CS8711- CLOUD COMPUTING LABORATORY

OBJECTIVES:

The student should be made to:

- Be exposed to tool kits for grid and cloud environment.
- Be familiar with developing web services/Applications in grid framework
- Learn to run virtual machines of different configuration.
- Learn to use Hadoop

LIST OF EXPERIMENTS:

Course Objective:

- To develop web applications in cloud
- To learn the design and development process involved in creating a cloud based application
- To learn to implement and use parallel programming using Hadoop

Exercises:

1. Install Virtualbox/VMware Workstation with different flavours of linux or windows OS on top of windows7 or 8.
2. Install a C compiler in the virtual machine created using virtual box and execute Simple Programs
3. Install Google App Engine. Create hello world app and other simple web applications using python/java.
4. Use GAE launcher to launch the web applications.
5. Simulate a cloud scenario using CloudSim and run a scheduling algorithm that is not present in CloudSim.
6. Find a procedure to transfer the files from one virtual machine to another virtual machine.
7. Find a procedure to launch virtual machine using trystack (Online Openstack Demo Version)
8. Install Hadoop single node cluster and run simple applications like wordcount.

Course Outcome:

On completion of this course, the students will be able to:

- Configure various virtualization tools such as Virtual Box, VMware workstation.
- Design and deploy a web application in a PaaS environment.
- Learn how to simulate a cloud environment to implement new schedulers.
- Install and use a generic cloud environment that can be used as a private cloud.
- Manipulate large data sets in a parallel environment.

TOTAL: 45 PERIODS

CLOUD COMPUTING

EX.No:1 <u>Install Virtualbox/VMware Workstation</u>

Aim:

Find procedure to Install Virtualbox/VMware Workstation with different flavours of linux or windows OS on top of windows7 or 8.

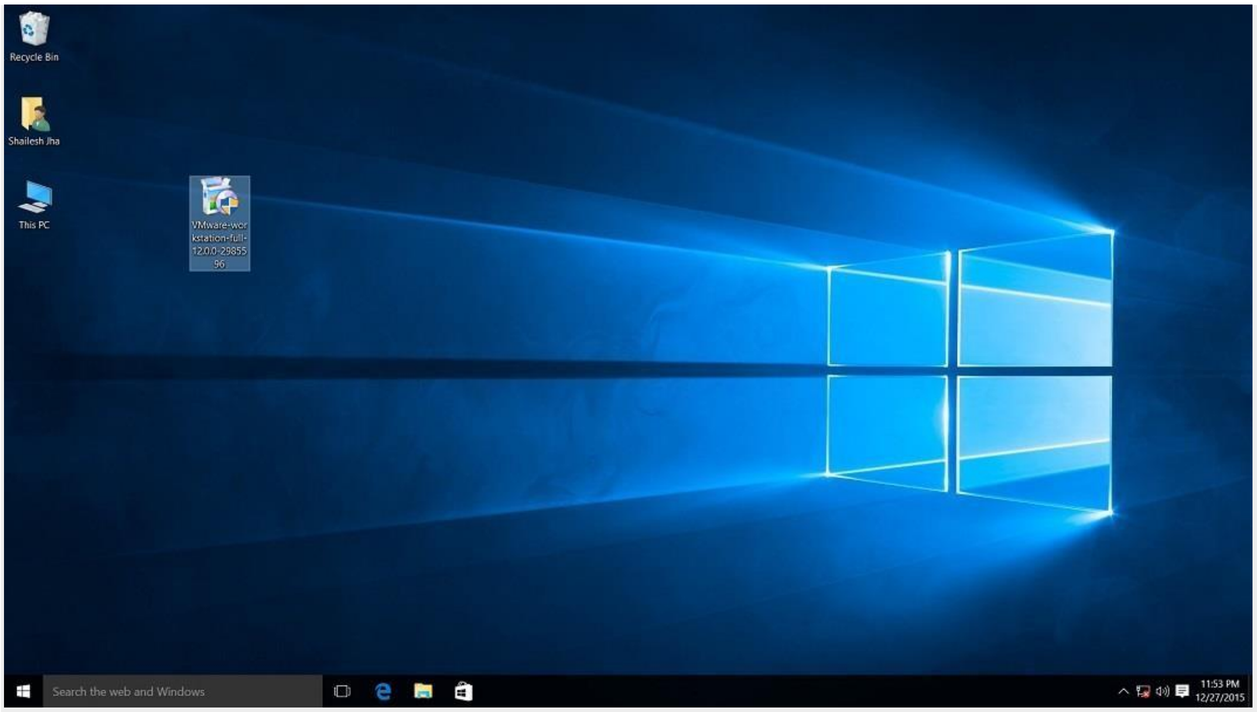
This experiment is to be performed through portal.

<u>PROCEDURE TO INSTALL</u>

Step 1- Download Link
Link for downloading the software is https://www.vmware.com/products/workstation-pro/workstation-pro-evaluation.html. Download the software for windows. Good thing is that there is no signup process. Click and download begins. Software is around 541 MB.

Step 2- Download the installer file
It should probably be in the download folder by default, if you have not changed the settings in your browser. File name should be something like <u>VMware-workstation-full-15.5.1-15018445.exe</u>. This file name can change depending on the version of the software currently available for download. But for now, till the next version is available, they will all be VMware Workstation 15 Pro.

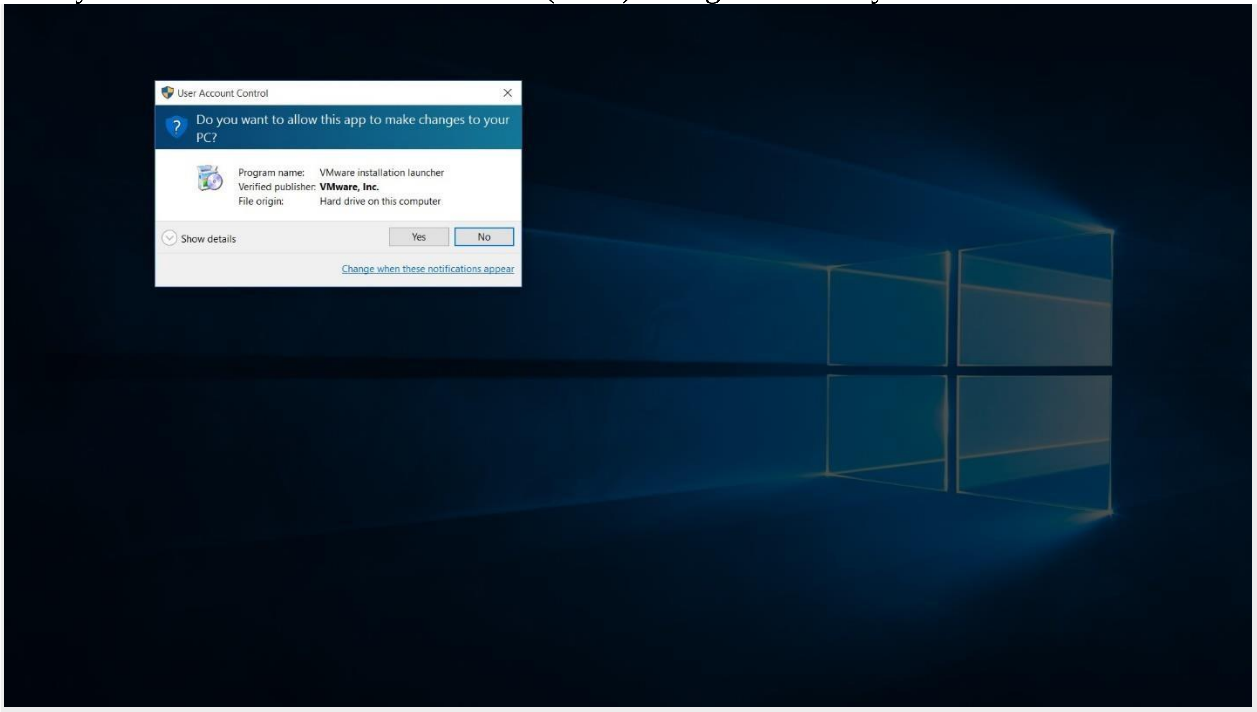
Step 3- Locate the downloaded installer file
For demonstration purpose, I have placed the downloaded installer on my desktop. Find the installer on your system and double click to launch the application.


Recycle Bin
Shailesh Jha
This PC
VMware-workstation-full-12.0.0-2985596
Search the web and Windows
11:53 PM
12/27/2015

VMware workstation 15 pro for windows 10 installer file screenshot.

Step 4- User Access Control (UAC) Warning

Now you should see User Access Control (UAC) dialog box. Click yes to continue.

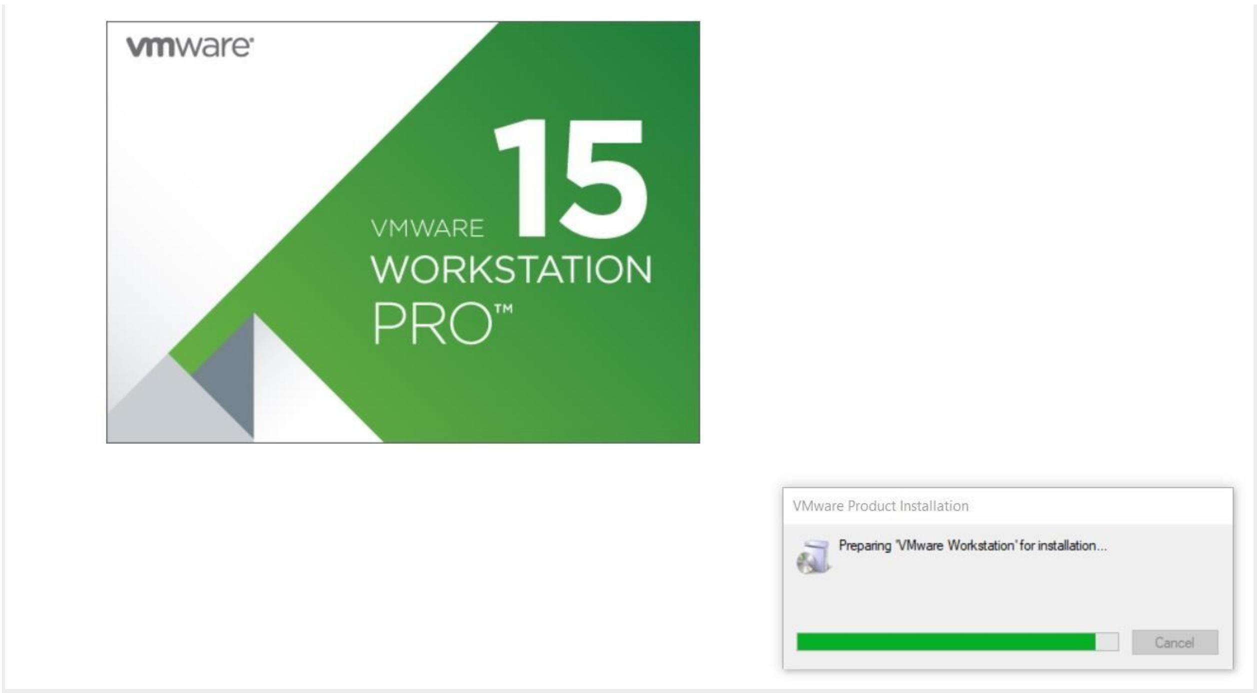

User Account Control

Do you want to allow this app to make changes to your PC?

Program name: VMware installation launcher
Verified publisher: **VMware, Inc.**
File origin: Hard drive on this computer

VMware Workstation 12 Pro installer windows 10 UAC screenshot

Initial Splash screen will appear. Wait for the process to complete.

