Installing VirtualBox to run ROS

Wyatt Newman

6/26/16

ROS code can be computationally demanding, notably in running and displaying dynamic simulations. Good performance is best achieved with a relatively powerful, multi-core computer with compatible graphics co-processing (GPU). However, it can be convenient, e.g. for instructional purposes, to run ROS within an emulator. This document will assume use of VirtualBox (https://www.virtualbox.org/wiki/Downloads) to emulate the operating system Ubuntu 14.04 running ROS "Indigo". The example installation described here uses a Dell laptop running Windows 7. All of the software involved is freeware. The steps include:

1) Install VirtualBox
2) Create/configure a "virtual machine" within VirtualBox
3) Install Ubuntu 14.04 on this virtual machine
4) Install "Guest Additions" to improve the graphical display
5) Install ROS, Learning-ROS example code, and associated programming tools (e.g. git, git-gui, Netbeans) and set up ROS environment parameters
6) Compile and test the example code

One can then run ROS tools, such as Gazebo and Rviz, and run the example ROS code, and develop and run new ROS code.

Note that if you have Ubuntu installed on your computer (e.g., as the only operating system, or in a multi-boot set-up), you can skip steps 1 through 4; instead, proceed from step 5 to install ROS Indigo, CWRU ROS code repositories, and useful development tools.

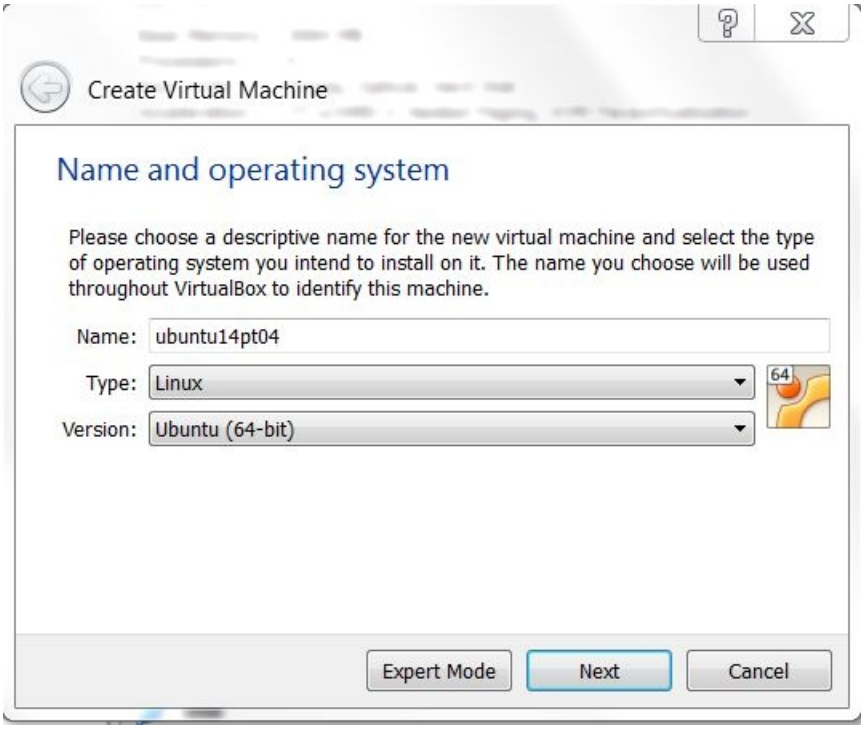
1) Install VirtualBox:

A manual for VirtualBox can be found at http://www.virtualbox.org/manual/, but the following steps should be adequate for quick set-up.

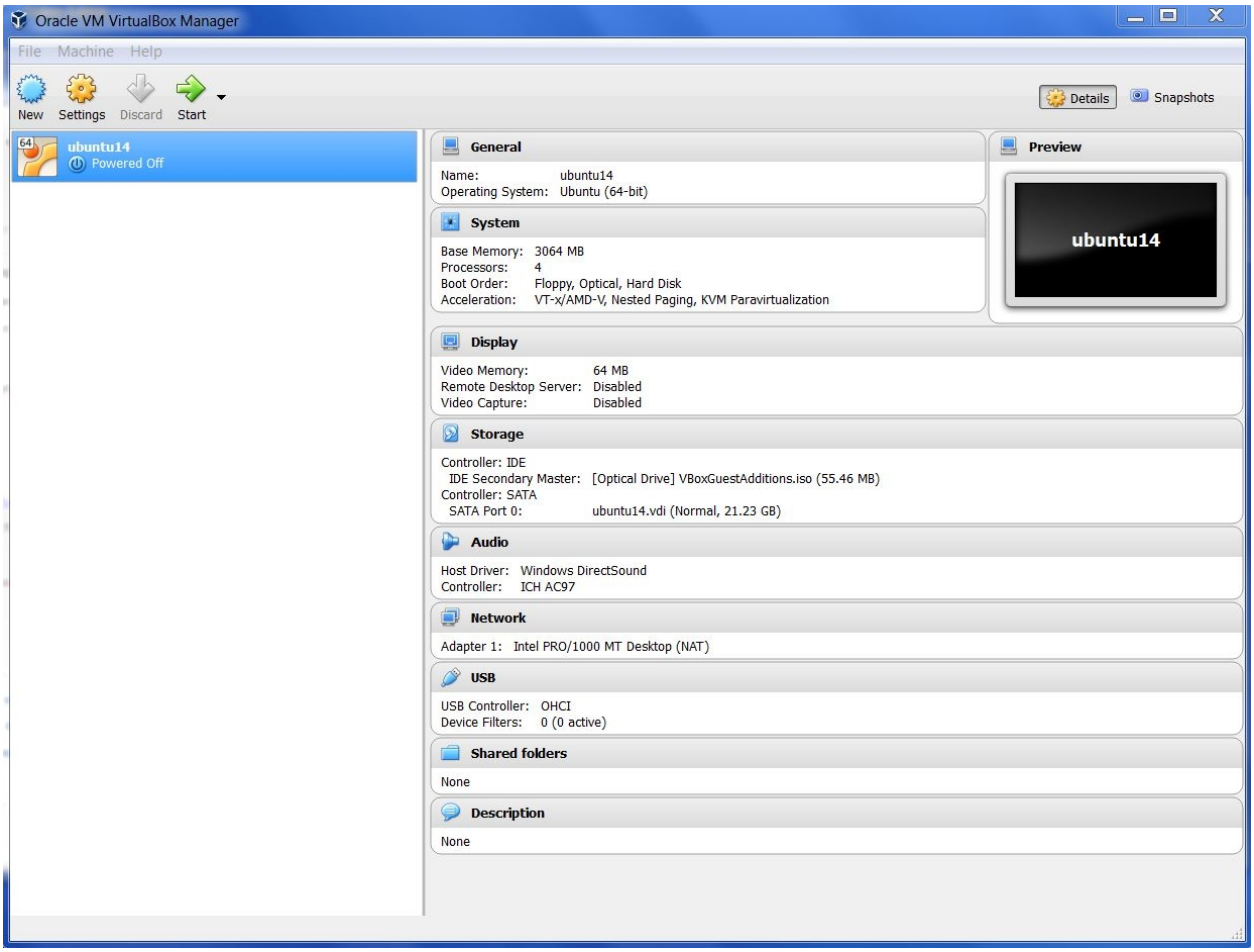
From the download page (https://www.virtualbox.org/wiki/Downloads), choose the version of VirtualBox appropriate for your OS. In the present example, this is **VirtualBox 5.0.22 for Windows hosts.** Download and run the executable, which will create the application "Oracle VM VirtualBox."

2) **Define/create a "virtual machine" in VirtualBox:**

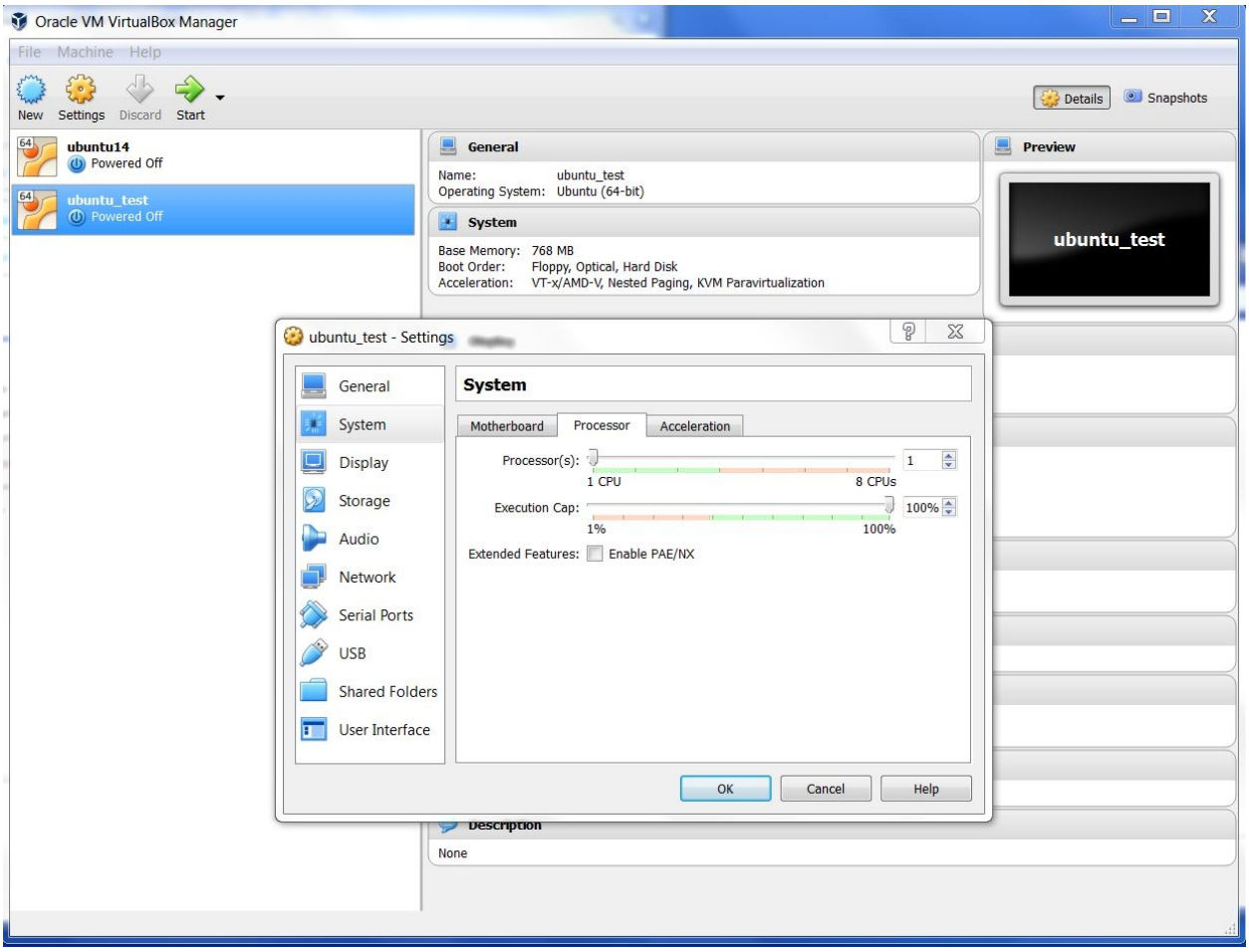
Start up “virtualBox” (which should appear as a new icon on your desktop), and within VirtualBox, click on the blue icon labeled “new.” This will initiate creation of a new virtual machine (VM) with your chosen OS and system settings. The figure below shows a screenshot of the VirtualBox prompt at this point (for which I have chosen a name for the new virtual machine and a choice of Linux OS (Ubuntu 64-bit).



