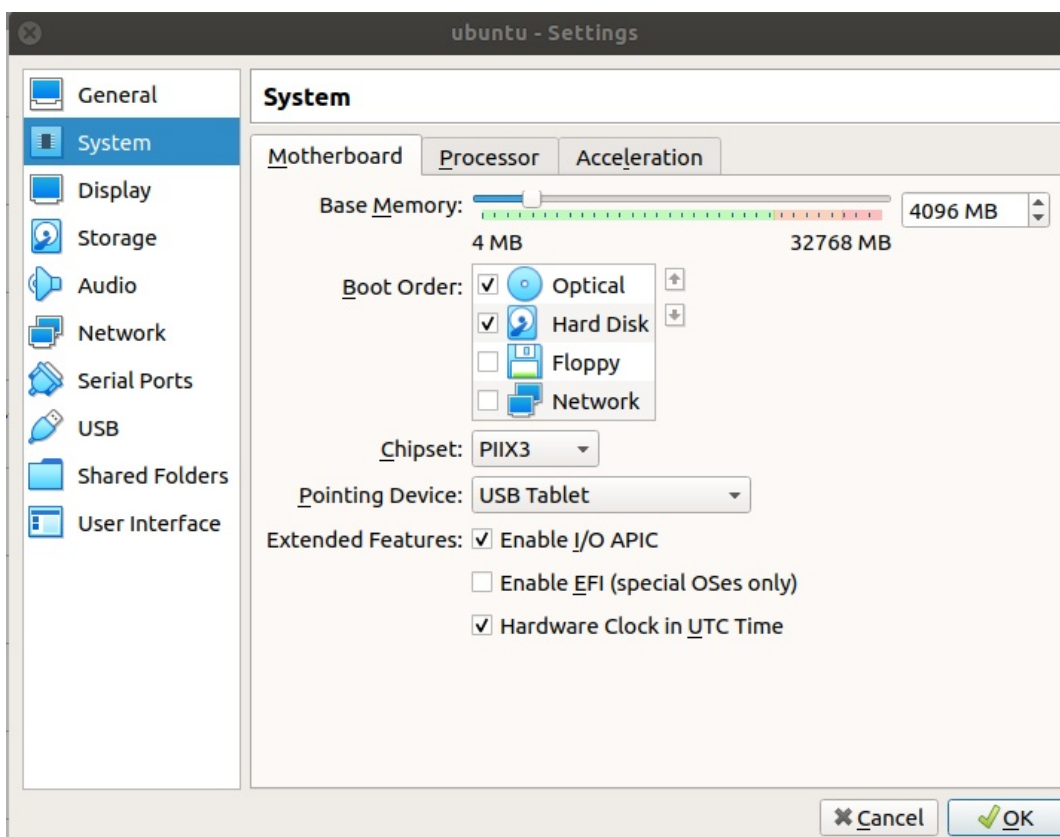


So far we learned how to create a guest Ubuntu virtual machine instance on the Virtual Box hypervisor. We learned that Virtual Box is a type 2 hypervisor because Virtual Box is an application running under the hosted operating system. In our installation, Windows 10 was the hosting operating system.

Recall that we created a virtual disk named "ubuntu.vdi" which is a single file on the host operating system that emulates a hardware hard drive.



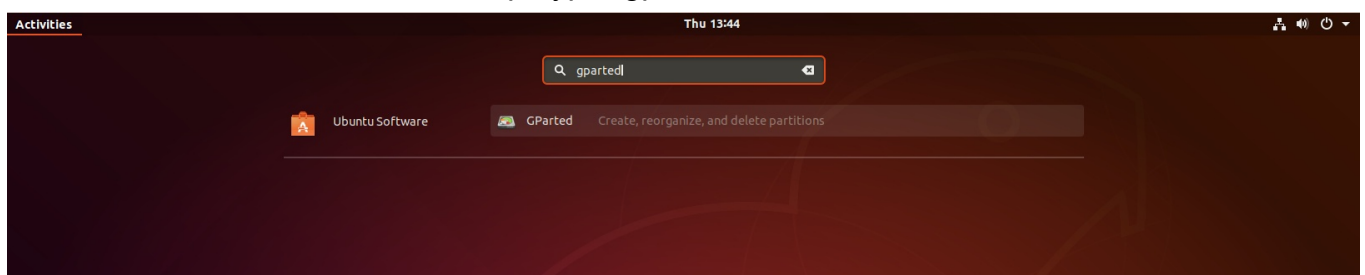
In the "settings" of the Ubuntu Virtual Machine guest, we defined motherboard settings as follows.