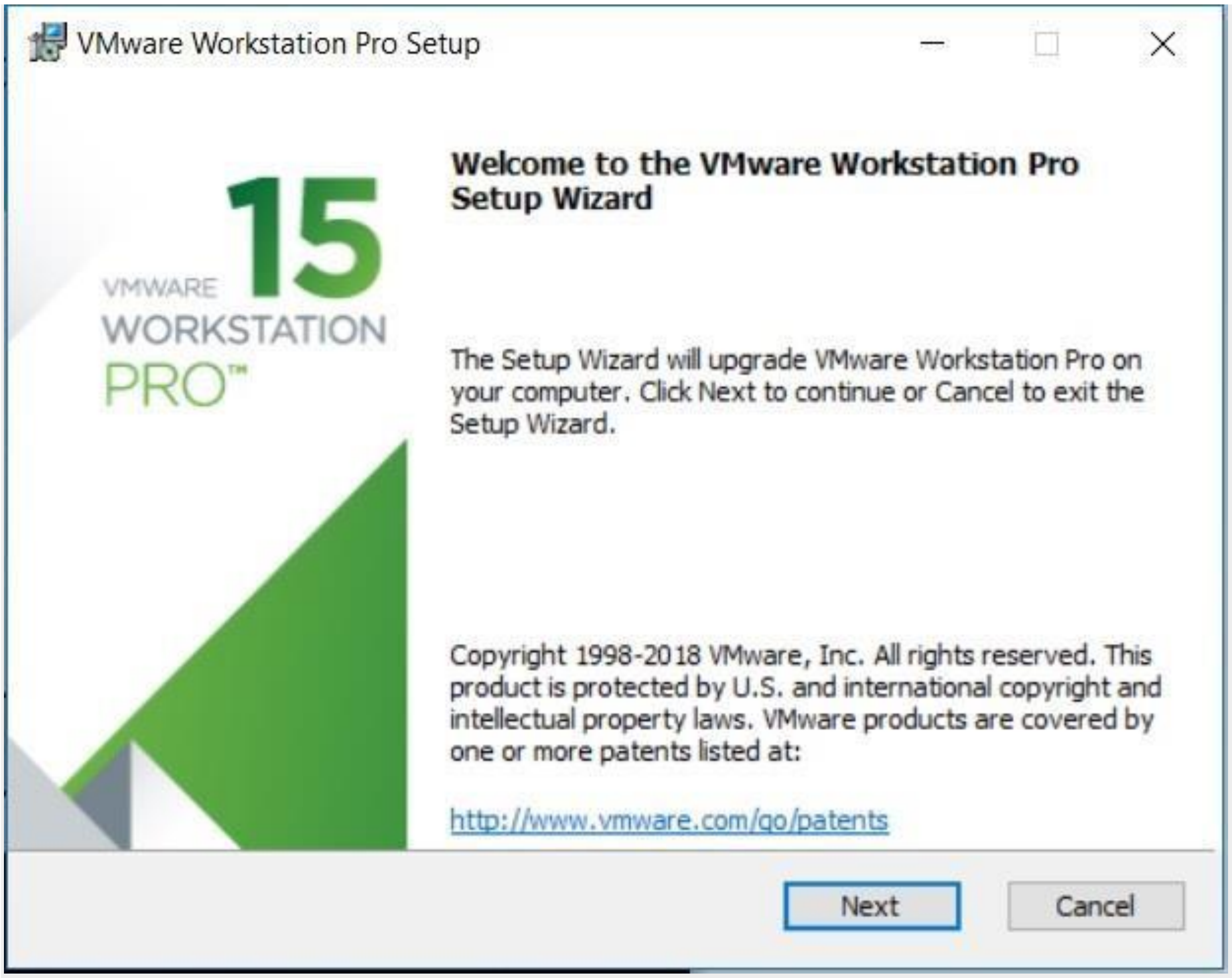


Preparing 'VMware Workstation' for installation...

VMware Workstation 15 Installation Splash Screen

Step 5- VMware Workstation Setup wizard

Now you will see VMware Workstation setup wizard dialog box. Click next to continue.


VMware Workstation Pro Setup

Welcome to the VMware Workstation Pro Setup Wizard

The Setup Wizard will upgrade VMware Workstation Pro on your computer. Click Next to continue or Cancel to exit the Setup Wizard.

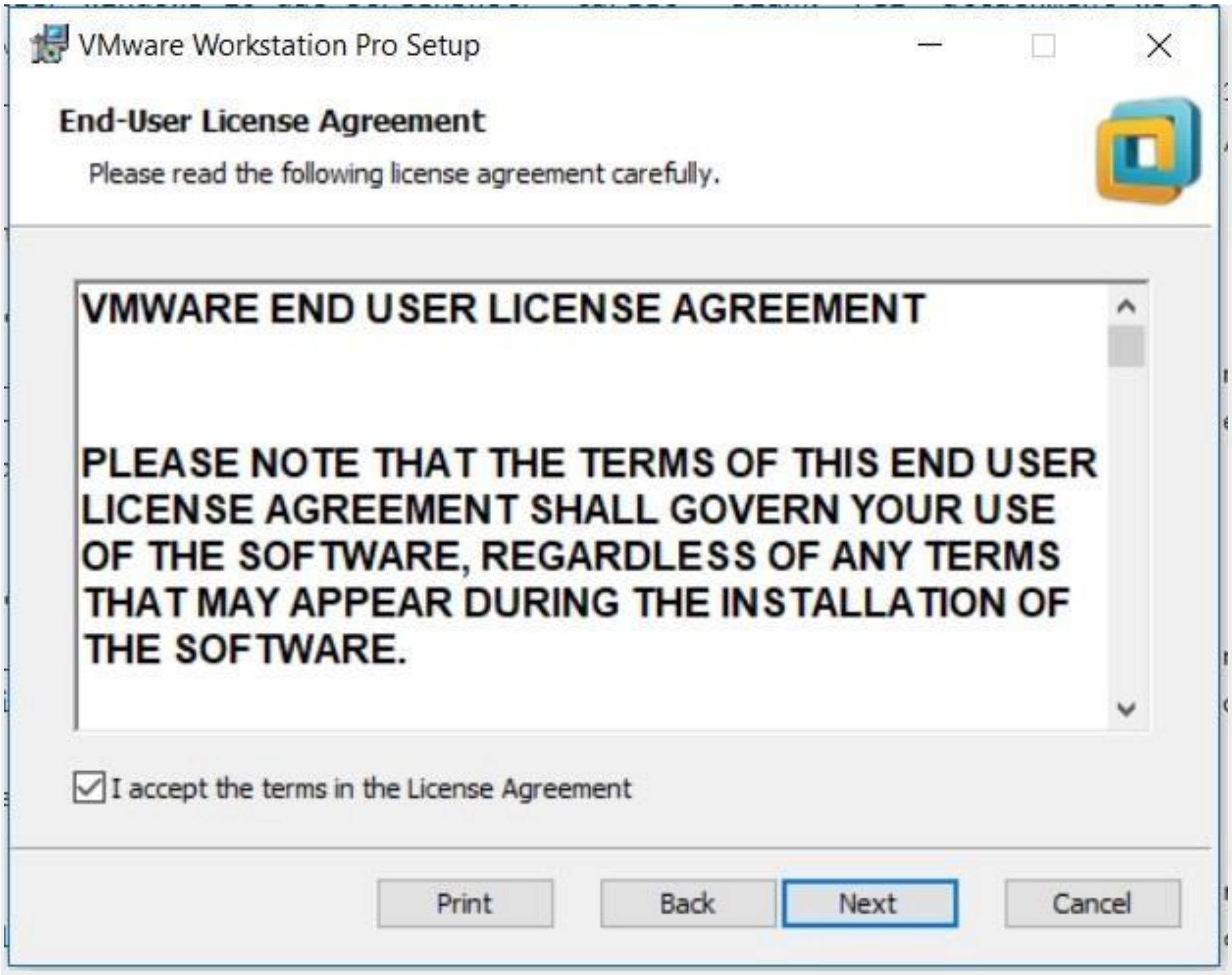
Copyright 1998-2018 VMware, Inc. All rights reserved. This product is protected by U.S. and international copyright and intellectual property laws. VMware products are covered by one or more patents listed at:

http://www.vmware.com/go/patents

VMware Workstation 15 Installation – Setup Wizard

Step 6- End User Licence Agreement

This time you should see End User Licence Agreement dialog box. Check “I accept the terms in the Licence Agreement” box and press next to continue.



VMware Workstation Pro Setup

End-User License Agreement

Please read the following license agreement carefully.

VMWARE END USER LICENSE AGREEMENT

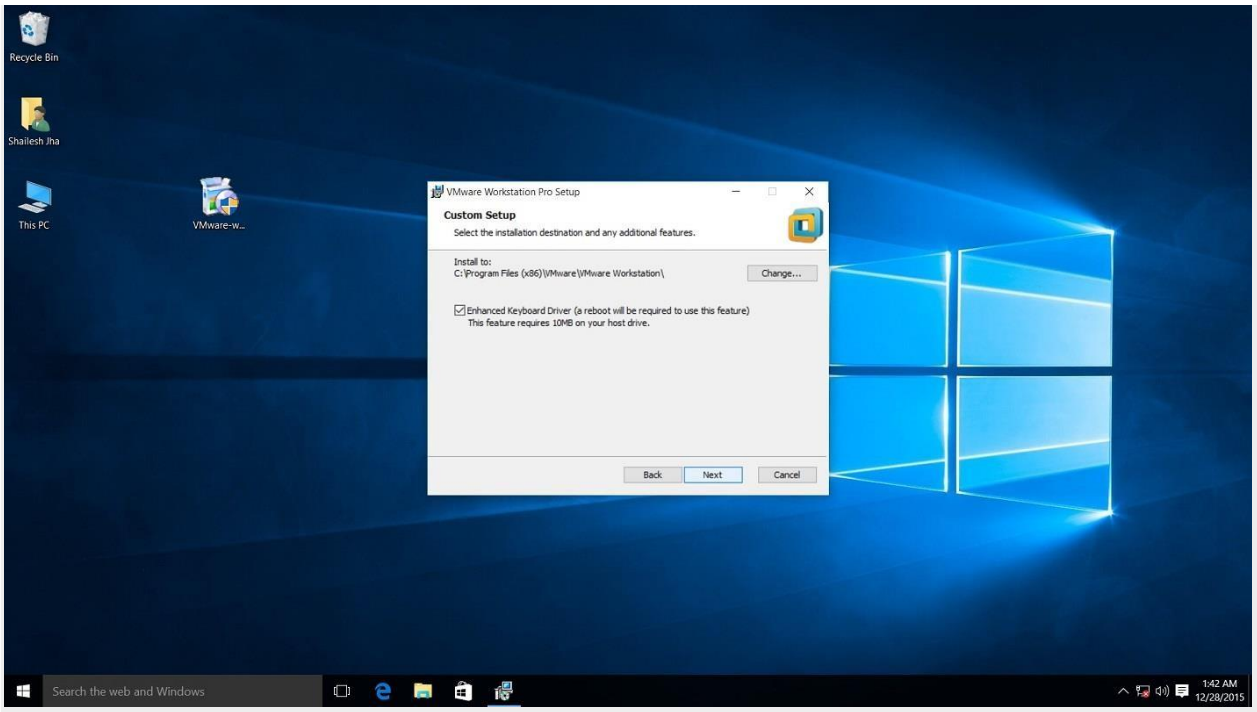
PLEASE NOTE THAT THE TERMS OF THIS END USER LICENSE AGREEMENT SHALL GOVERN YOUR USE OF THE SOFTWARE, REGARDLESS OF ANY TERMS THAT MAY APPEAR DURING THE INSTALLATION OF THE SOFTWARE.

[✓] I accept the terms in the License Agreement

VMware Workstation 15 Installation – End User Licence Agreement

Step 7- Custom Setup options

Select the folder in which you would like to install the application. There is no harm in leaving the defaults as it is. Also select Enhanced Keyboard Driver check box.



Custom Setup

Select the installation destination and any additional features.