You will be prompted to enter parameters for your new virtual machine. Anticipating use of ROS, I chose 3GB of RAM, 4 processor cores, 64MB of video memory and over 20GB of disk space. (Default options were accepted for all other prompts, except for choosing “fixed” disk size). When running your virtual machine, these resources will be unavailable to your "host” system (e.g. Windows). A display of my chosen parameters appears below.



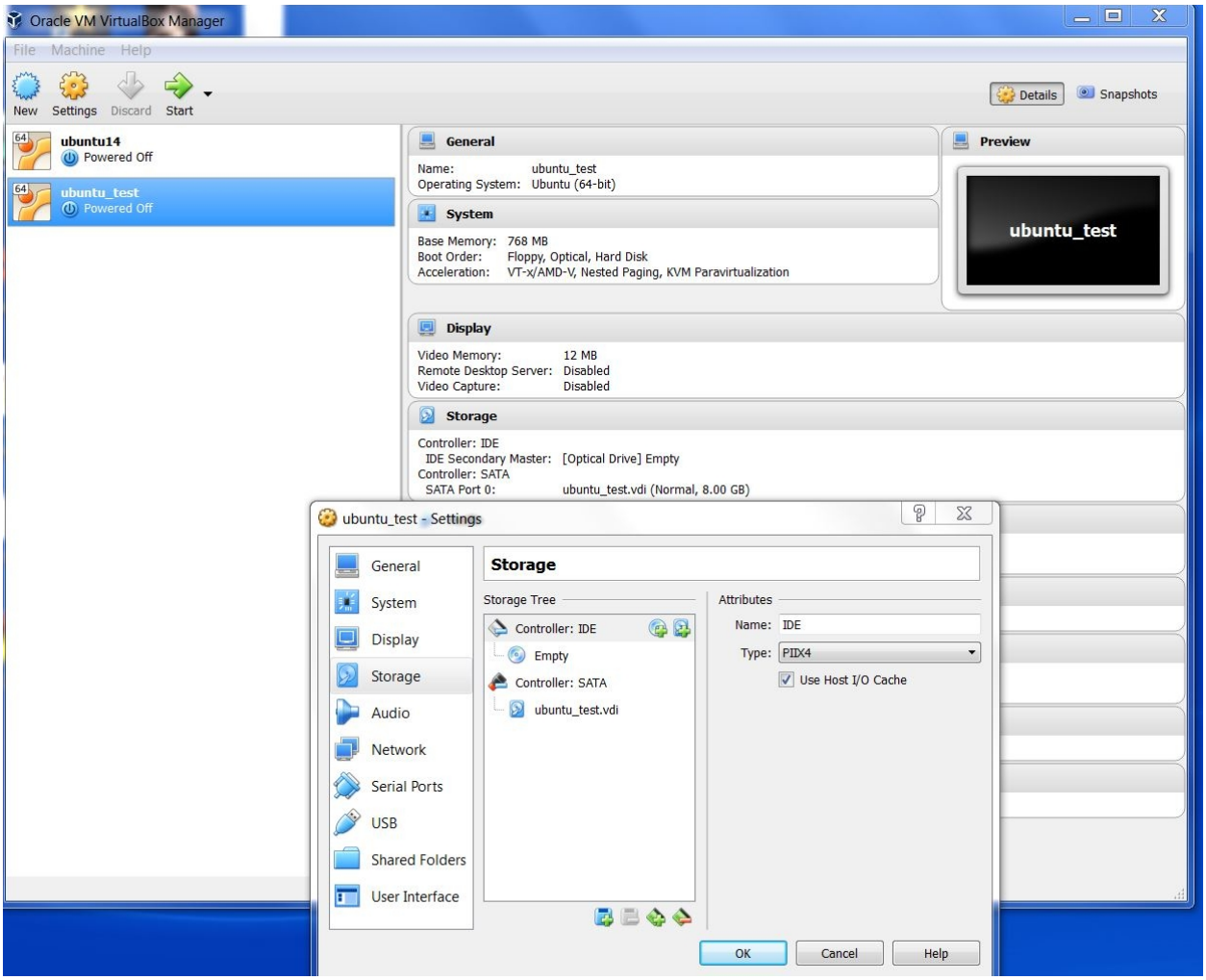
You will want more than one CPU core assigned to your virtual machine. To increase the default from 1, click the “system” item in VirtualBox, and select the “Processor” tab. The view will appear as below:



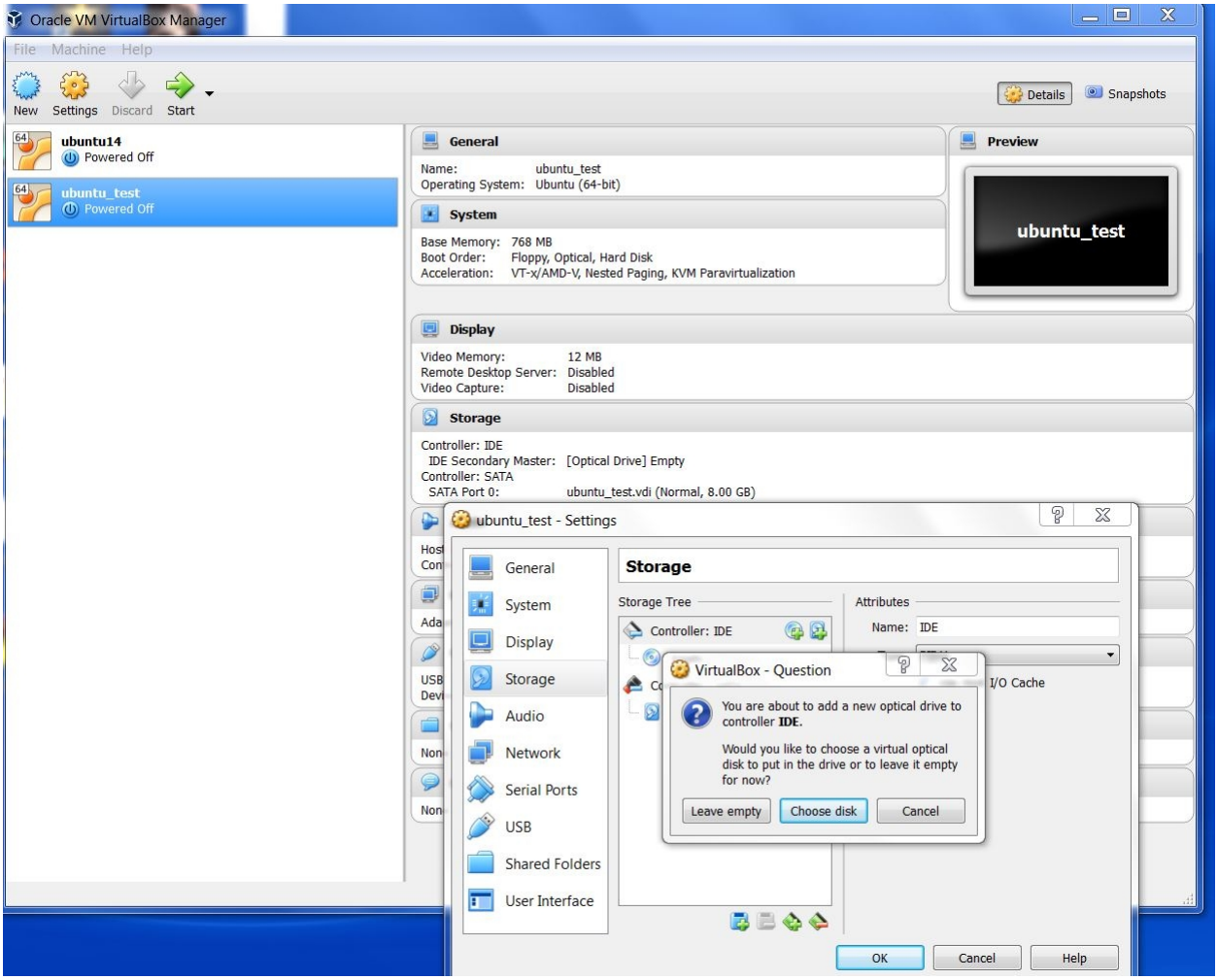
Use the slider to assign more cores. (I chose 3 cores in my set-up).

3) **Download and Install Ubuntu in the virtual machine:**
Download a copy of Ubuntu. For this example, I went to http://releases.ubuntu.com/14.04/ and selected Ubuntu-14.04.4-desktop-amd64.iso (Desktop image for 64-bit PC (AMD64) computers (standard download).