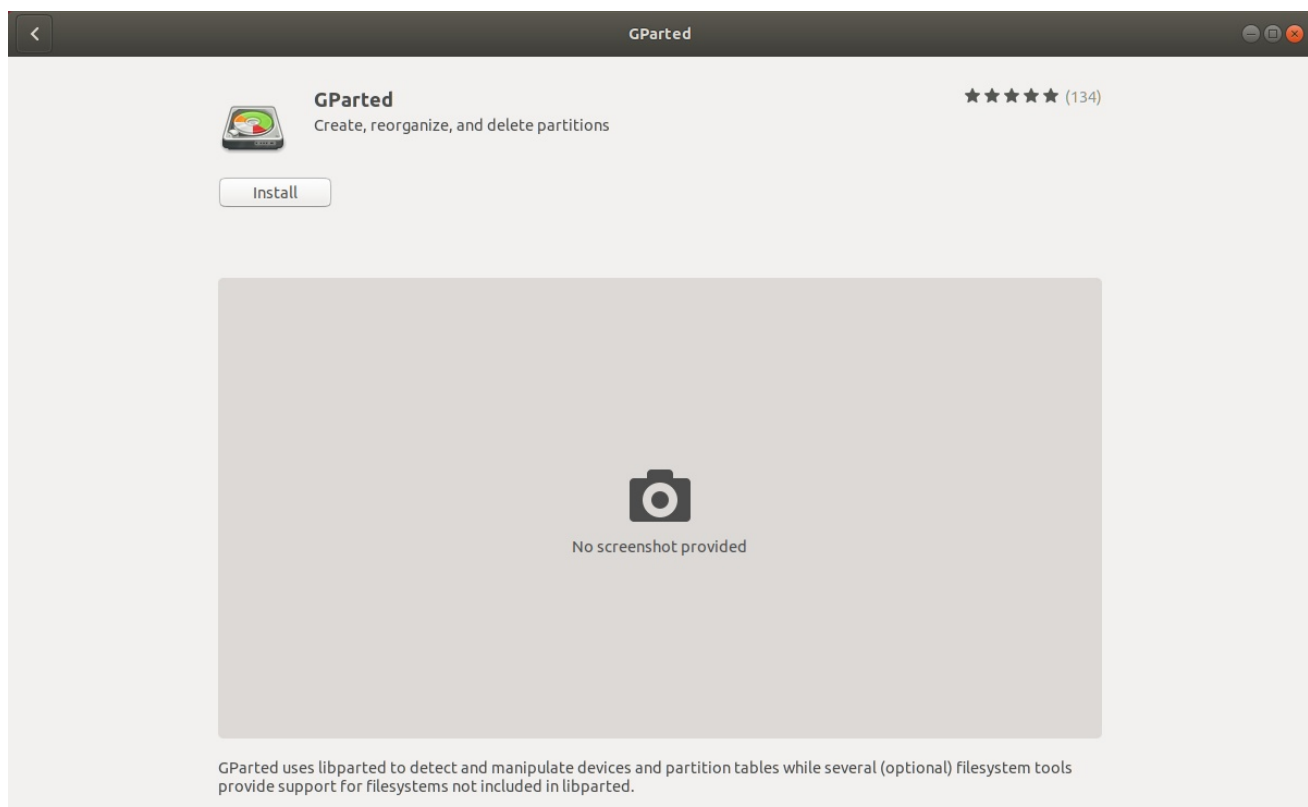
Notice that the "Enable EFI (special OSes only) checkbox was unchecked. Since this was the case, we are emulating a BIOS based motherboard with a Master Boot Record (MBR) formatted partition table. The drive has a single EXT-4 partition in which all data is stored. An MBR is a special boot sector at the very beginning of the disk that holds the information on how the logical partitions containing the file systems are organized on the drive.