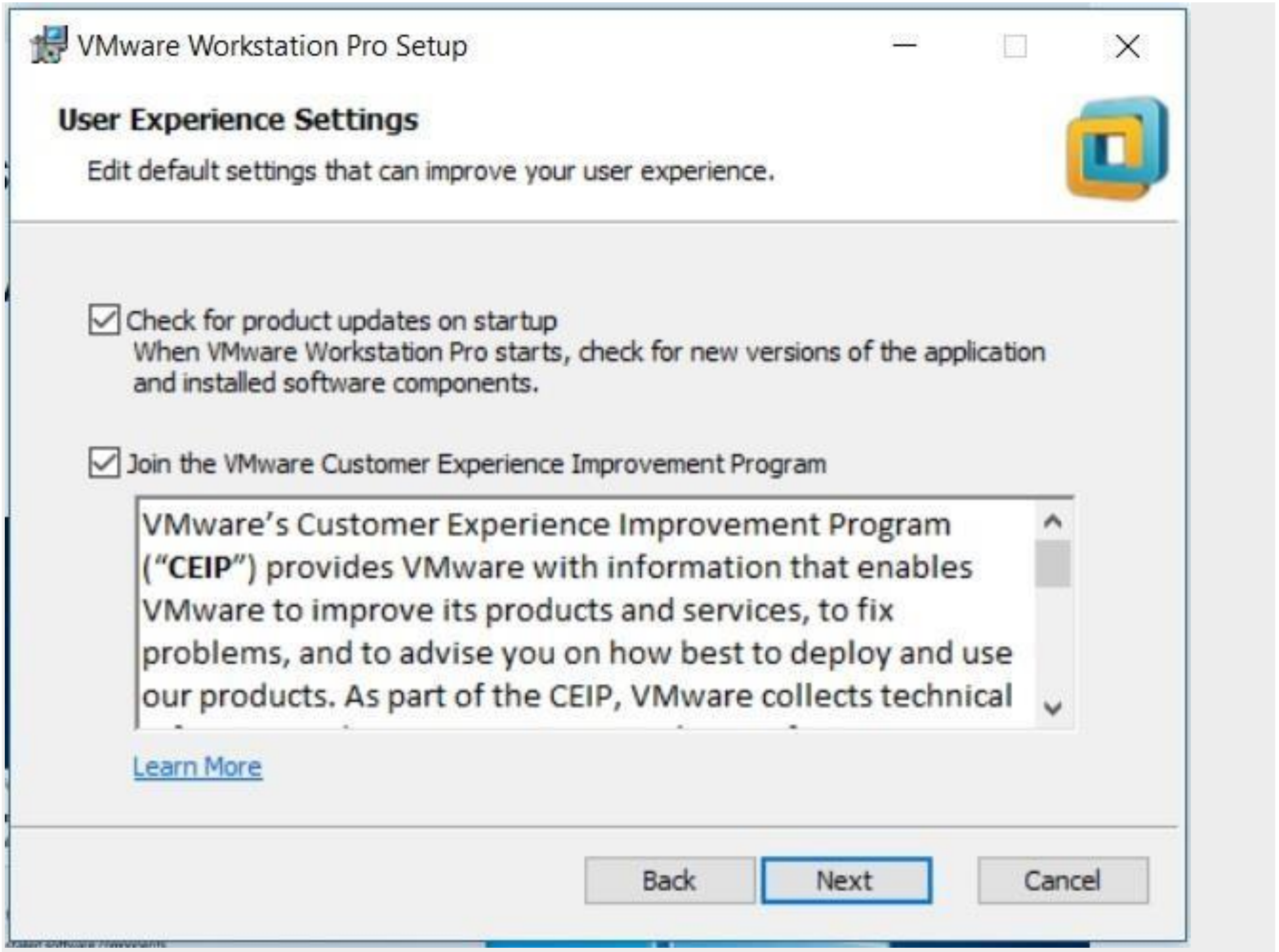
Install to:
C:\Program Files (x86)\VMware\VMware Workstation\ [Change...]

☑ Enhanced Keyboard Driver (a reboot will be required to use this feature)
This feature requires 10MB on your host drive.

VMware Workstation 15 Pro installation – select installation folder

Step 8- User Experience Settings

Next you are asked to select “Check for Updates” and “Help improve VMware Workstation Pro”. Do as you wish. I normally leave it to defaults that is unchecked.


VMware Workstation Pro Setup

User Experience Settings

Edit default settings that can improve your user experience.

[✓] Check for product updates on startup
When VMware Workstation Pro starts, check for new versions of the application and installed software components.

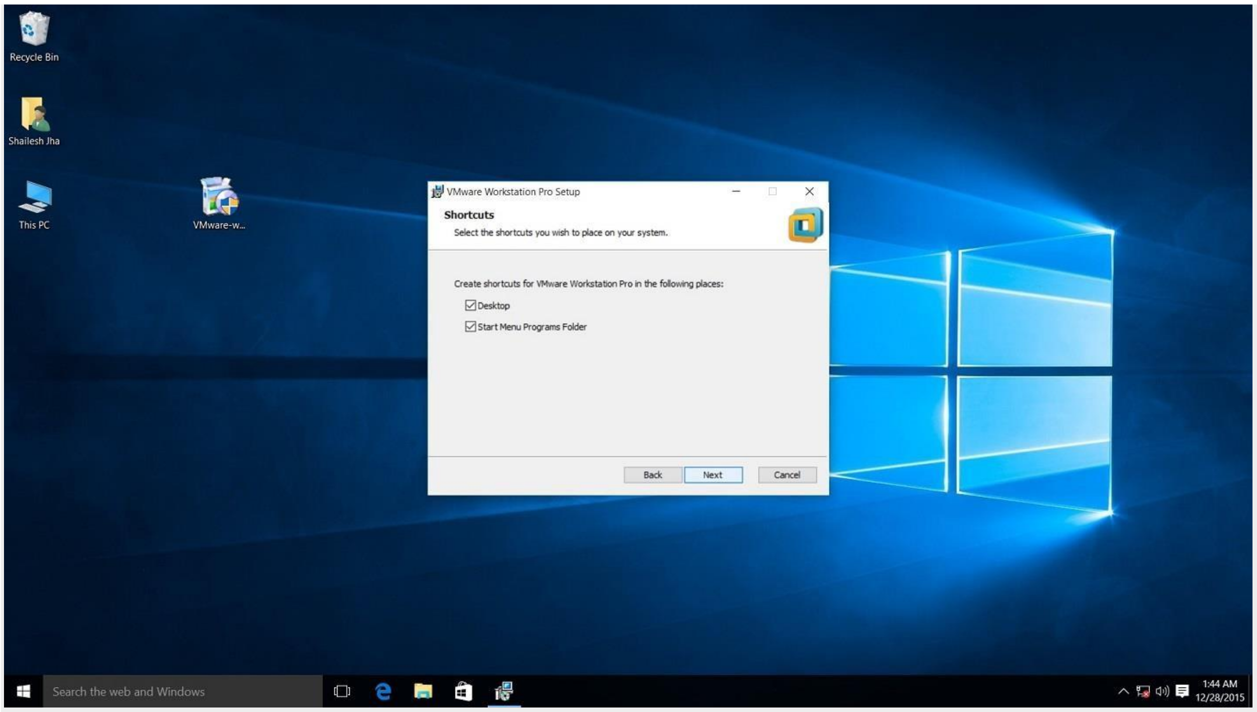
[✓] Join the VMware Customer Experience Improvement Program

VMware's Customer Experience Improvement Program ("**CEIP**") provides VMware with information that enables VMware to improve its products and services, to fix problems, and to advise you on how best to deploy and use our products. As part of the CEIP, VMware collects technical

VMware Workstation 15 Installation – User Experience Settings

Step 9- Application Shortcuts preference

Next step is to select the place you want the shortcut icons to be placed on your system to launch the application. Please select both the options, desktop and start menu and click next.


VMware Workstation Pro Setup

Shortcuts

Select the shortcuts you wish to place on your system.

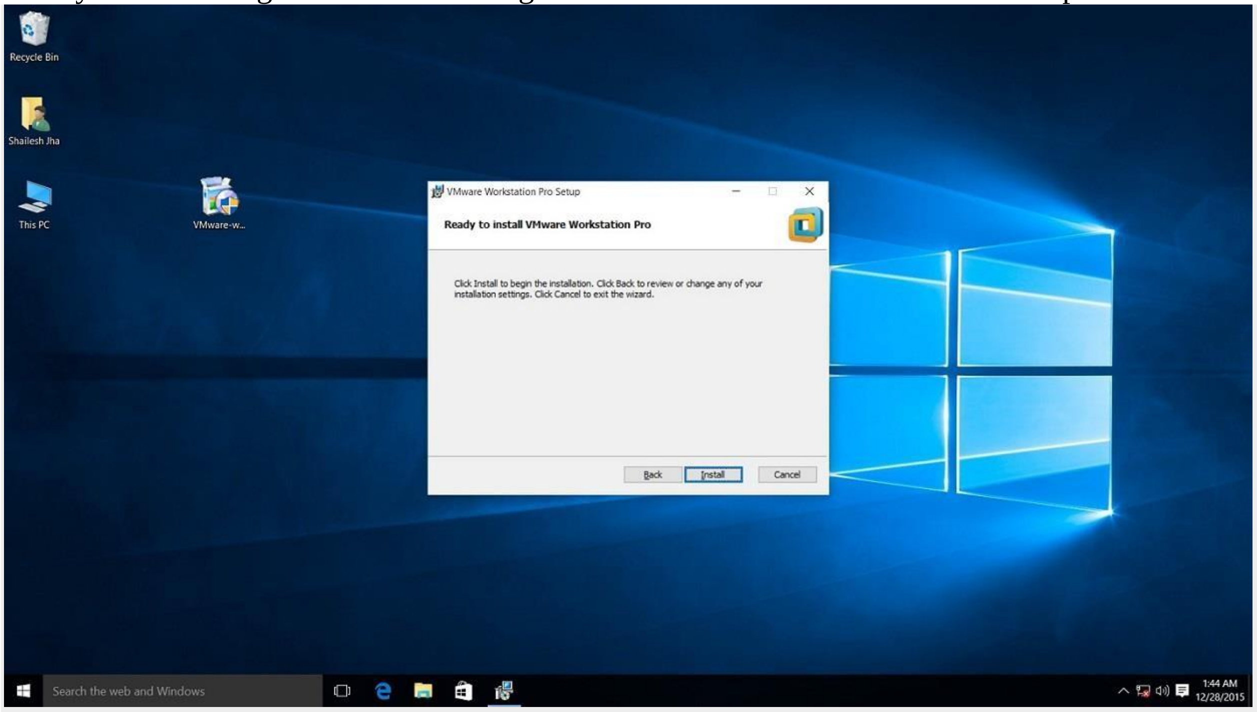
Create shortcuts for VMware Workstation Pro in the following places:

- [✓] Desktop
- [✓] Start Menu Programs Folder

VMware workstation 15 pro installation shortcut selection checkbox screenshot.

Step 10- Installation begins

Now you see the begin installation dialog box. Click install to start the installation process.

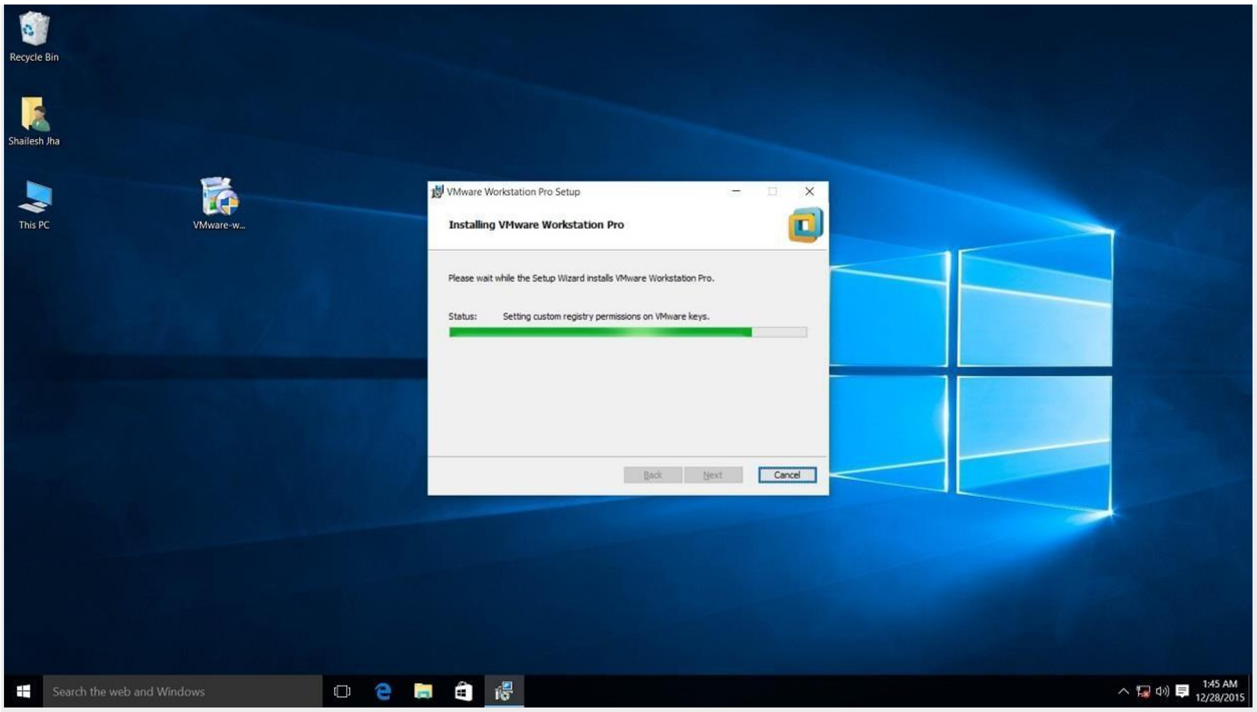

VMware Workstation Pro Setup

Ready to install VMware Workstation Pro

Click Install to begin the installation. Click Back to review or change any of your installation settings. Click Cancel to exit the wizard.

Screenshot for VMware Workstation 15 pro installation begin confirmation dialog box on windows 10.

Below screenshot shows Installation in progress. Wait for this to complete.