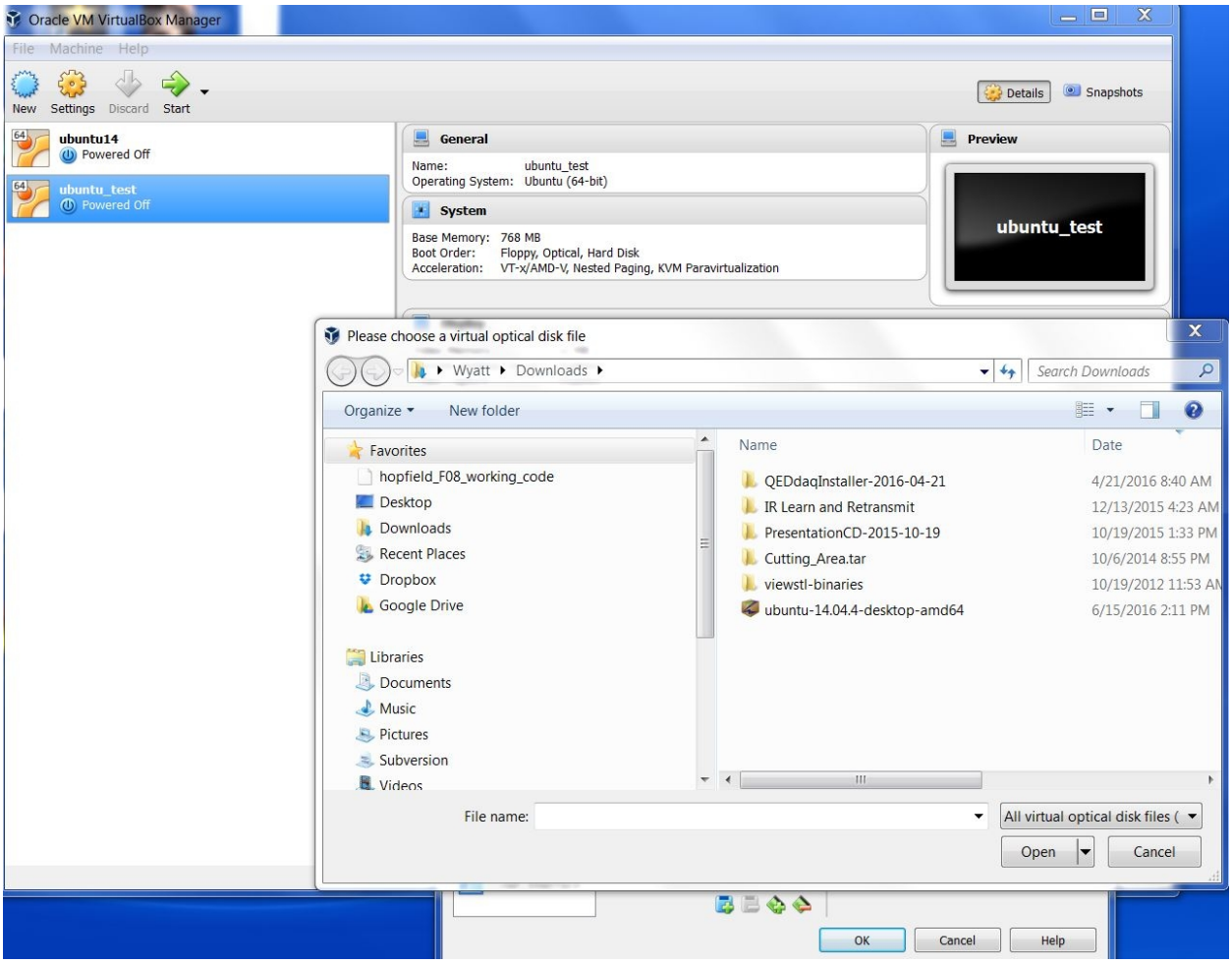
After downloading this ISO, attach this file as a "Virtual Optical Disk File", as follows. Within VirtualBox, in the left panel, you will see your newly-created virtual machine. Click on this icon to highlight it. This will result in a display of the VM properties (right side panel within VirtualBox). Click on "storage", which will bring up options shown below.



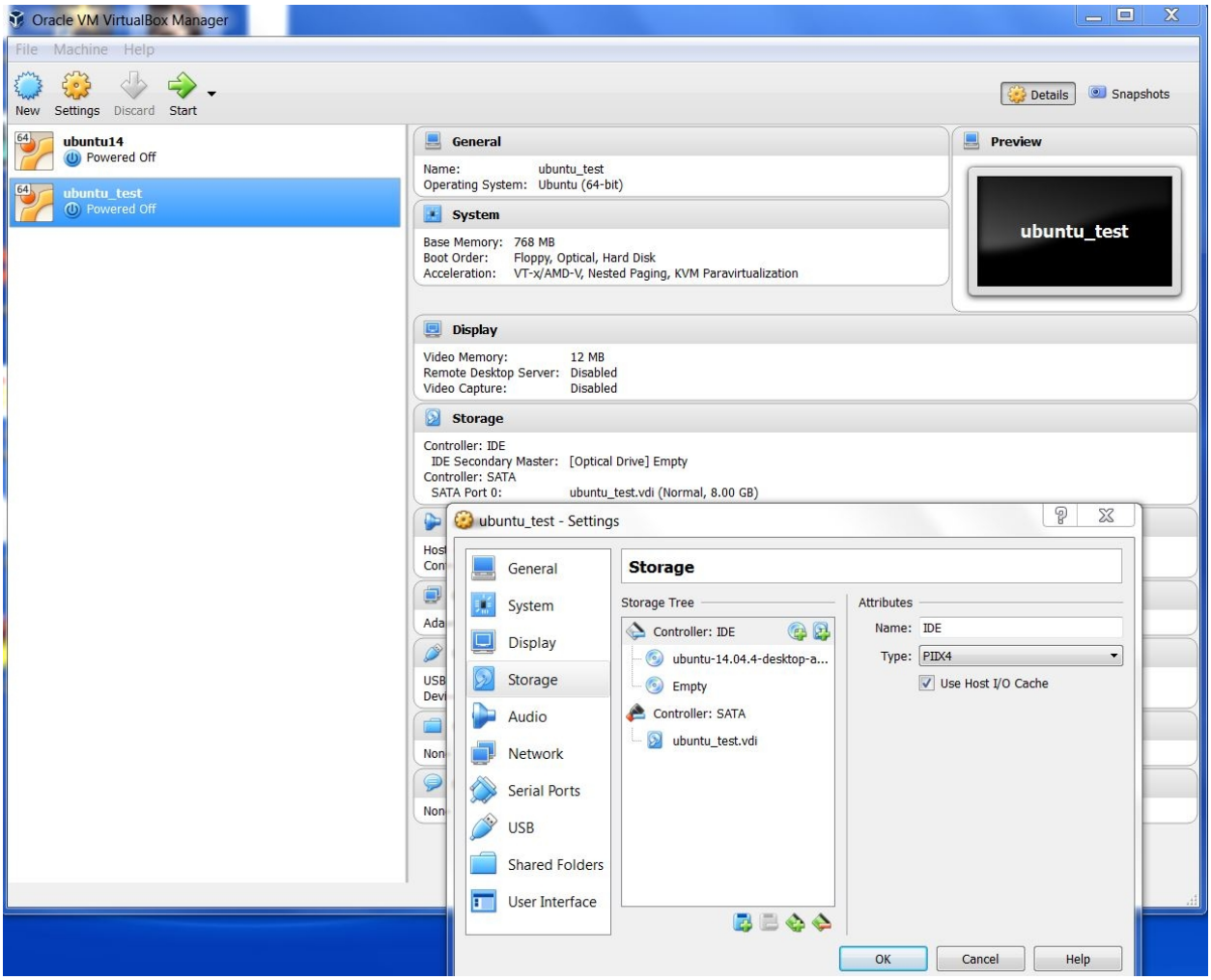
Under “controller IDE” (in my case), click the “adds optical drive” icon (the label will appear when your mouse hovers over this icon, next to “Controller:IDE”. Clicking this icon will bring up a prompt to choose your virtual optical disc, as shown below:



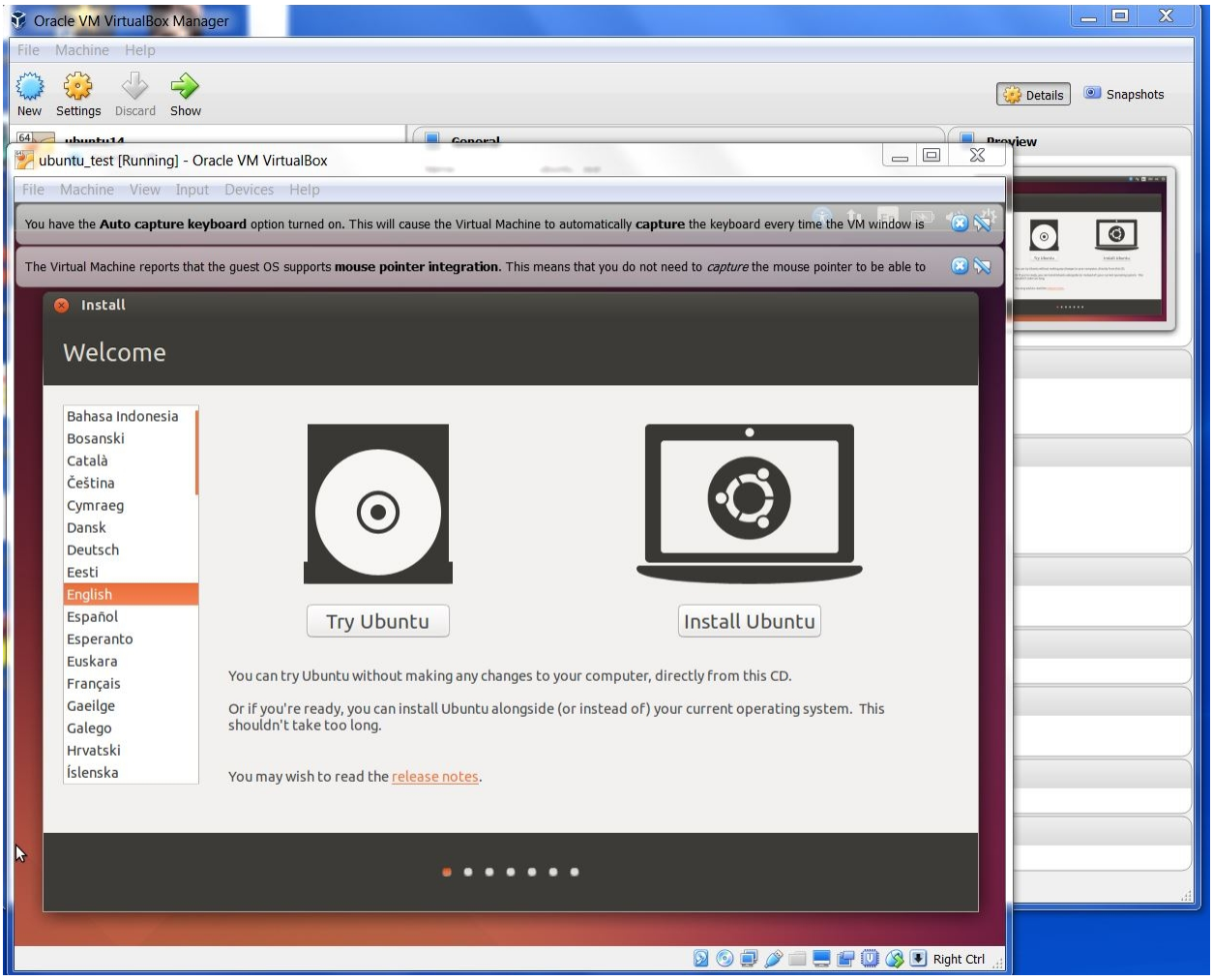
Click “Choose disk”, then navigate to your downloaded Ubuntu ISO file (as shown below).