The MBR also contains executable code that functions as a boot loader for the installed operating system. The organization of the partition table in the MBR limits the maximum addressable storage space of a partitioned disk to 2 TiB. The only way that an MBR formatted disk can contain more data is if the logical block size is increased. For that reason, most newer systems do not use MBR formatted partition tables.

There is a program called the Gnome Partition Editor (GParted) which can examine and manipulate disk partitions. Boot up your Ubuntu Virtual Machine, log in, and open the app drawer. In the search field at the top, type "gparted".



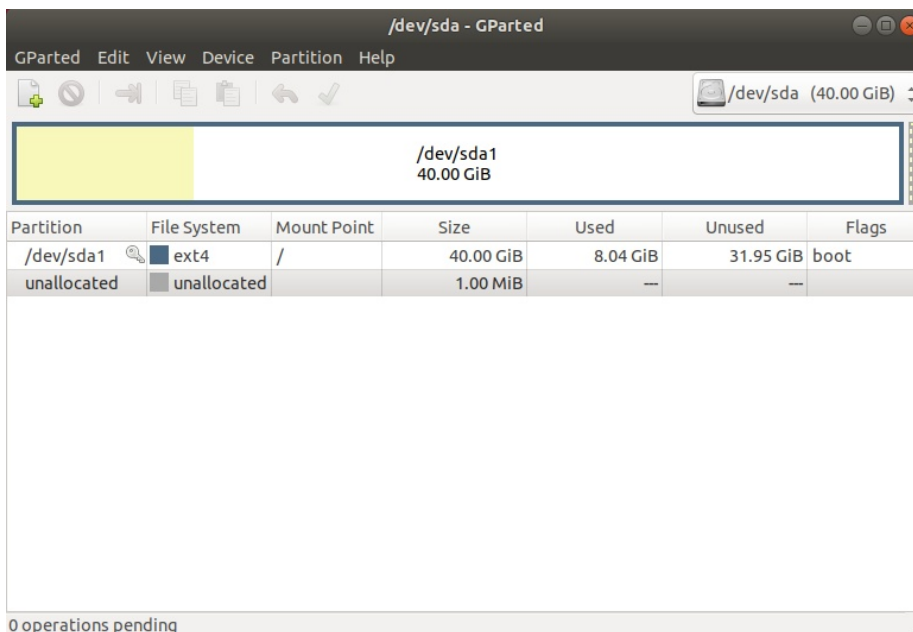
You will note that gparted is listed as available from the Ubuntu Software store, but is not presently installed. Click on "gparted" above to install it.



You will be asked for your password, and then "gparted" will be installed. You have an option to launch it from the store, but exit the store instead. Open the app drawer again.



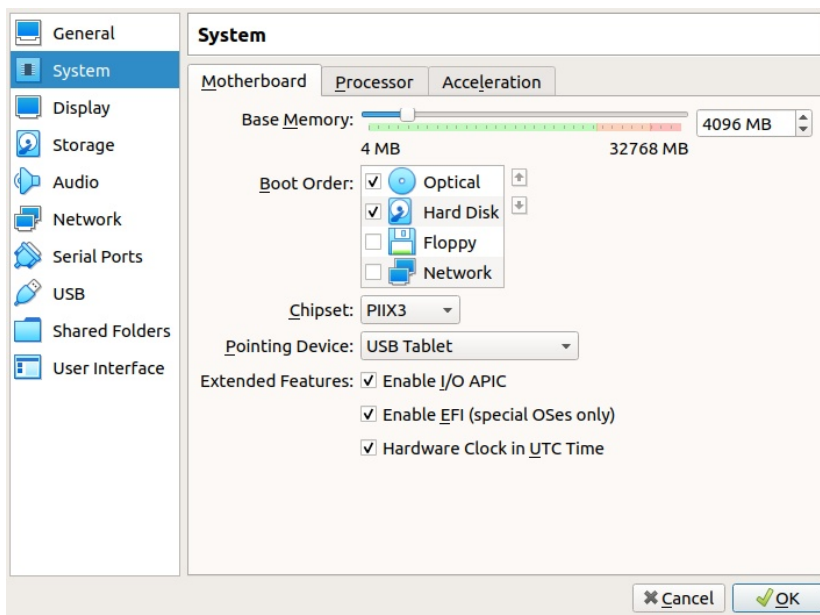
If the app drawer is set to show all programs, Gparted will be listed. Click on it to launch it. It will ask for your password.