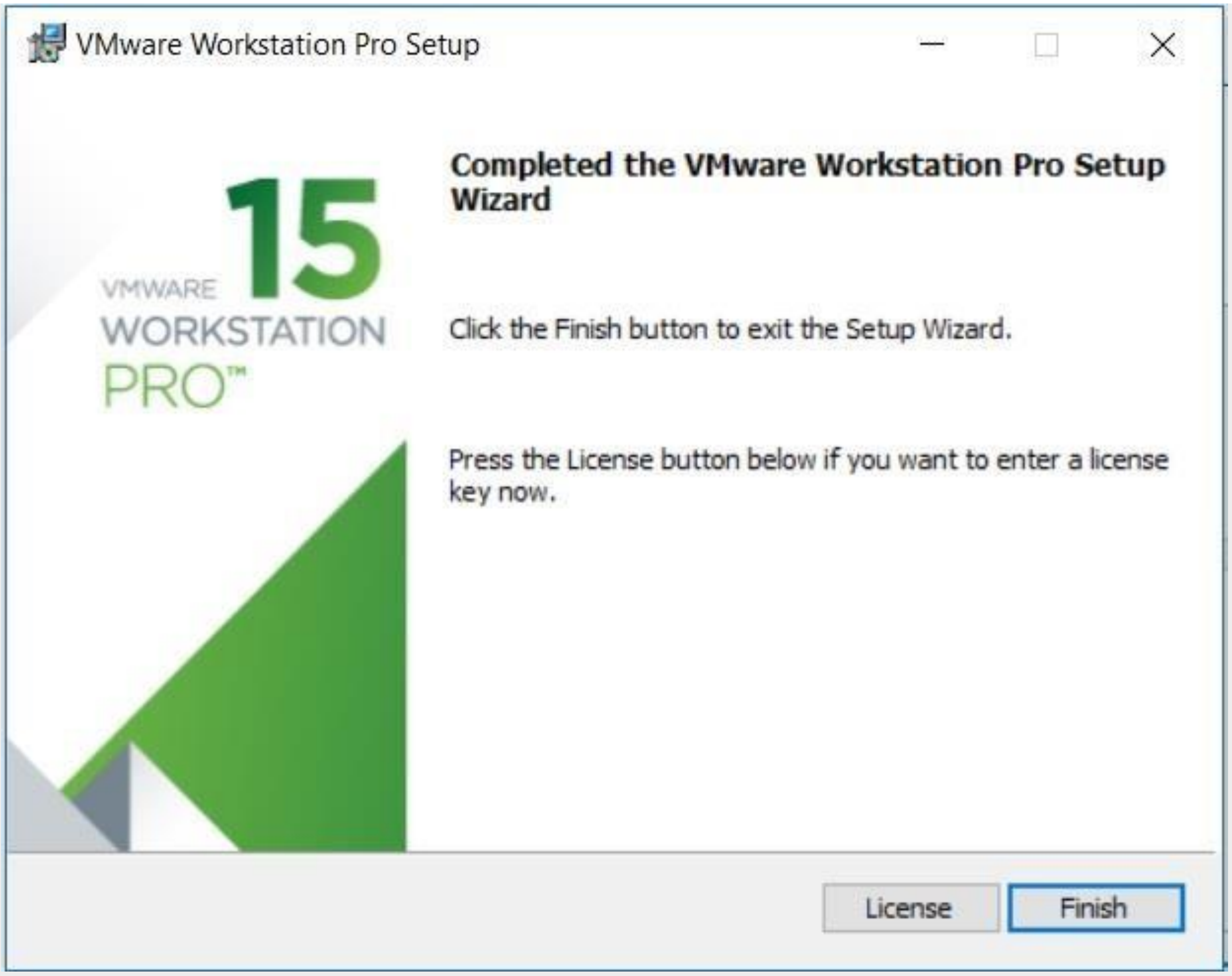
Installing VMware Workstation Pro

Please wait while the Setup Wizard installs VMware Workstation Pro.

Status: Setting custom registry permissions on VMware keys.

Screenshot for VMware Workstation 15 pro installation process.

At the end you will see installation complete dialog box. Click finish and you are done with the installation process. You may be asked to restart your computer. Click on Yes to restart.



VMware Workstation Pro Setup

Completed the VMware Workstation Pro Setup Wizard

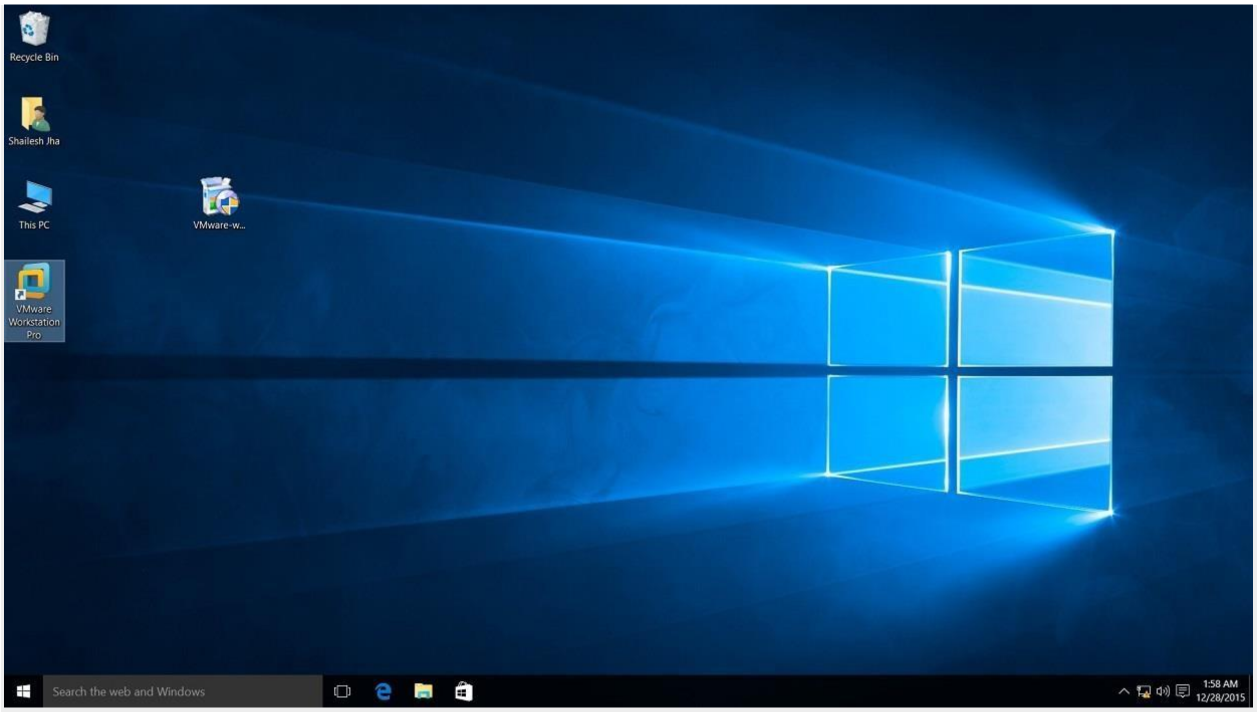
Click the Finish button to exit the Setup Wizard.

Press the License button below if you want to enter a license key now.

VMware Workstation 15 Installation – Installation Complete

Step 11- Launch VMware Workstation

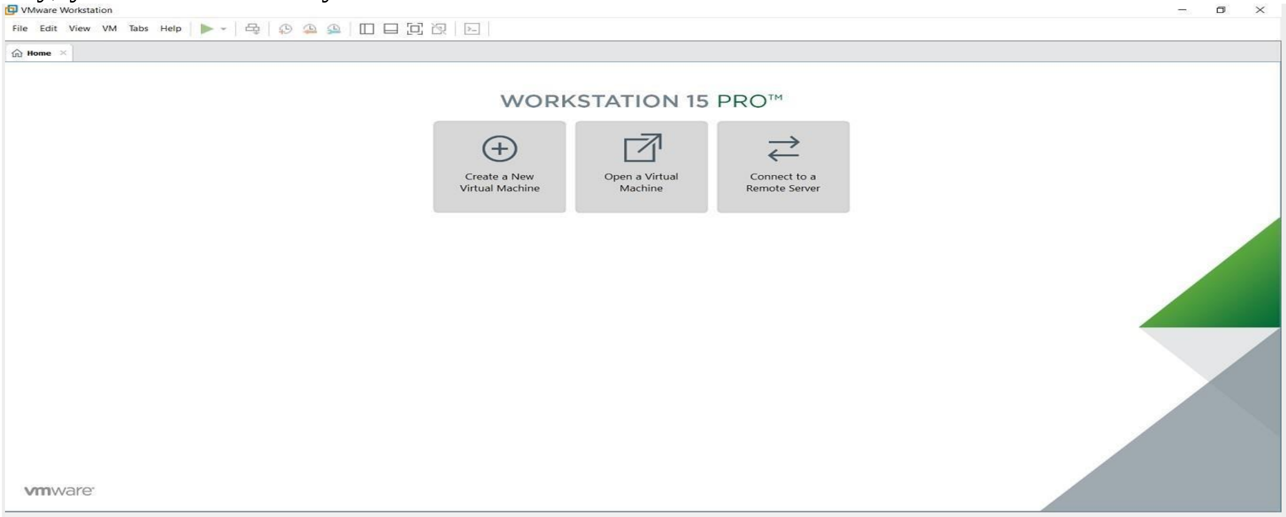
After the installation completes, you should see VMware Workstation icon on the desktop. Double click on it to launch the application.



Screenshot for VMware Workstation 15 Pro icon on windows 10 desktop.

Step 12- Licence Key

If you see the dialog box asking for licence key, click on trial or enter the licence key. Then what you have is the VMware Workstation 15 Pro running on your windows 10 desktop. If don't have the licence key, you will have 30 days trial.


VMware Workstation
File Edit View VM Tabs Help
Home
WORKSTATION 15 PRO™
Create a New Virtual Machine
Open a Virtual Machine
Connect to a Remote Server
vmware
VMware Workstation 15 Pro home screen

Step 13- At some point if you decide to buy

At some point of time if you decide to buy the Licence key, you can enter the Licence key by going to **Help->Enter a Licence Key**

You can enter the 25 character licence key in the dialog box shown below and click OK. Now you have the licence version of the software.

Setting up C Programming Environment

Before you start..

✓ Please note that this is “NOT” a required part of the course and is not a homework.

✓ This manual is written for someone who has never programmed / never used Linux before.

If you have background knowledge, you can ignore this.

✓ If you have any questions, please email
Yoonji Shin ys2476@columbia.edu

✓ Don’t be scared, programming is fun! Enjoy :)