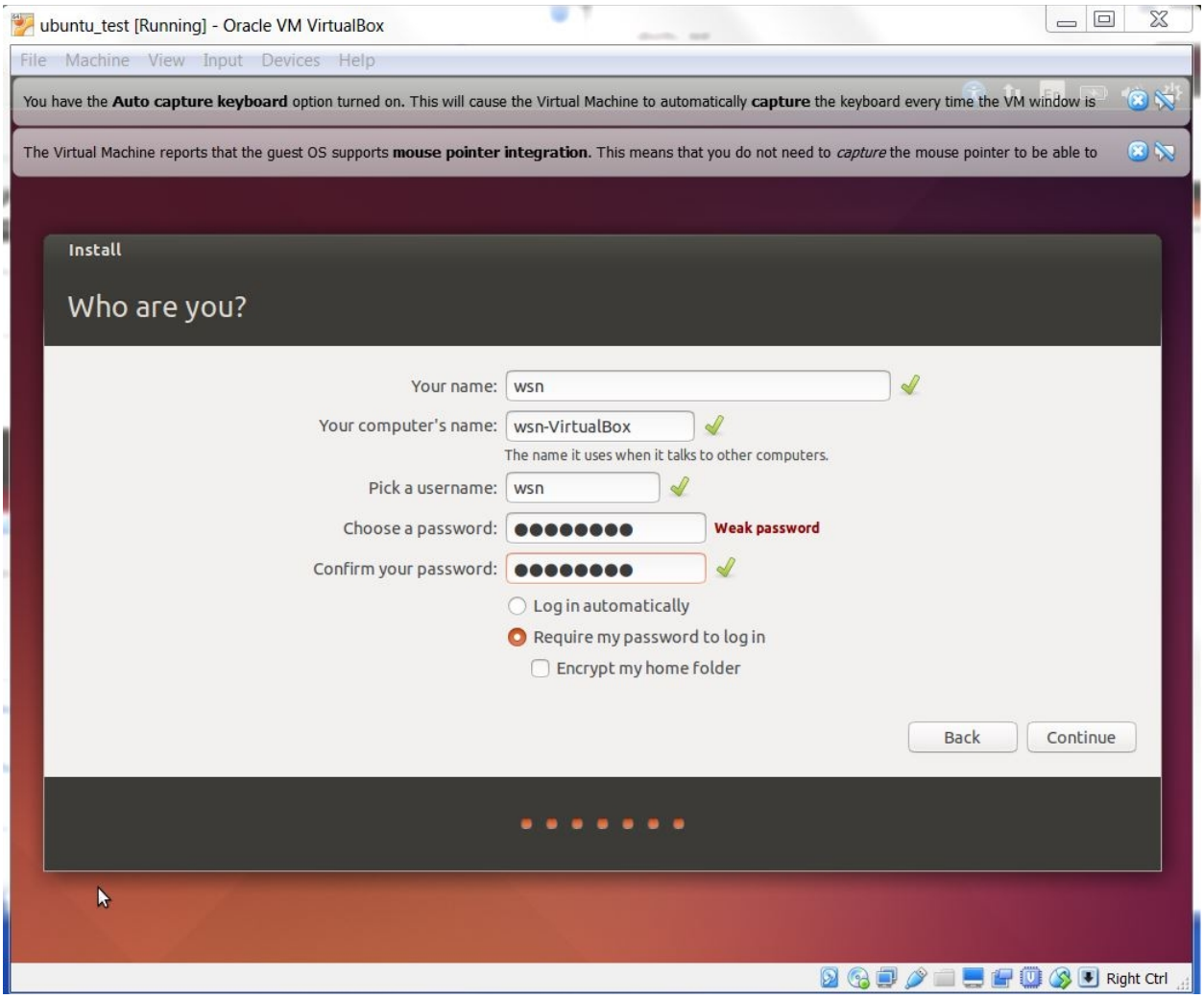
After double-clicking this installation file, it will be shown as being attached, as in the figure below (note Ubuntu-14.04-desktop under controller: IDE).



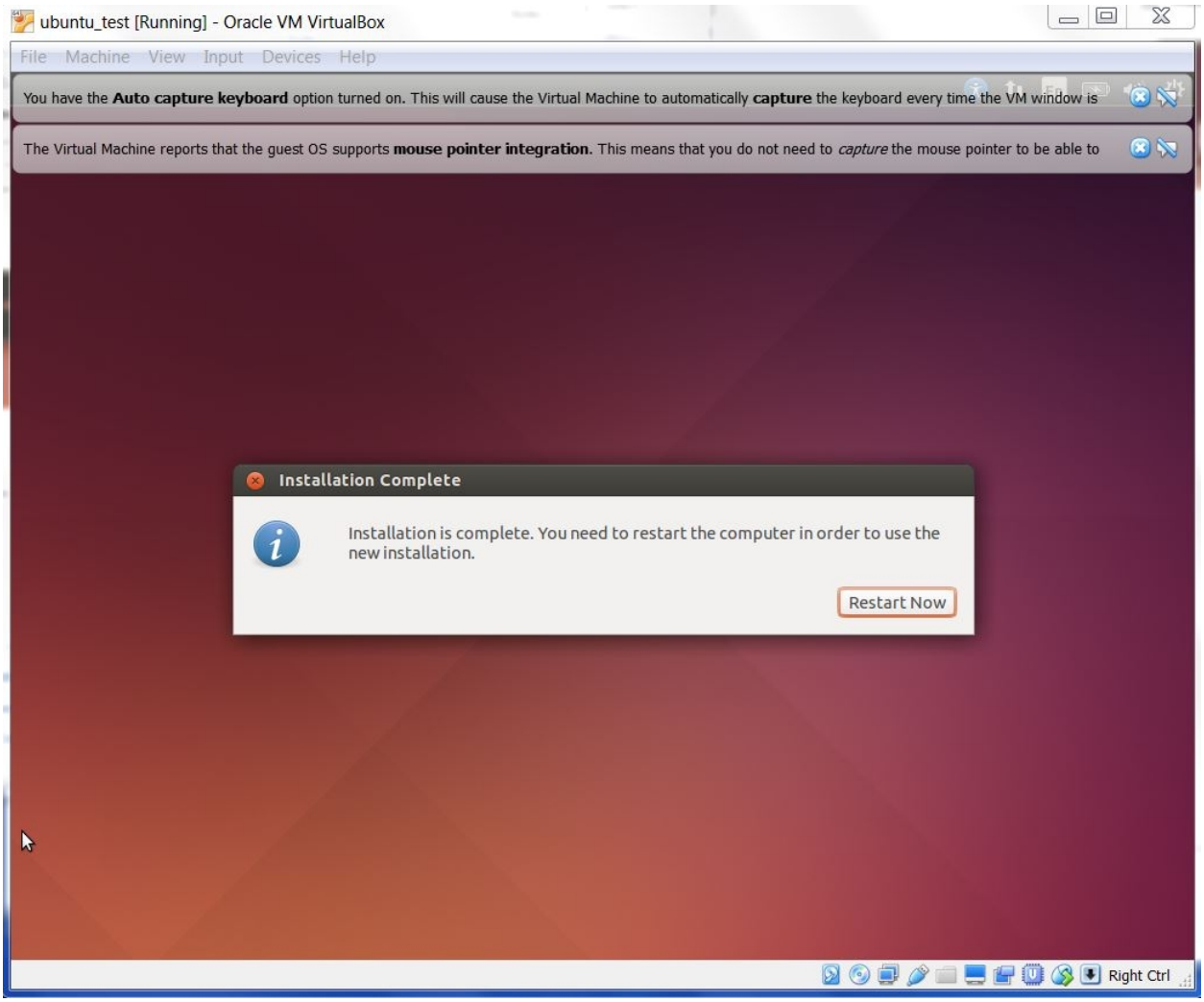
Now, start up the new VM by double-clicking its icon in VirtualBox. After a brief wait, the virtual machine's window will look like the scene below:



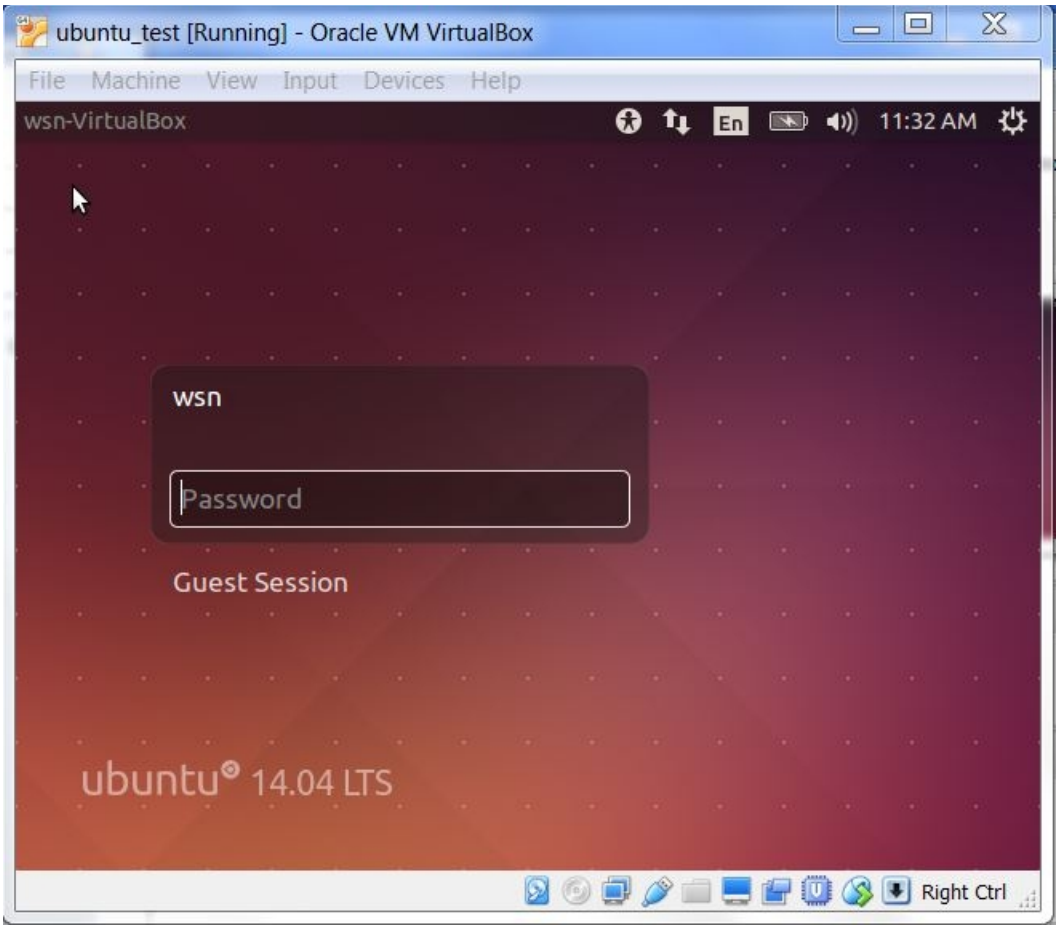
Click on “Install Ubuntu.” You will be presented with a sequence of set-up pages. Accepting the defaults is fine, although you will need to choose a user name and password, e.g. as below:



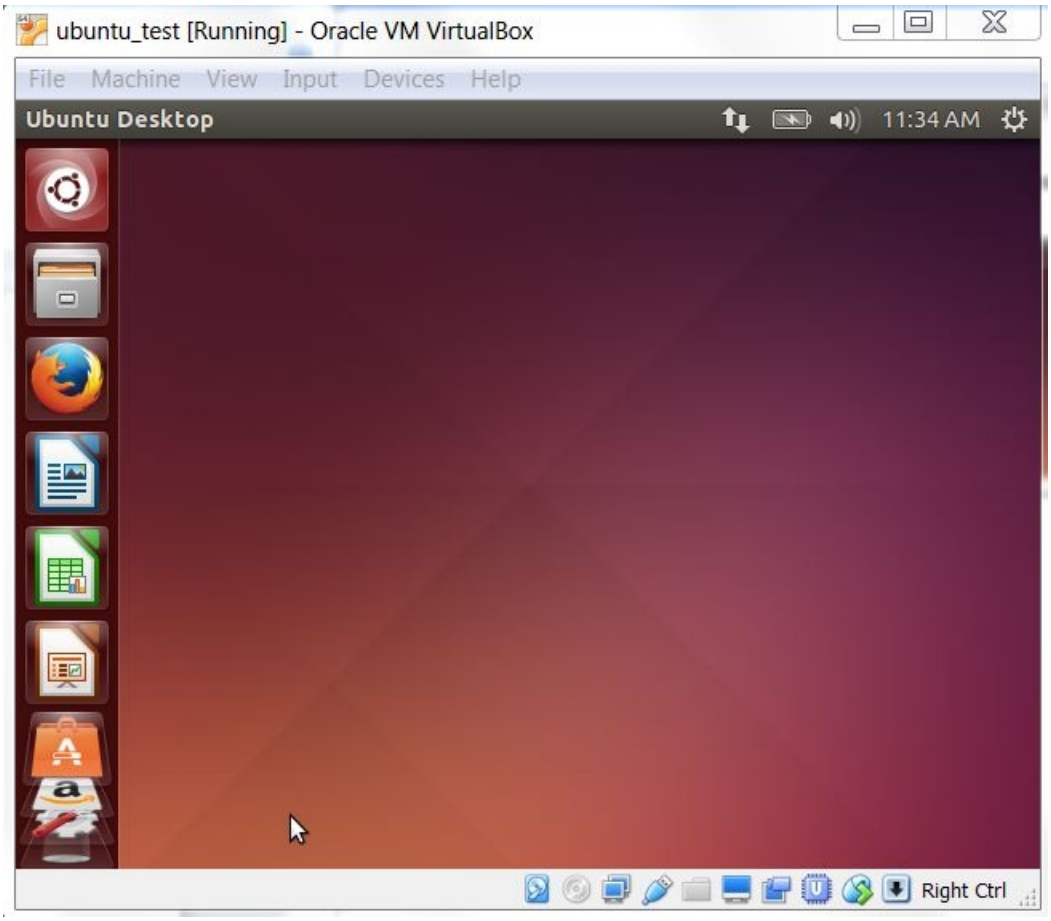
Wait for installation (which should be less than 10 minutes). You will be presented with a prompt to reboot the machine.



Click “Restart Now”, then “enter” when prompted. After rebooting this VM, it should appear as below:



You should be able to login with the password you assigned earlier, and your view should then look like the following:

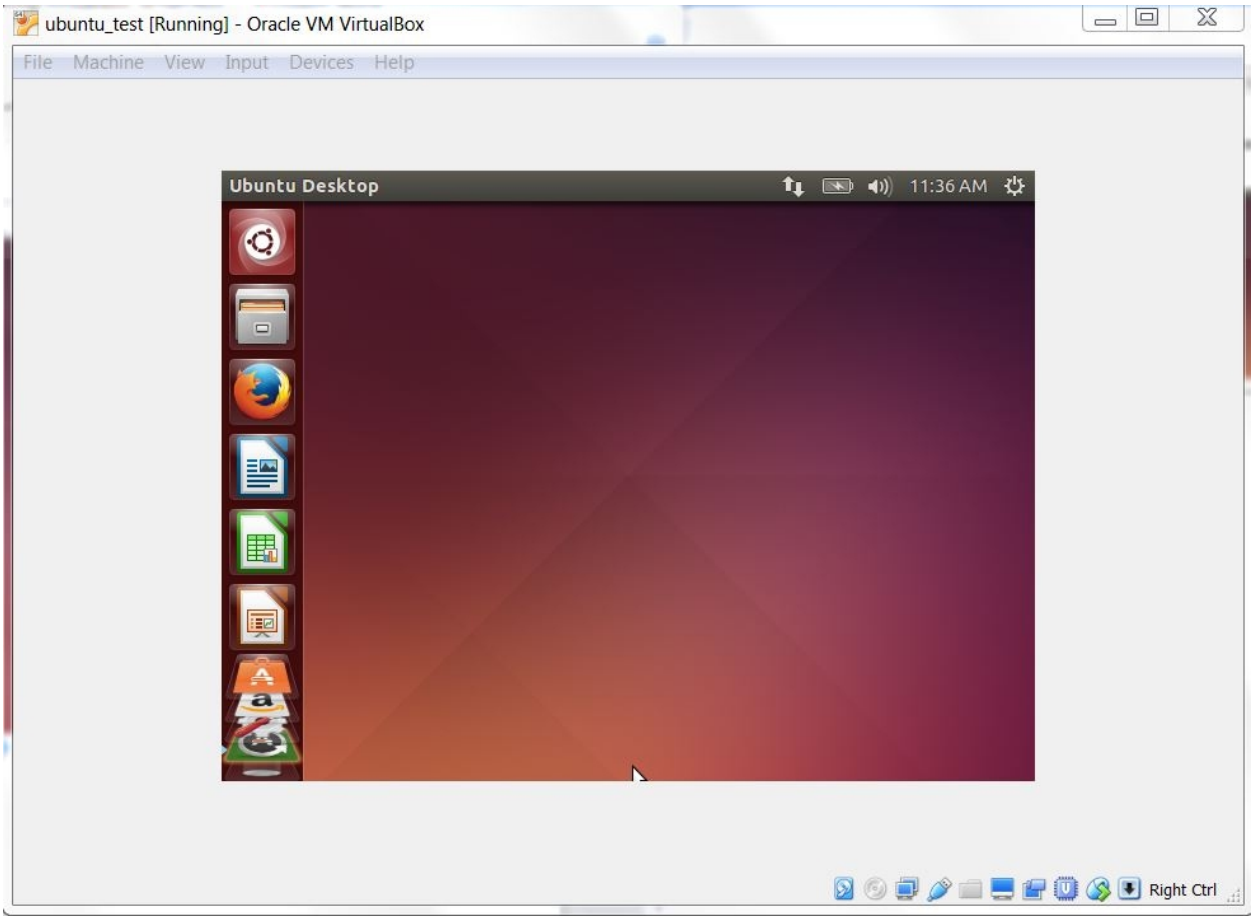


One of the most frequently used programs in Ubuntu with ROS is a simple ASCII “terminal.” You can find the terminal program by clicking on the “search your computer” icon (topmost icon on the left column in the Ubuntu window of your VM), then start to type “terminal”. After the first “t”, the searcher will show the terminal icon, as in the view below:

Since terminals are used frequently in ROS, I find it convenient to drag this icon to the task bar on the left, so it is quickly available to bring up new terminals. Clicking on the terminal brings up a simple, black command terminal.