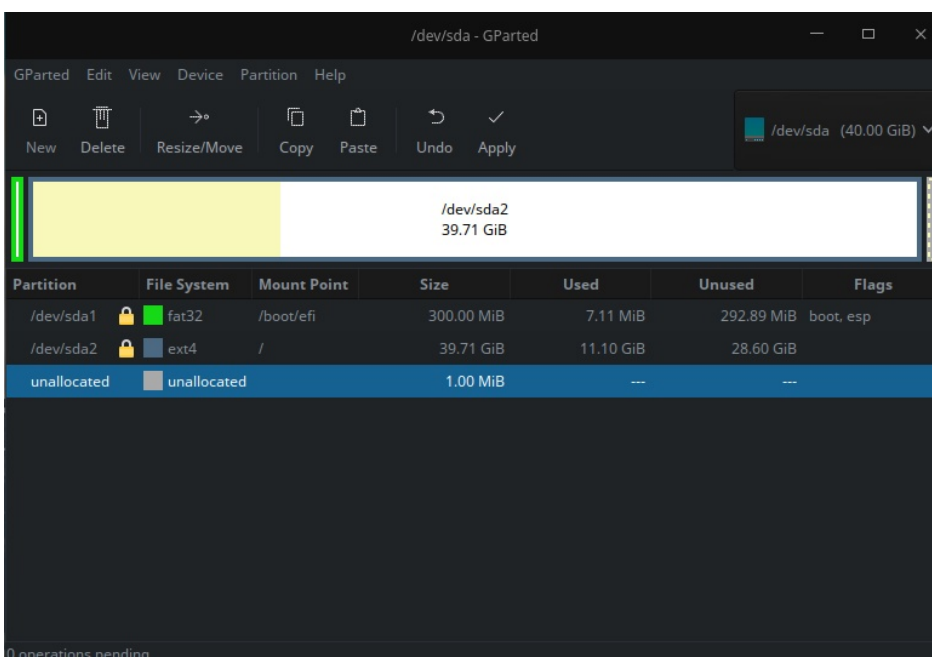
You can see in "gparted" above that there is a 40GB partition formatted ext4 (the most common linux file format). Note that drives in linux have names like /dev/sda, /dev/sdb, /dev/sdc and so on. The first partition on a drive would be sda1, then sda2, and so on. Everything in Linux is a file. That includes disk drives. Windows gives drives letters like C, D, and E. That is inefficient since you run out of letters in that scheme pretty quickly on a large system. The yellow highlight above shows the used space. We have 8GB used on our 40GB drive partition.

if we had checked the "Enable EFI" box, we would have booted the initial installation CD Rom through the Extensible Firmware Interface (EFI) found on all newer computers. Had this been the case, we would have been emulating an EFI based firmware. This would have triggered the operating system installation to create a Globally Unique Identifier (GUID) formatted

partition table. A GUID partition table (GPT) is a newer standard than MBR. All modern computers support GPT. GPT's use logical block addressing on hard drives like MBR, but it uses 64 bits for logical block addresses, allowing a maximum disk size of 2^{64} sectors. For disks with 512 byte sectors the maximum size is 9.4 ZB (9.4×10^{23}) as opposed to the 2TiB for 512 byte sectors that MBR can support.



I had previously built an Ubuntu that used EFI. Because the "Enable EFI" checkbox is checked above, this makes the emulated system an EFI based system and so we will have a GPT partition table.