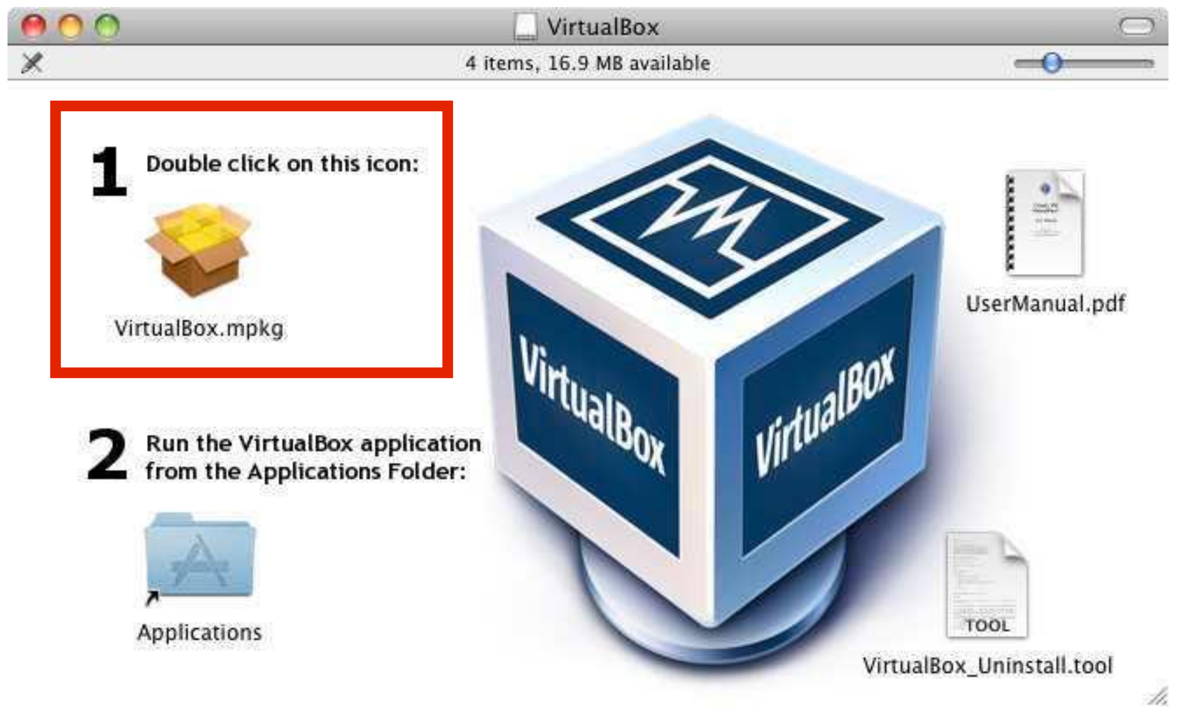
Install VirtualBox

I.Visit http://www.virtualbox.org/wiki/downloads

2. Download VirtualBox platform packages for your OS

3. Open the Installation Package by double clicking

MAC



VirtualBox

4 items, 16.9 MB available

1 Double click on this icon:

VirtualBox.mpkg

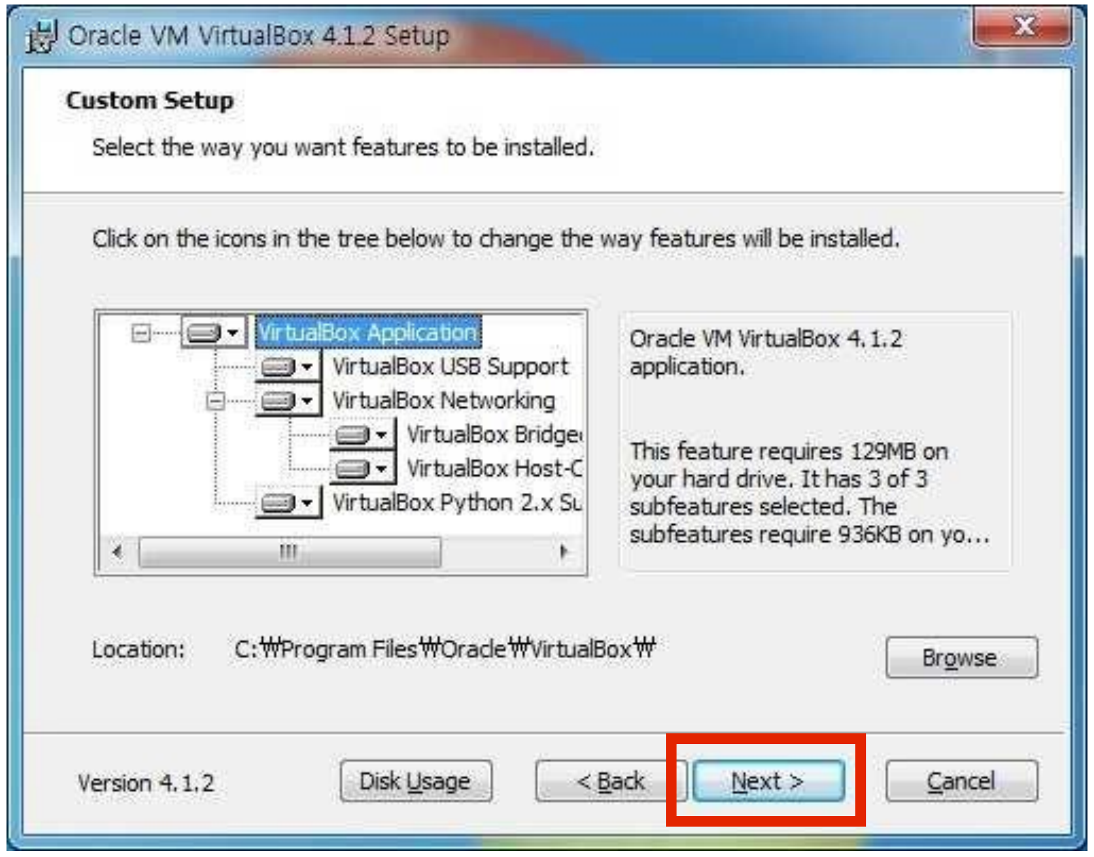
2 Run the VirtualBox application from the Applications Folder:

Applications

UserManual.pdf

VirtualBox_Uninstall.tool

PC



Oracle VM VirtualBox 4.1.2 Setup

Custom Setup

Select the way you want features to be installed.

Click on the icons in the tree below to change the way features will be installed.

- VirtualBox Application
  - VirtualBox USB Support
  - VirtualBox Networking
    - VirtualBox Bridge
    - VirtualBox Host-C
  - VirtualBox Python 2.x Su

Oracle VM VirtualBox 4.1.2 application.

This feature requires 129MB on your hard drive. It has 3 of 3 subfeatures selected. The subfeatures require 936KB on yo...

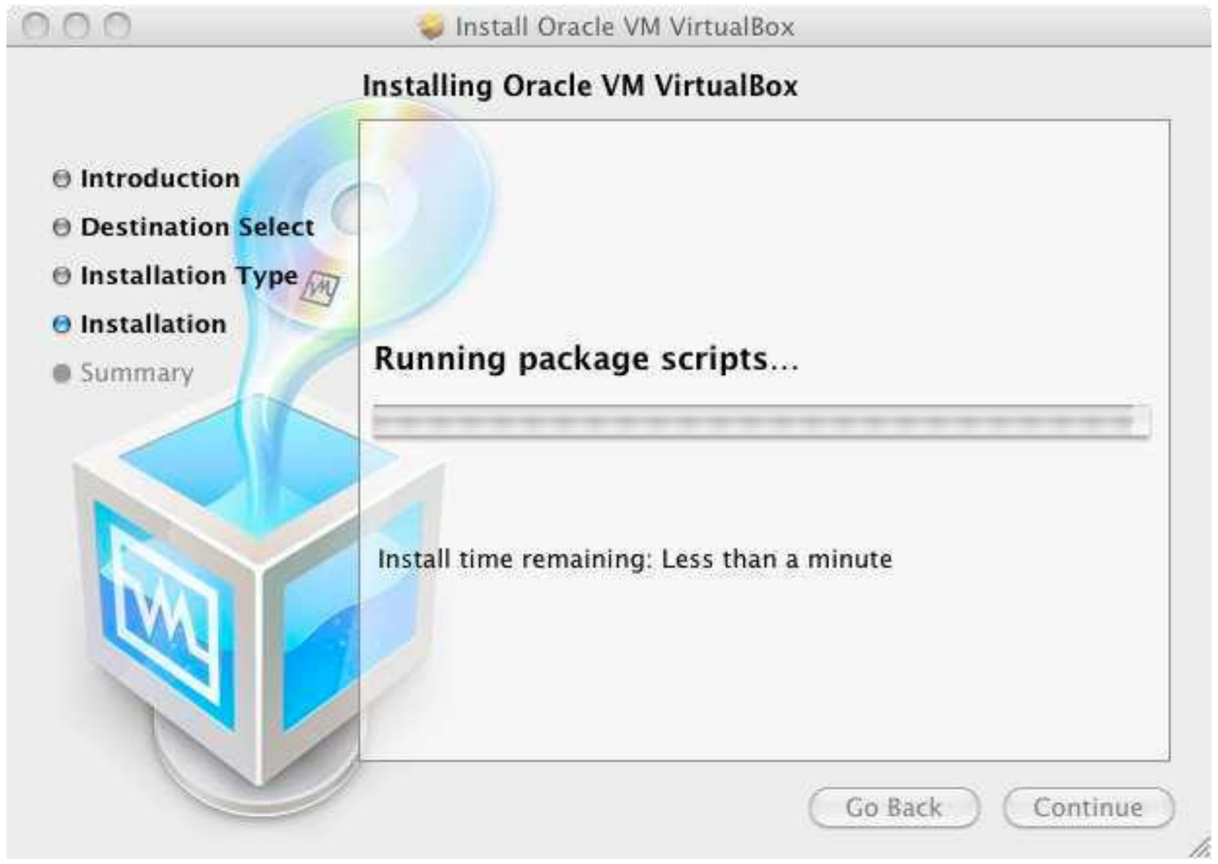
Location: C:₩Program Files₩Oracle₩VirtualBox₩

Version 4.1.2

Install VirtualBox

4. Click continue and finish installing VirtualBox

MAC


Install Oracle VM VirtualBox

Installing Oracle VM VirtualBox

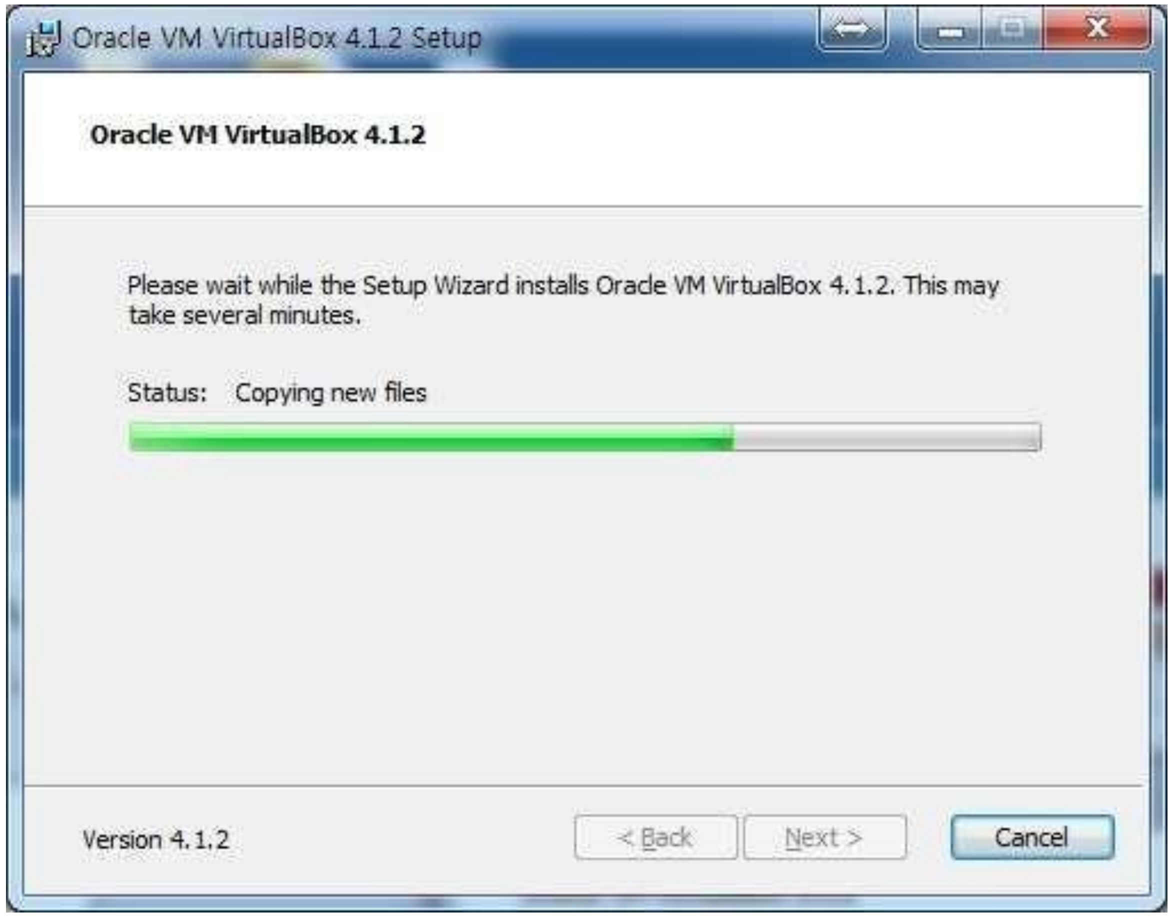
- Introduction
- Destination Select
- Installation Type
- Installation
- Summary

Running package scripts…

Install time remaining: Less than a minute

Go Back Continue

PC


Oracle VM VirtualBox 4.1.2 Setup

Oracle VM VirtualBox 4.1.2

Please wait while the Setup Wizard installs Oracle VM VirtualBox 4.1.2. This may take several minutes.