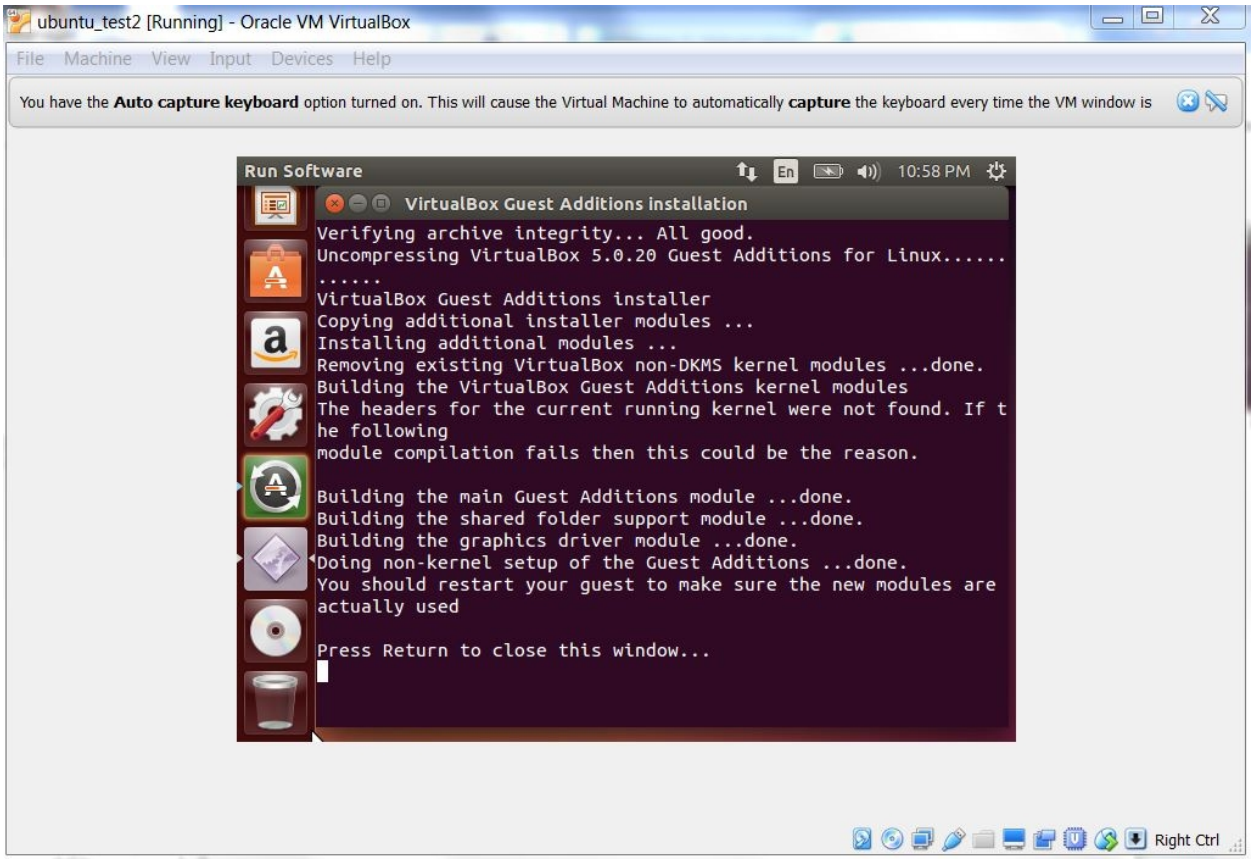
4) **Install Guest Additions:**

The VM view at this point looks like a normal Ubuntu machine. However, if you try to resize this window, it will appear something like the following:



The Ubuntu window gets no larger—resizing only adds a blank border. To fix this, install “guest additions”, as follows.

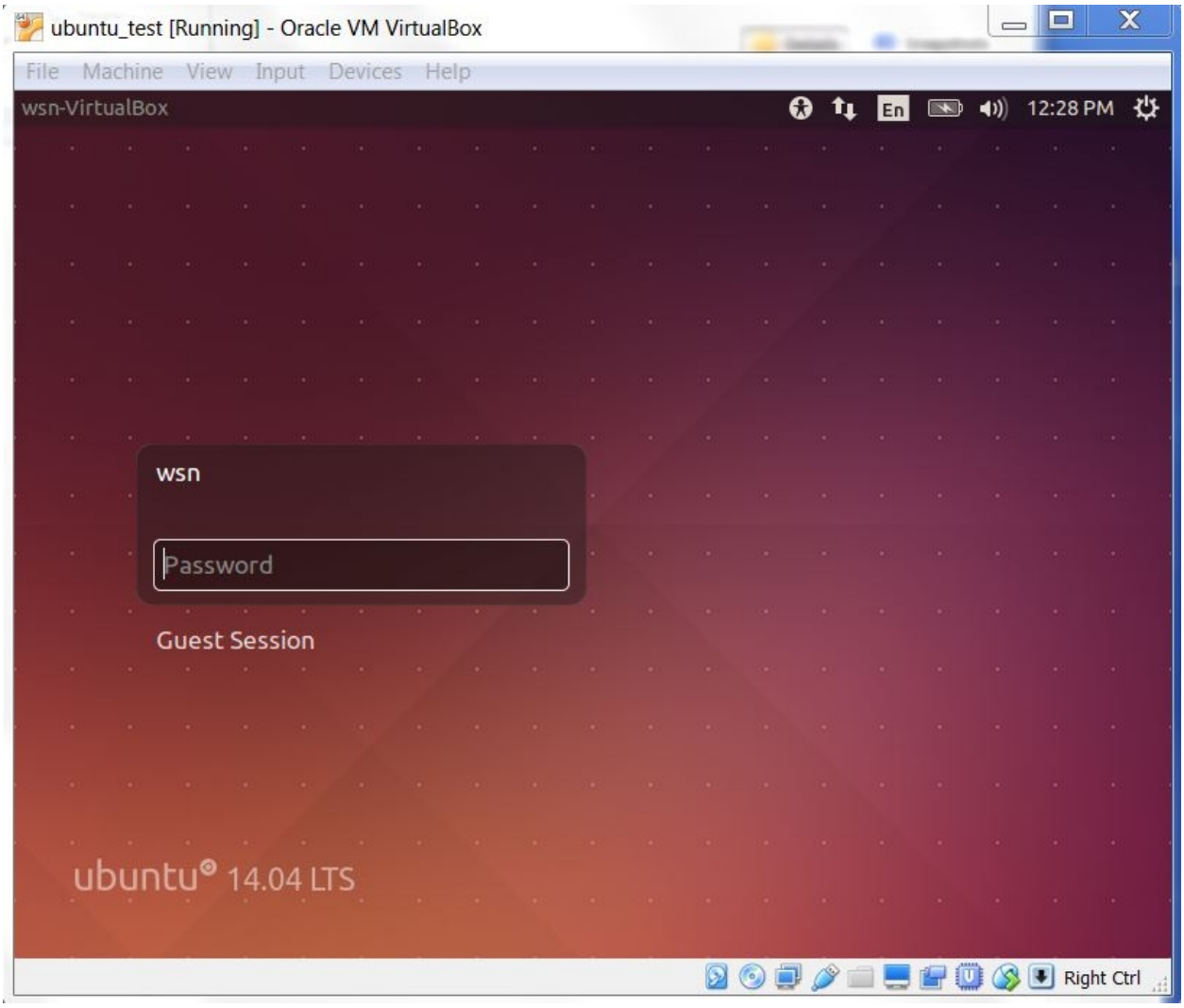
In the virtual machine (as seen above), note that the top menu bar has an item named “Devices”. Under “devices”, choose “Insert Guest Additions CD image …”. Accept the installation (although the “accept” button may be partially obscured). You will be prompted to enter your password again. After installation, the VM will display:



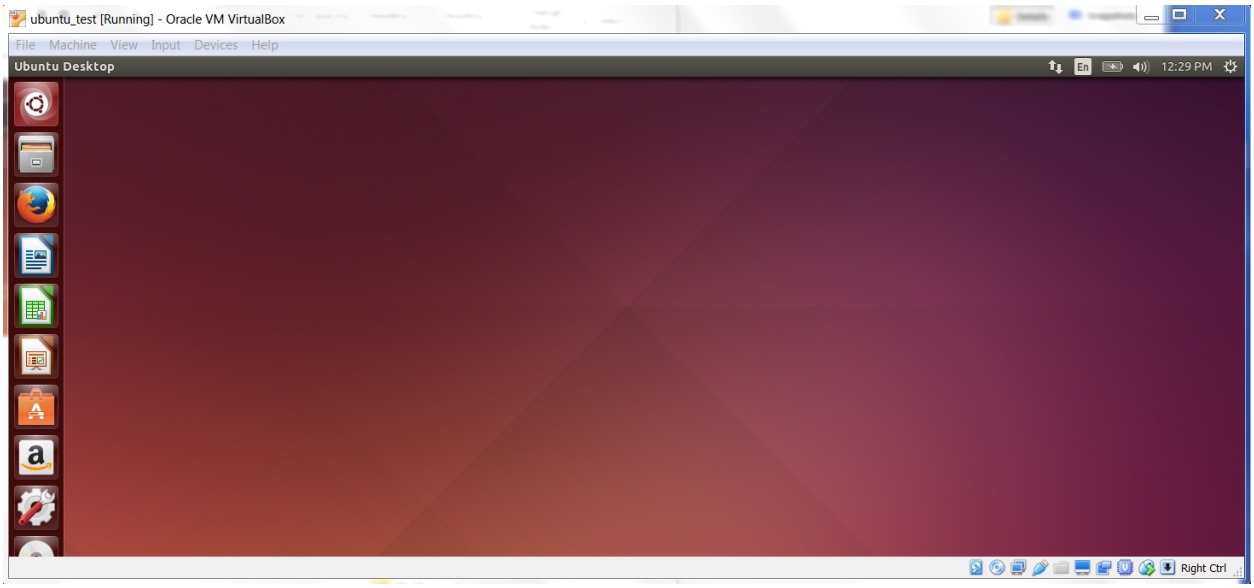
Enter “return” to close the virtual machine, then re-launch it by double-clicking its icon in VirtualBox.

(Note: if this process does not work, an alternative installation approach may be required).