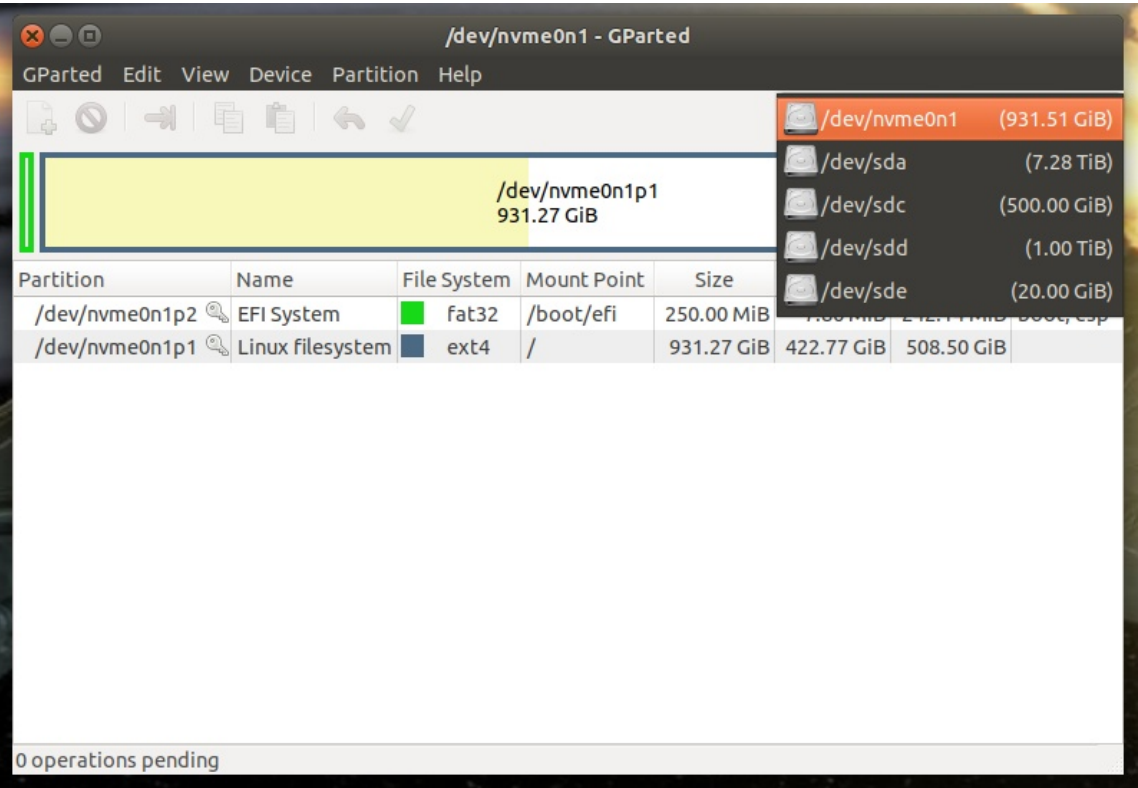
In the screenshot above, you can see an example of a GPT formatted drive. Note that there is a /dev/sda1 partition that is formatted "fat32", followed by the /dev/sda2 drive which is formatted ext4 and contains the operating system. GPT formatted systems have a boot partition, rather than a boot block. The boot partition can actually boot several operating systems from multiple partitions on the same drive. I have a desktop computer that dual boots Windows 10 and Ubuntu 18.04. The efi boot partition contains the instructions to boot each installed operating system from the partition on the drive on which it was installed. MBR formatted disks can only boot a single operating system. I do not generally create VMs with GPT partitions, because I do not generally need a VM to host more than one operating

system or a virtual drive larger than 2TB.

I have created VMs with GPT formatted drives to test installation of a particular operating system with EFI. I do this as a test bench and so it is helpful at times.