Status: Copying new files

Version 4.1.2

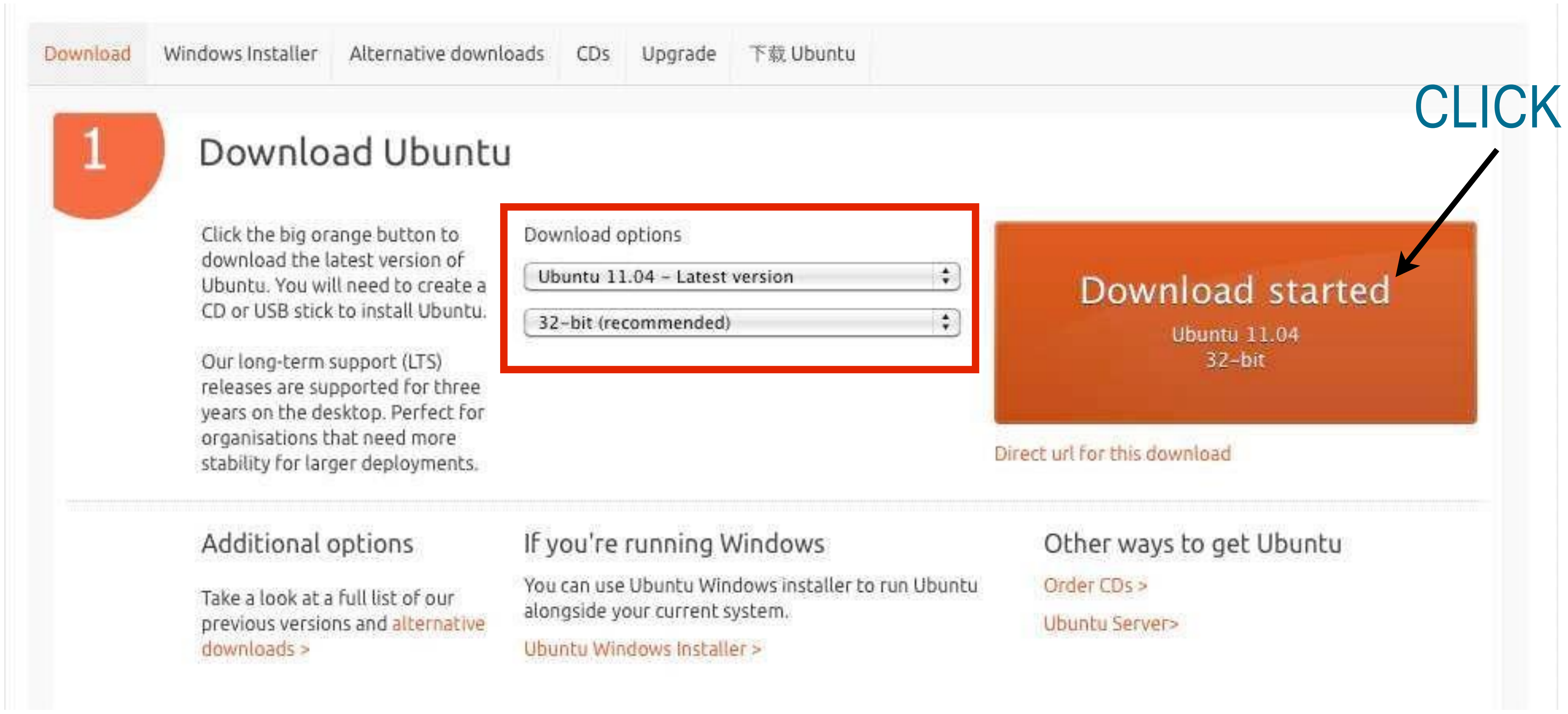
< Back Next > Cancel

5. When finished installation, close the window.

Download Linux

I. Visit the page http://www.ubuntu.com/download/ubuntu/download

2. Choose the Latest version of Ubuntu and 32-bit and click "Start Download"



Download Ubuntu

Click the big orange button to download the latest version of Ubuntu. You will need to create a CD or USB stick to install Ubuntu.

Our long-term support (LTS) releases are supported for three years on the desktop. Perfect for organisations that need more stability for larger deployments.

Download options

Ubuntu 11.04 – Latest version

32-bit (recommended)

Additional options

Take a look at a full list of our previous versions and alternative downloads >

If you're running Windows

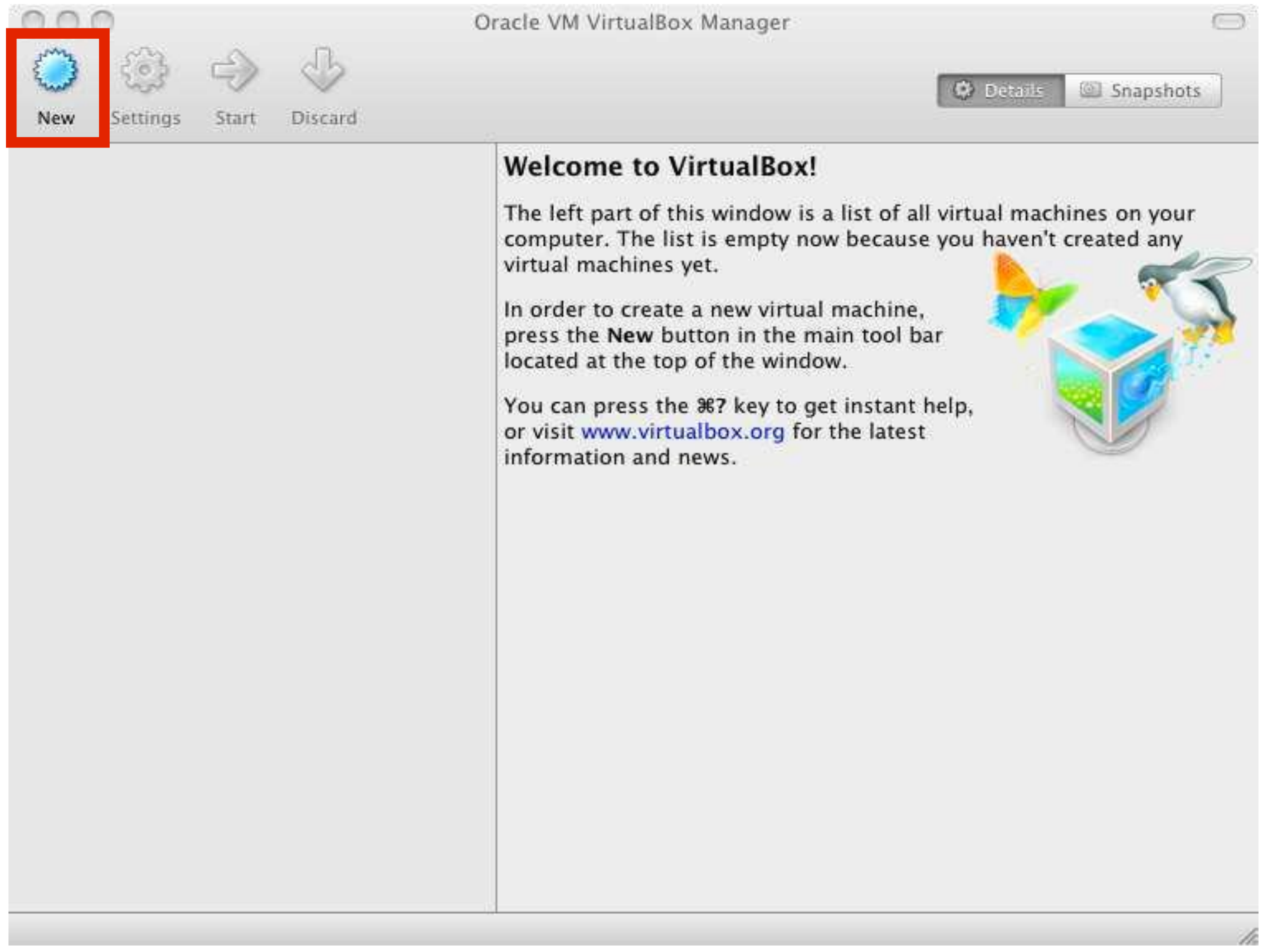
You can use Ubuntu Windows installer to run Ubuntu alongside your current system.

Other ways to get Ubuntu

Install Linux using Virtual Box

1. Run VirtualBox by double-clicking the icon
2. Click “New” button on the top left corner

MAC



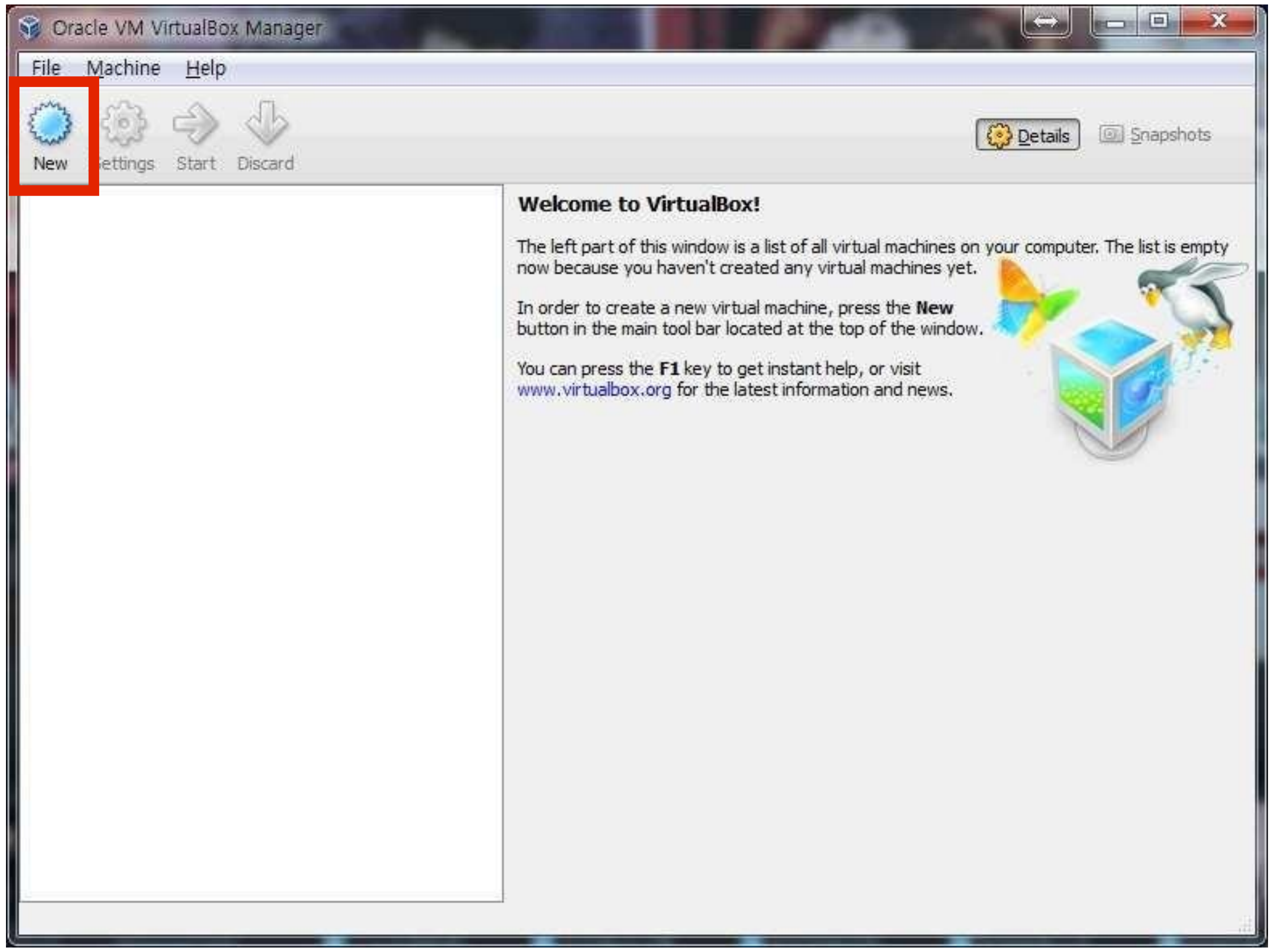
Welcome to VirtualBox!

The left part of this window is a list of all virtual machines on your computer. The list is empty now because you haven't created any virtual machines yet.

In order to create a new virtual machine, press the **New** button in the main tool bar located at the top of the window.

You can press the ⌘? key to get instant help, or visit www.virtualbox.org for the latest information and news.

PC



Welcome to VirtualBox!

The left part of this window is a list of all virtual machines on your computer. The list is empty now because you haven't created any virtual machines yet.

In order to create a new virtual machine, press the **New** button in the main tool bar located at the top of the window.