Restart your virtual machine (double-click its icon in VirtualBox). You can close (x-out) the notices that appear at the top of your VM display, so your login view looks like:



Now, when you resize your VM's window, you can make it smaller, larger, or change its aspect ratio, obtaining alternative graphics views, e.g. as below:

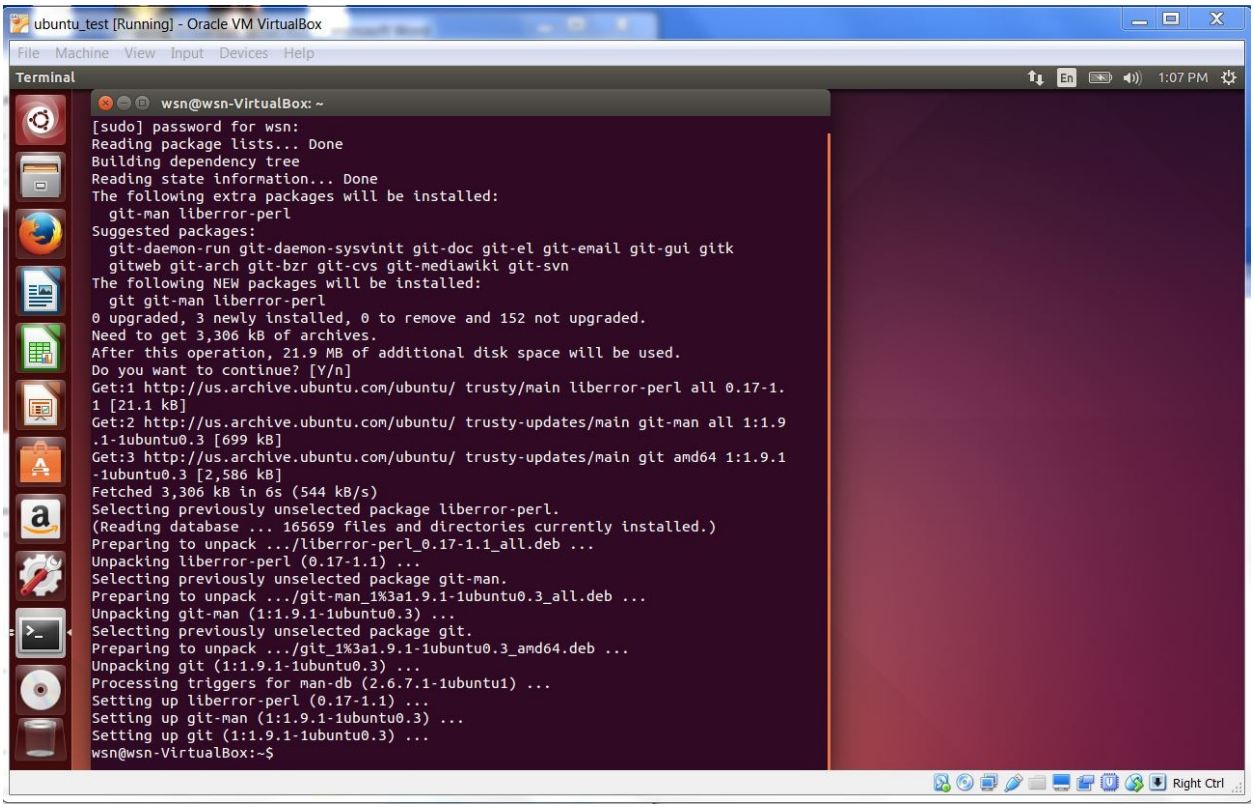


5) **Install ROS and development tools:**

To obtain on-line repositories, we will need to install “git.” To do so, open up a terminal in your VM and enter the command:

sudo apt-get install git

Accept the Y/n prompt. Your result should appear as below:



“git” is now installed. To clone repositories from github, you will want a “github” username and password, which you can create for free at “github.com”.

The following site contains instructions and scripts to automate setting up ROS on your machine (which works both for native installation of Ubuntu or for a virtual machine):
https://github.com/wsnewman/learning_ros_setup_scripts

The scripts here automate installing ROS, installing the example code for “Learning ROS,” installing additional useful tools, and setting up your ROS workspace. Instructions for obtaining and running these scripts are provided on-line at this site. You should have a good internet connection when running these scripts. Installing ROS and all associated code and tools will take approximately 20 minutes (with a good connection).

6) Test ROS and Gazebo:

Open a new terminal (which will be initialized with the new .bashrc file).
Within the new terminal, type:

```
roscd
```

This should bring you to the directory “ros_ws”. From here, enter the command:

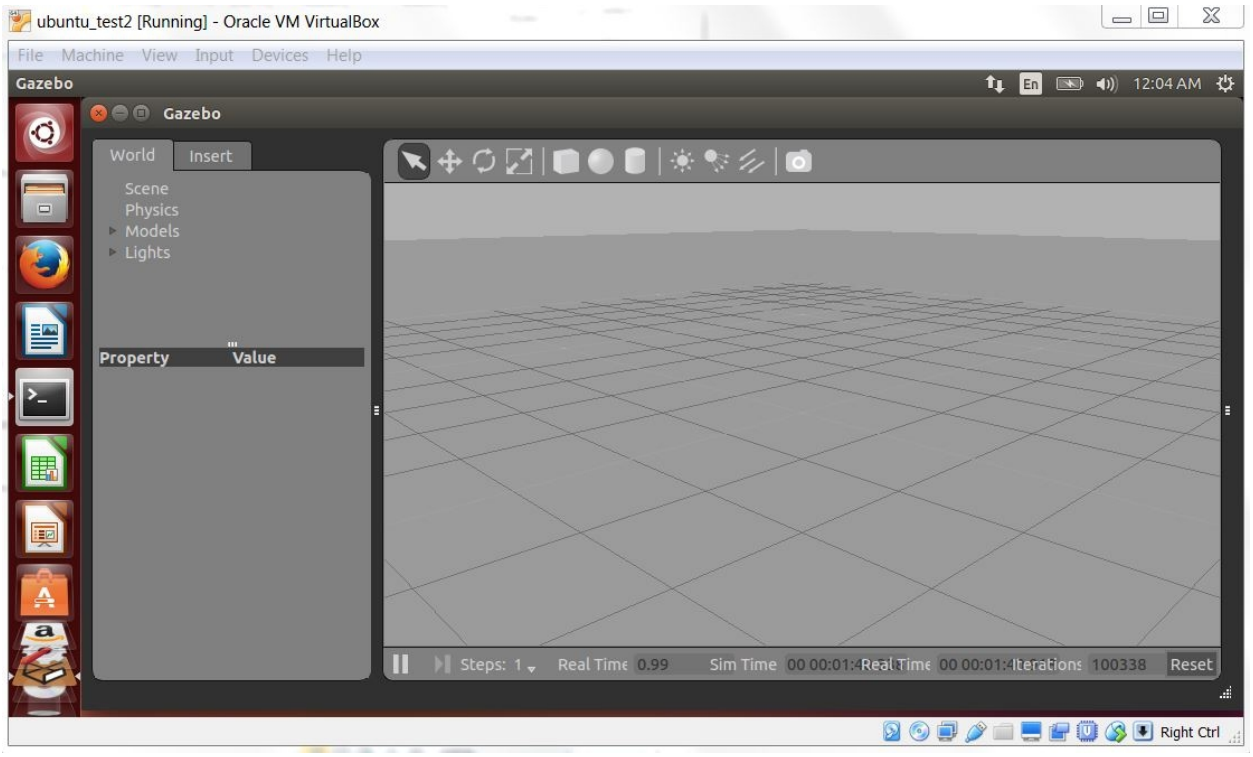
```
catkin_make
```

The code under ros_ws/src should compile without errors.

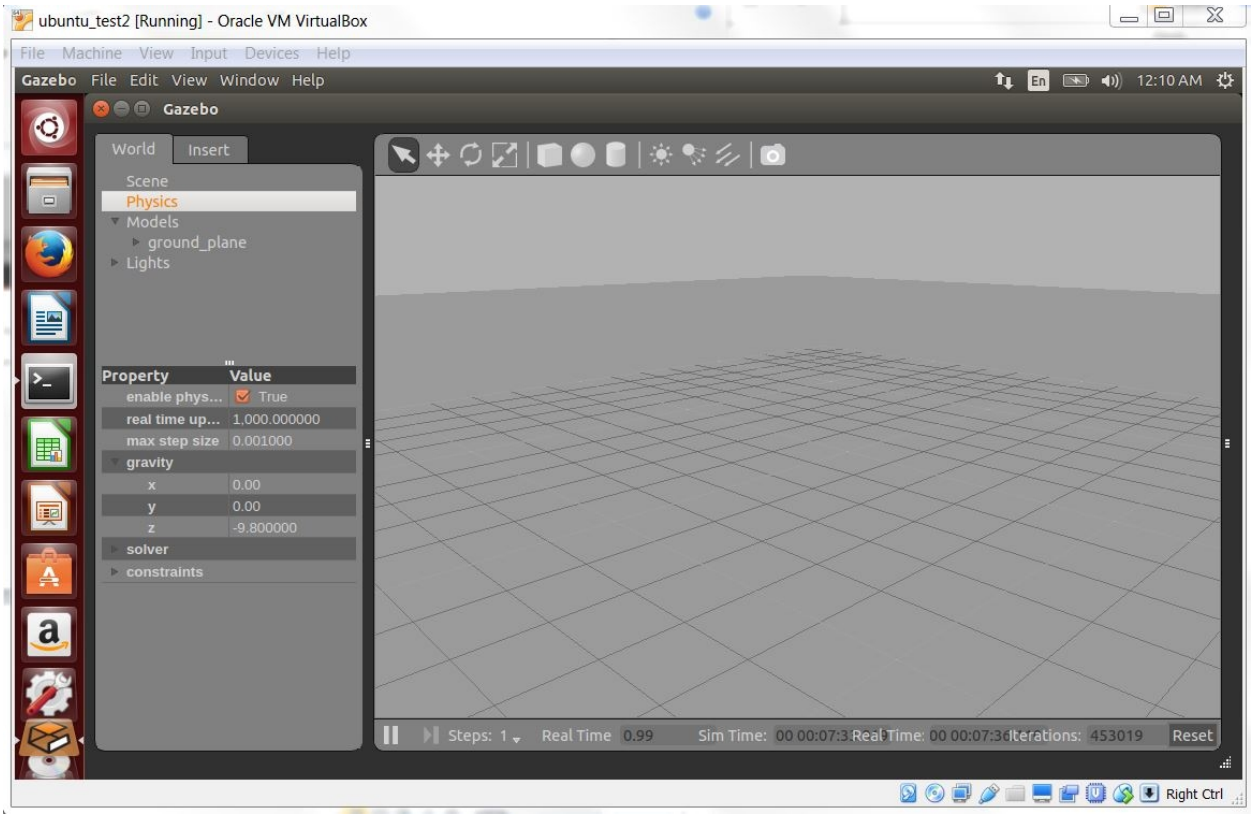
Try launching Gazebo by entering the command:

```
roslaunch gazebo_ros empty_world.launch
```

The screen should look like the following:



By default, the gravity model in Gazebo has 1g pointing down. For the next test, set gravity to zero. To do so, in the left panel of Gazebo, in the “world” tab, click on “physics”, then click on “gravity” to expand its parameter list. It will appear as below.