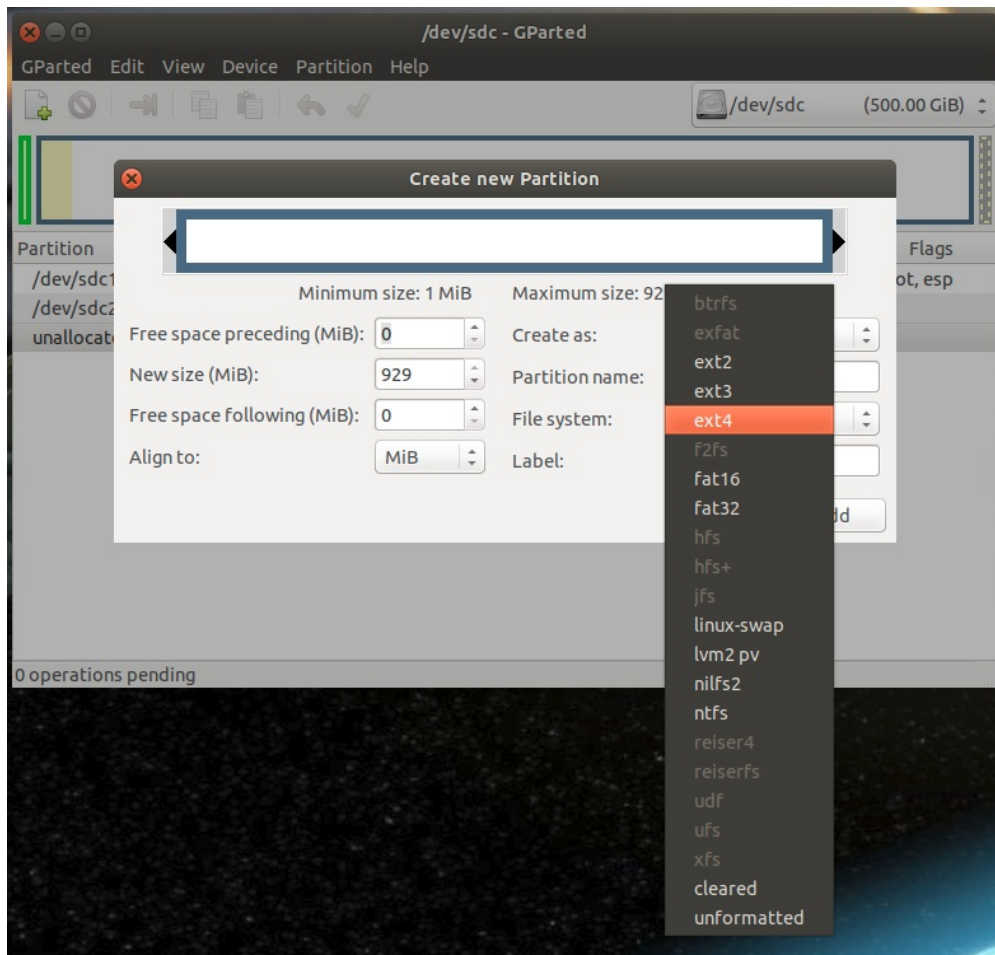
The lock next to the drive partition indicates that it is mounted and therefore locked against making changes to the partition while it is booted. To make changes to an existing partition, the technique is to boot a "Live CD". In the case of Ubuntu and many other Linux Distros, booting the installation CD and choosing the "Try out Ubuntu" option as opposed to the installation option lets you run the OS from a CD, thereby giving you the option to modify the partitions on the hard drive while the operating system is not running.

Once such example might be using Gparted to shrink an existing partition to make room for a new partition to add an OS to a drive. Gparted lets you resize, add, delete, move and copy partitions. Gparted can managed many disk drives.



The display above shows you Gparted on my desktop computer. Note sda, sdb, sdc, sdd, and sde are all separate drives. My main system drive is /dev/nvme0n1. My main drive is an NVME drive and so that is why the different nomenclature.