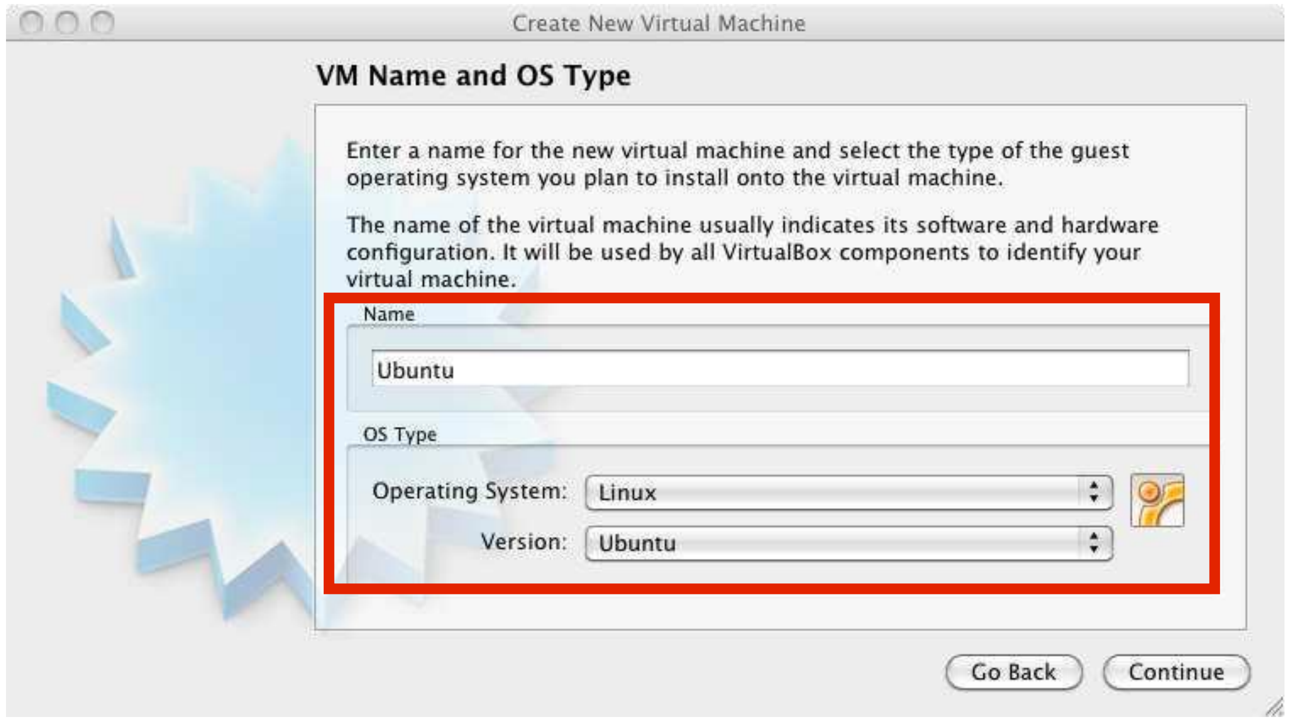
You can press the **F1** key to get instant help, or visit www.virtualbox.org for the latest information and news.

Install Linux using Virtual Box

3. Click “Continue” on the pop-up window

4. Type VM name, select “Linux” for the OS and choose “Ubuntu” for the version.

MAC


Create New Virtual Machine

VM Name and OS Type

Enter a name for the new virtual machine and select the type of the guest operating system you plan to install onto the virtual machine.

The name of the virtual machine usually indicates its software and hardware configuration. It will be used by all VirtualBox components to identify your virtual machine.

Name

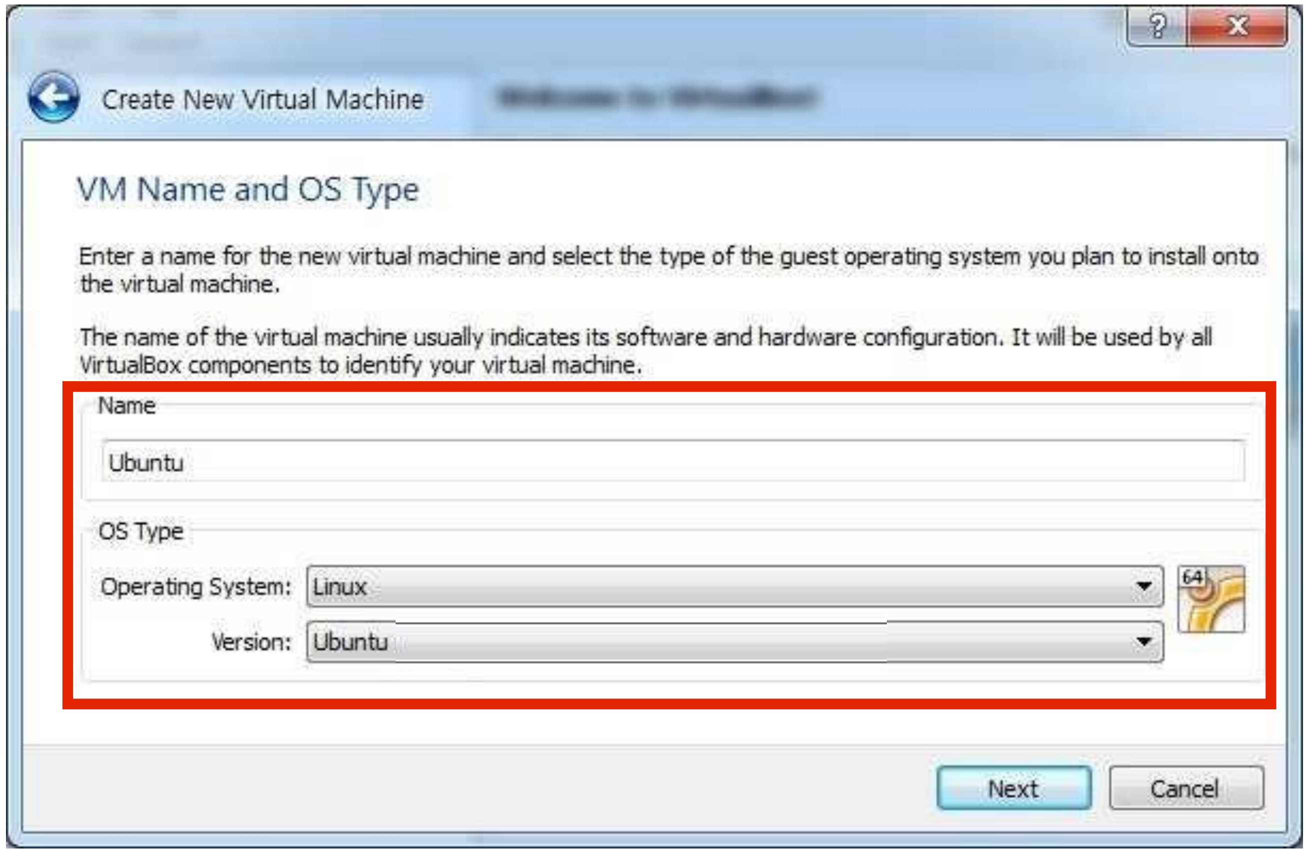
Ubuntu

OS Type

Operating System: Linux

Version: Ubuntu

PC


Create New Virtual Machine

VM Name and OS Type

Enter a name for the new virtual machine and select the type of the guest operating system you plan to install onto the virtual machine.

The name of the virtual machine usually indicates its software and hardware configuration. It will be used by all VirtualBox components to identify your virtual machine.

Name

Ubuntu

OS Type

Operating System: Linux

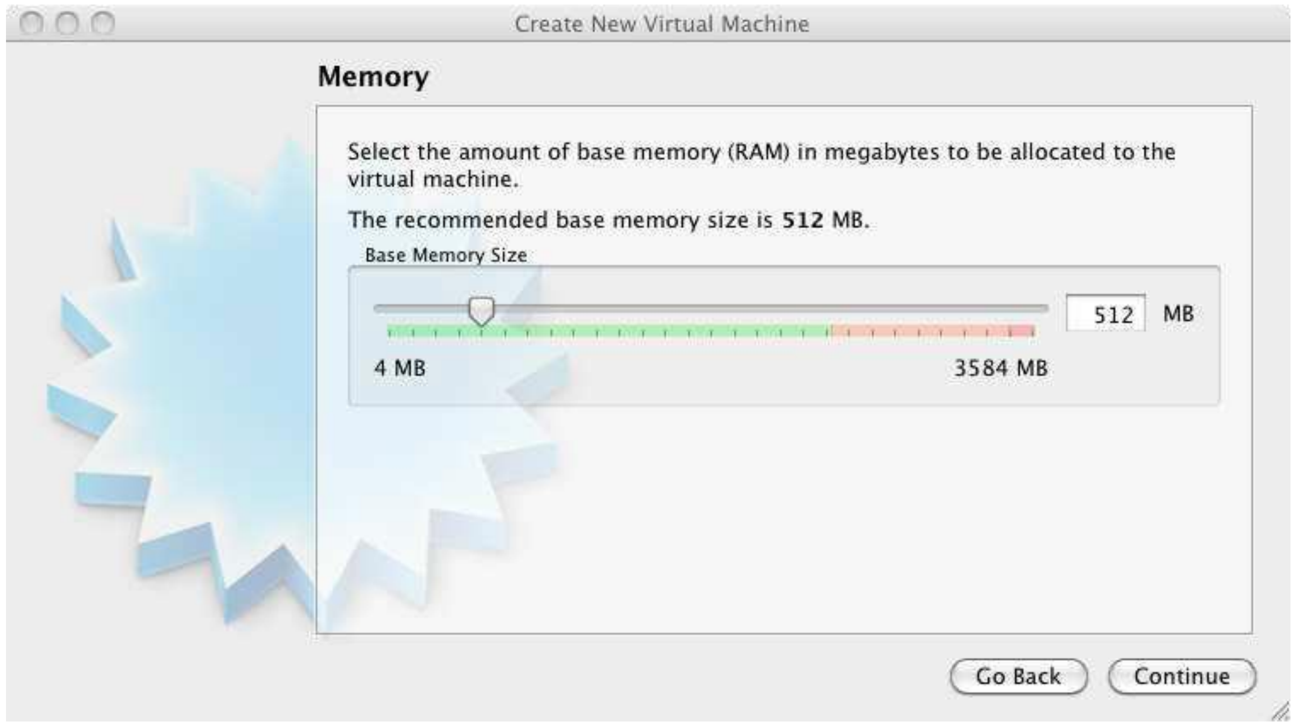
Version: Ubuntu

Install Linux using Virtual Box

5. Choose the amount of memory to allocate (I suggest choosing between 5l2 MB to l024 MB)

6. Click Continue or Next

MAC



Create New Virtual Machine

Memory

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

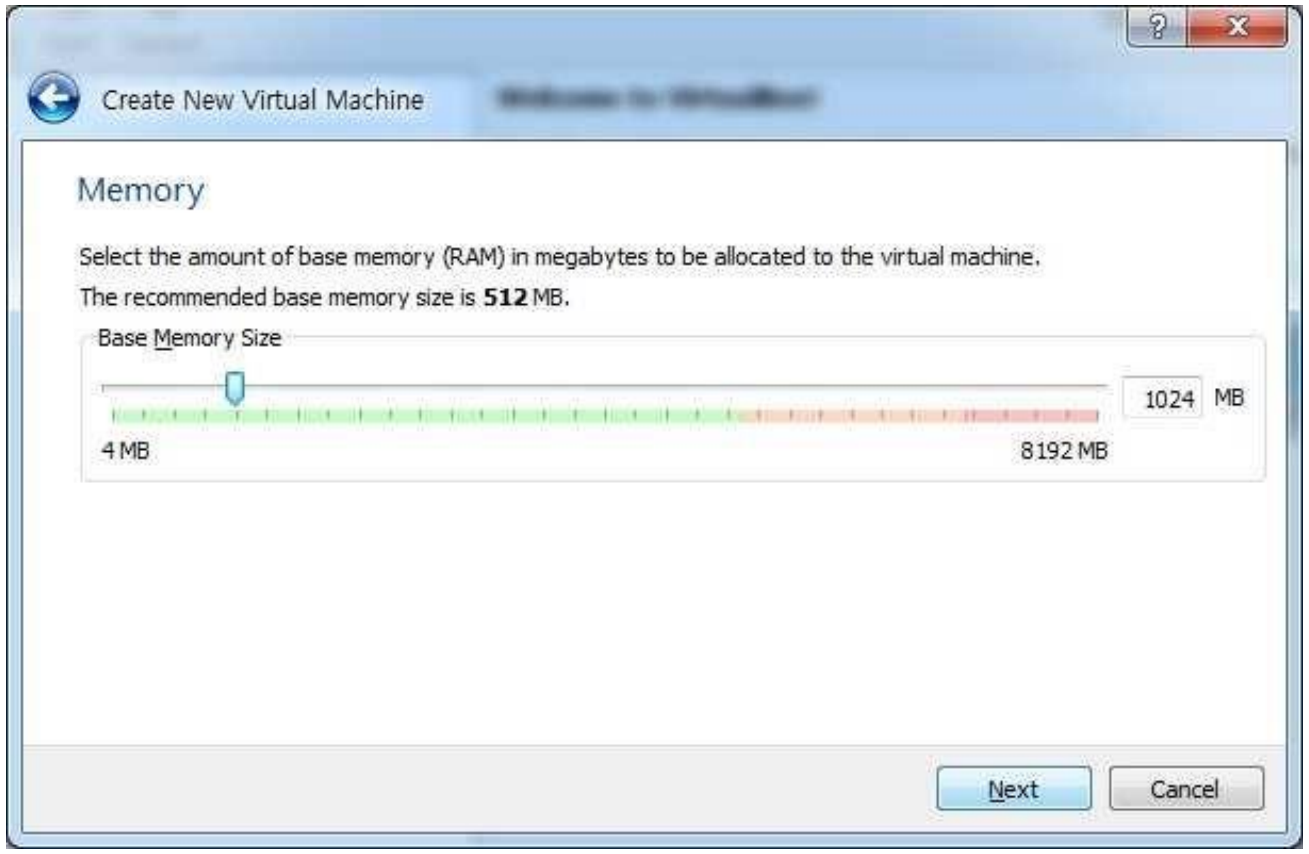
The recommended base memory size is **512** MB.