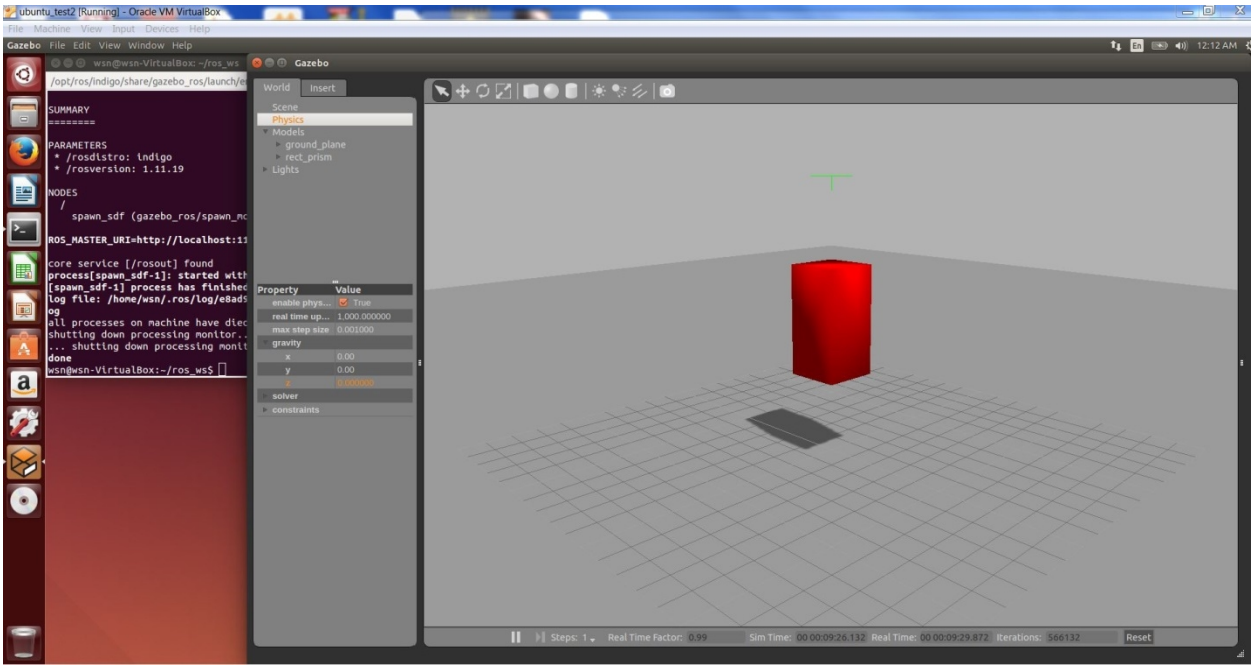


Edit the value of the z component of gravity, changing it from -9.8 to 0.

The terminal from which gazebo was launched must remain alive. Start up a new terminal. (A convenient technique is to enter ctl-shift-t, which will bring up a new terminal as a tab within a multi-terminal container). In the new terminal, enter:

```
roslaunch exmpl_models  add_rect_prism.launch
```

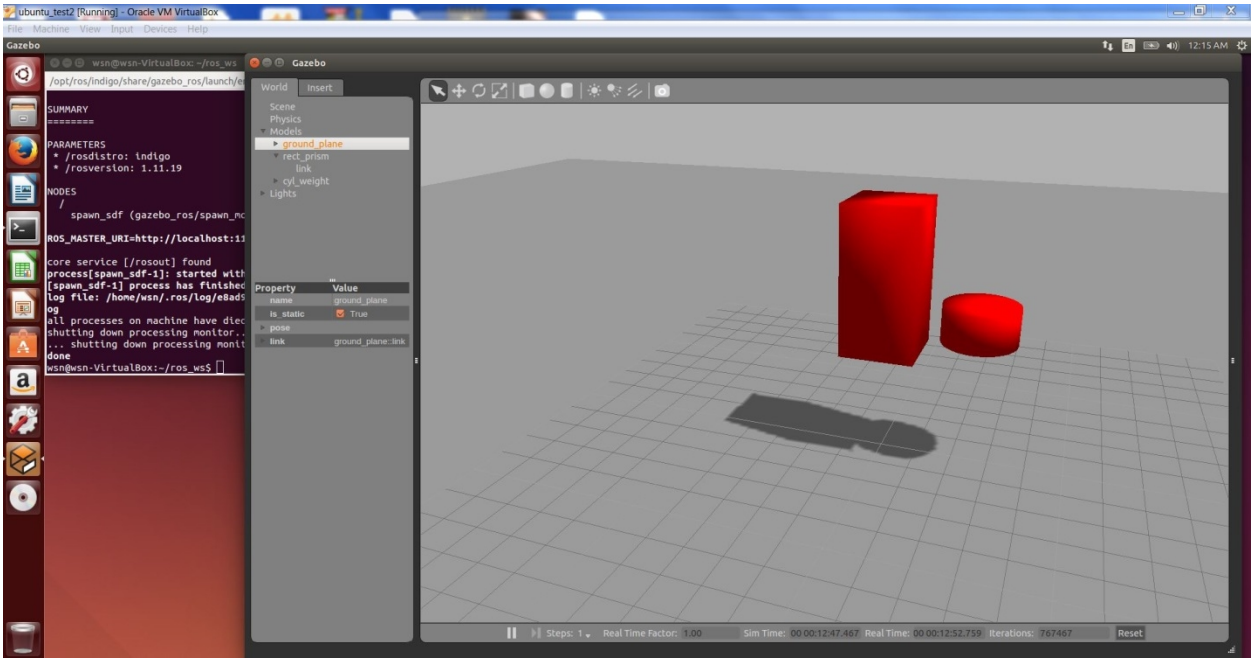
The result should appear as below, with a rectangular prism hovering above the ground plane:



The add_rect_prism.launch command will run to conclusion. In the same window, enter:

  roslaunch exmpl_models add_cylinder.launch

This should add a second model—a cylinder—which should appear as below:



This launch file will also run to conclusion. Next, impose an initial state on the prism by running:

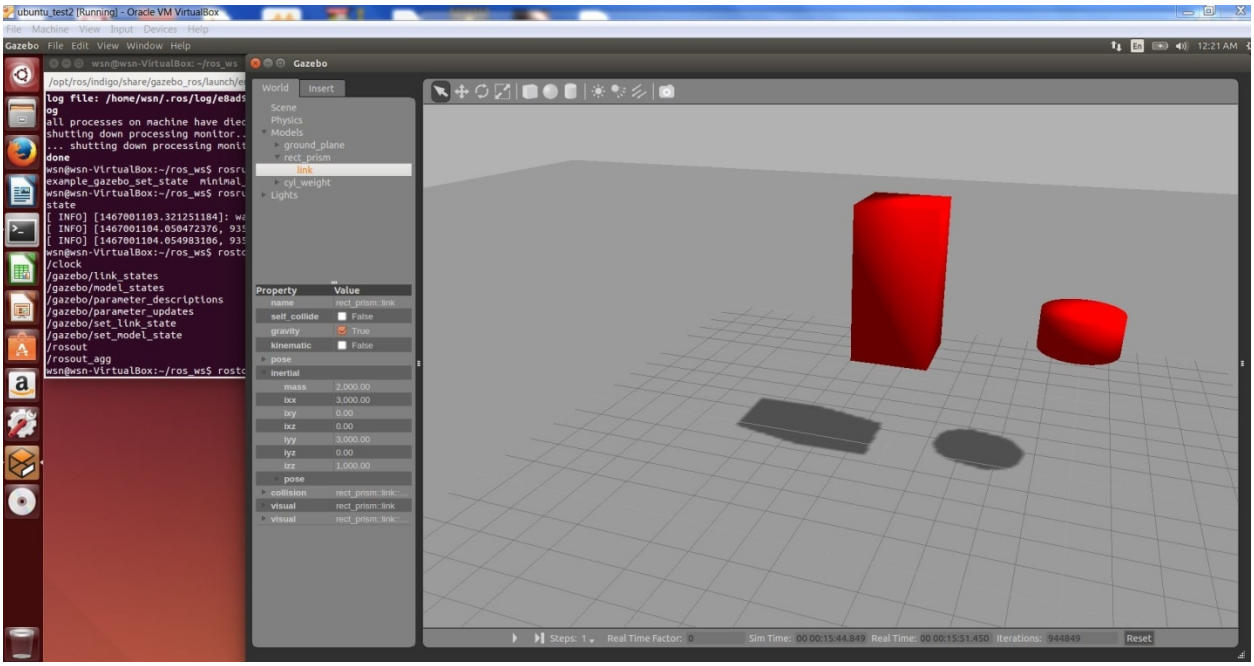
rosrun example_gazebo_set_state example_gazebo_set_prism_state

This will cause the prism to rotate and translate. It will move towards the cylinder and collide with the cylinder, inducing a momentum transfer. The states can be monitored by typing:

  rostopic echo /gazebo/model_states

Observing the momentum and angular momentum from this display, it can be proven that these quantities are conserved before and after the collision.

Properties of the inserted models can be seen from Gazebo. In the “World” tab, click on a model (e.g., rect_prism), select the (sole) “link” component, and expand the “inertial” property list. Gazebo will appear as follows:



This shows that the prism model has a mass of 2,000 kg and an moment of inertia about its z axis of 1,000 kg-m^2.

This demonstration shows that Gazebo is working, ROS is properly configured, and example ROS code is installed.