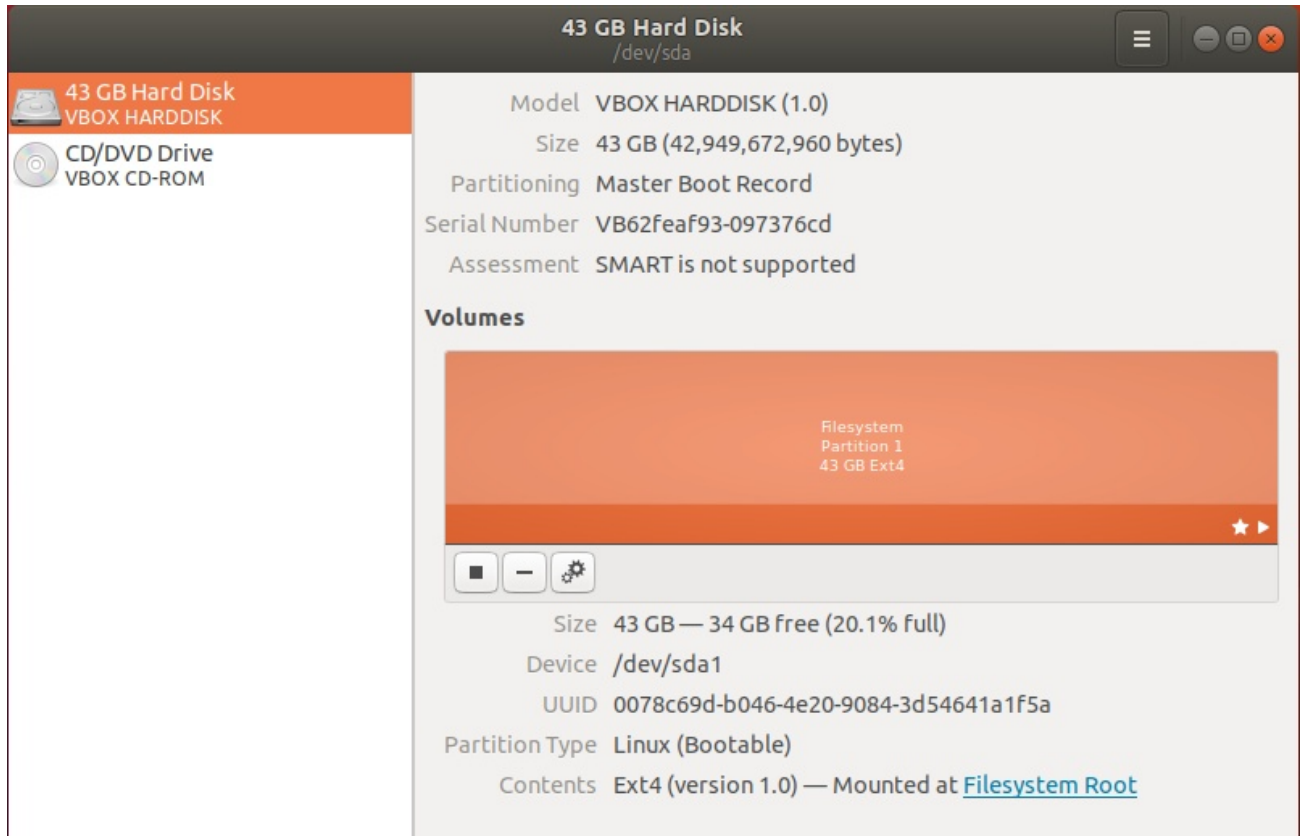
Gparted can be used to manage drive partitions from all operating systems including Windows and even Macintosh.



In the screen shot above, you can see how many formatted file system types are supported. In order to enable the ones that are grayed out, I only need to install a driver.



If you open the app drawer in Ubuntu and type "disks" in the search field, a program called "disks" will be displayed. Launch it.



"Disks" shows you all of the disks on your system and lets you do some rudimentary management of them, although not nearly as much as gparted can do.

I hope this update explaining MBR disks vs. GPT Disks and MBR booting vs. EFI booting has been helpful.

Understanding disk and partition structure is not only valuable in reference to the creation and management of virtual machines.

Disk management is critical to configuring bare metal desktops and servers as well.