Base Memory Size

512 MB

4 MB 3584 MB

PC



Create New Virtual Machine

Memory

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

The recommended base memory size is **512** MB.

Base Memory Size

1024 MB

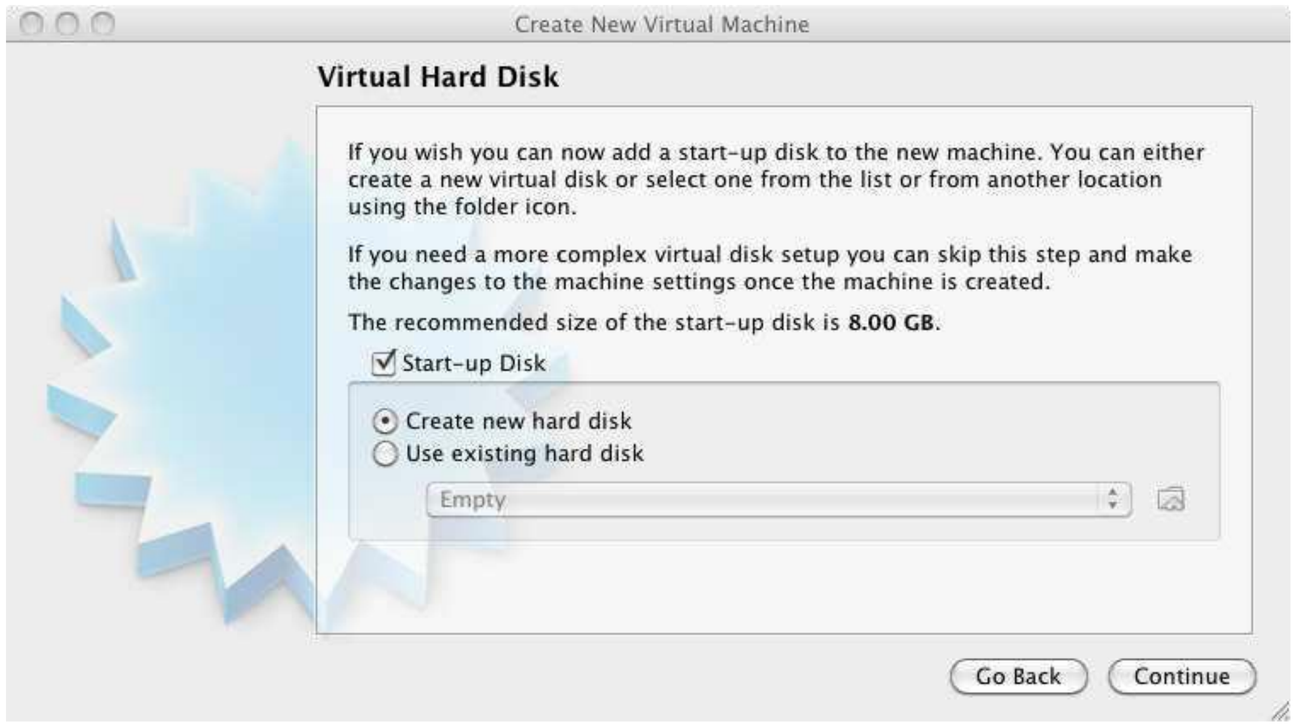
4 MB 8192 MB

Install Linux using Virtual Box

7. Choose create a new virtual hard disk

8. Click Continue or Next

MAC



Virtual Hard Disk

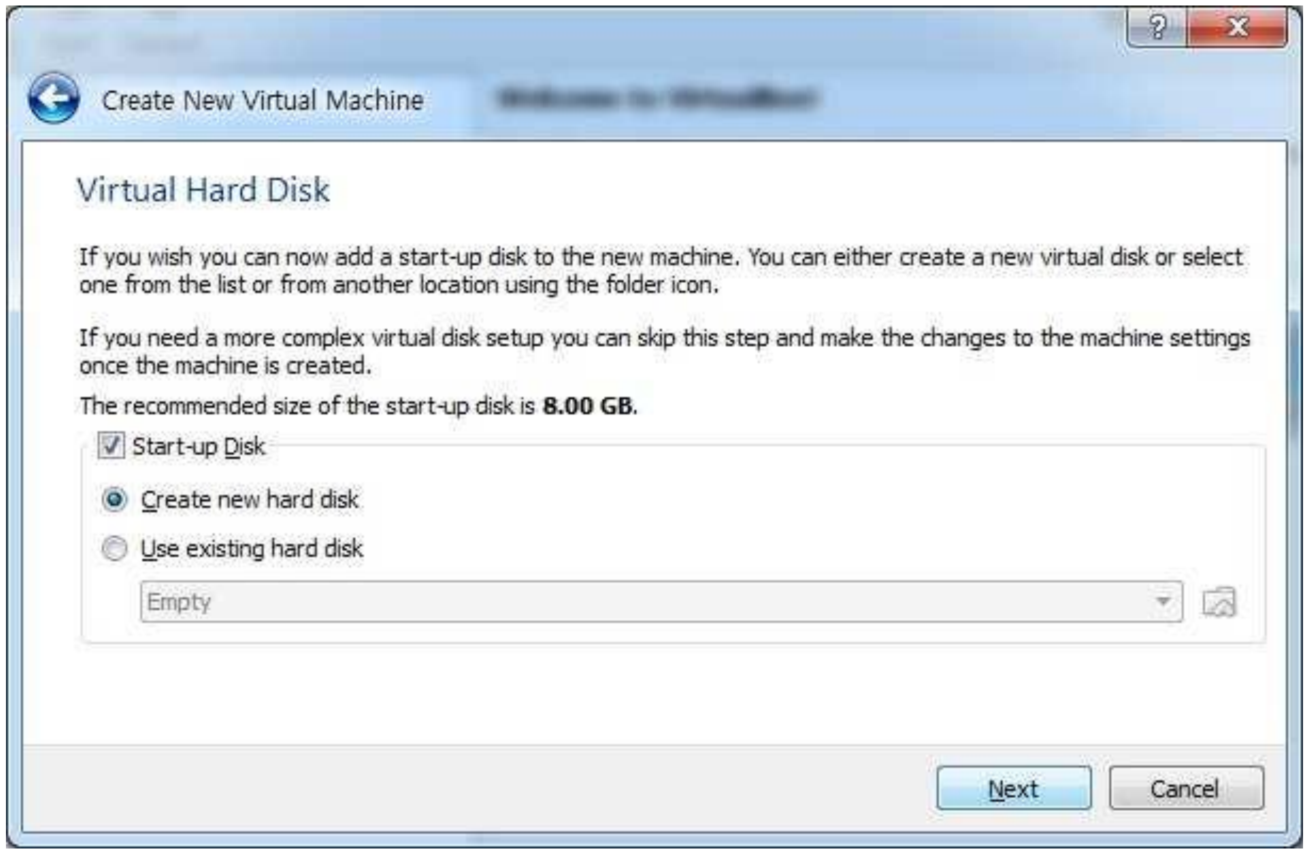
If you wish you can now add a start-up disk to the new machine. You can either create a new virtual disk or select one from the list or from another location using the folder icon.

If you need a more complex virtual disk setup you can skip this step and make the changes to the machine settings once the machine is created.

The recommended size of the start-up disk is **8.00 GB**.

- [✓] Start-up Disk
  - ● Create new hard disk
  - ○ Use existing hard disk
  - Empty

PC



Virtual Hard Disk

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

If you need a more complex virtual disk setup you can skip this step and make the changes to the machine settings once the machine is created.

The recommended size of the start-up disk is **8.00 GB**.

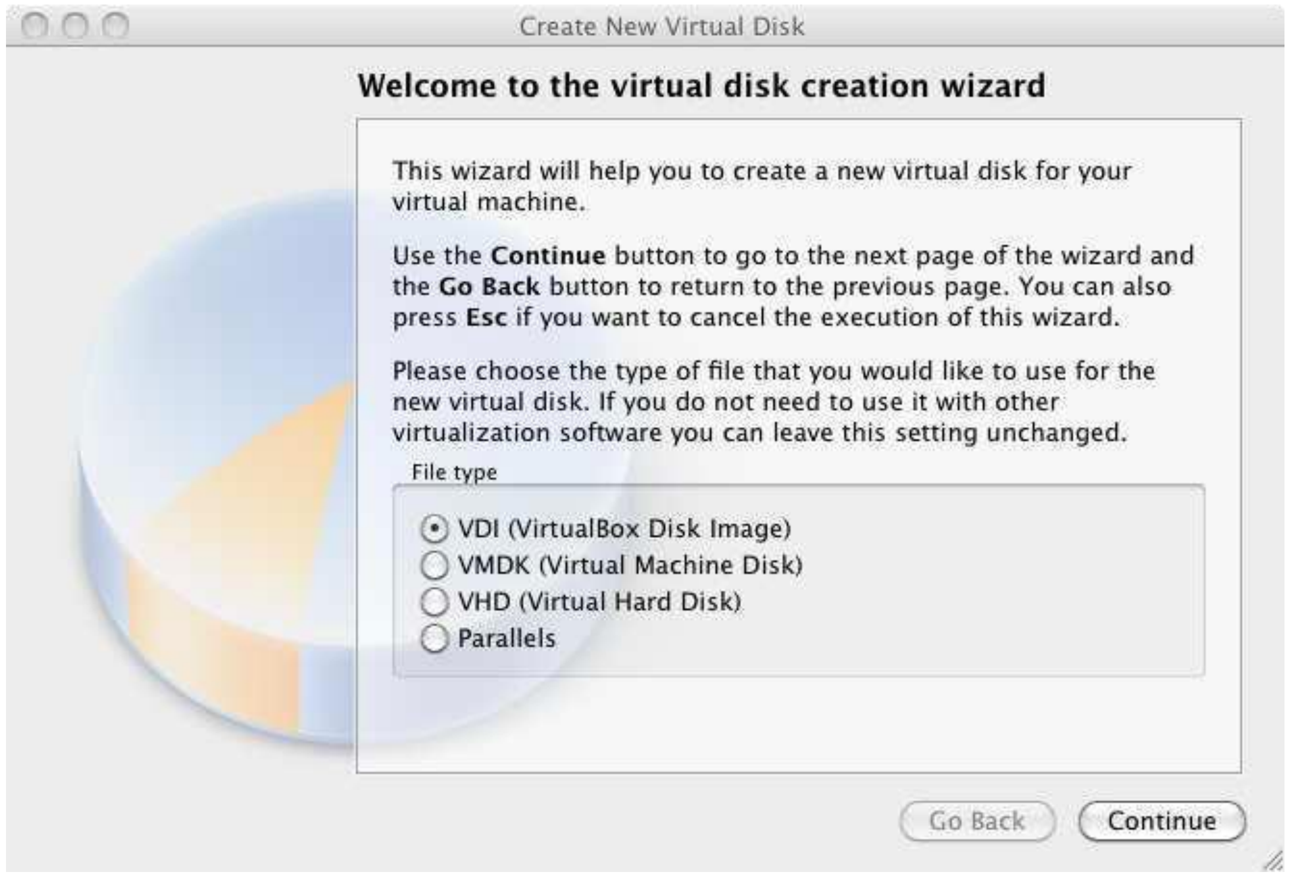
- [✓] Start-up Disk
  - ● Create new hard disk
  - ○ Use existing hard disk
  - Empty

Install Linux using Virtual Box

9. Choose VDI (VirtualBox Disk Image)

I0. Click Continue or Next

MAC



Create New Virtual Disk

Welcome to the virtual disk creation wizard

This wizard will help you to create a new virtual disk for your virtual machine.

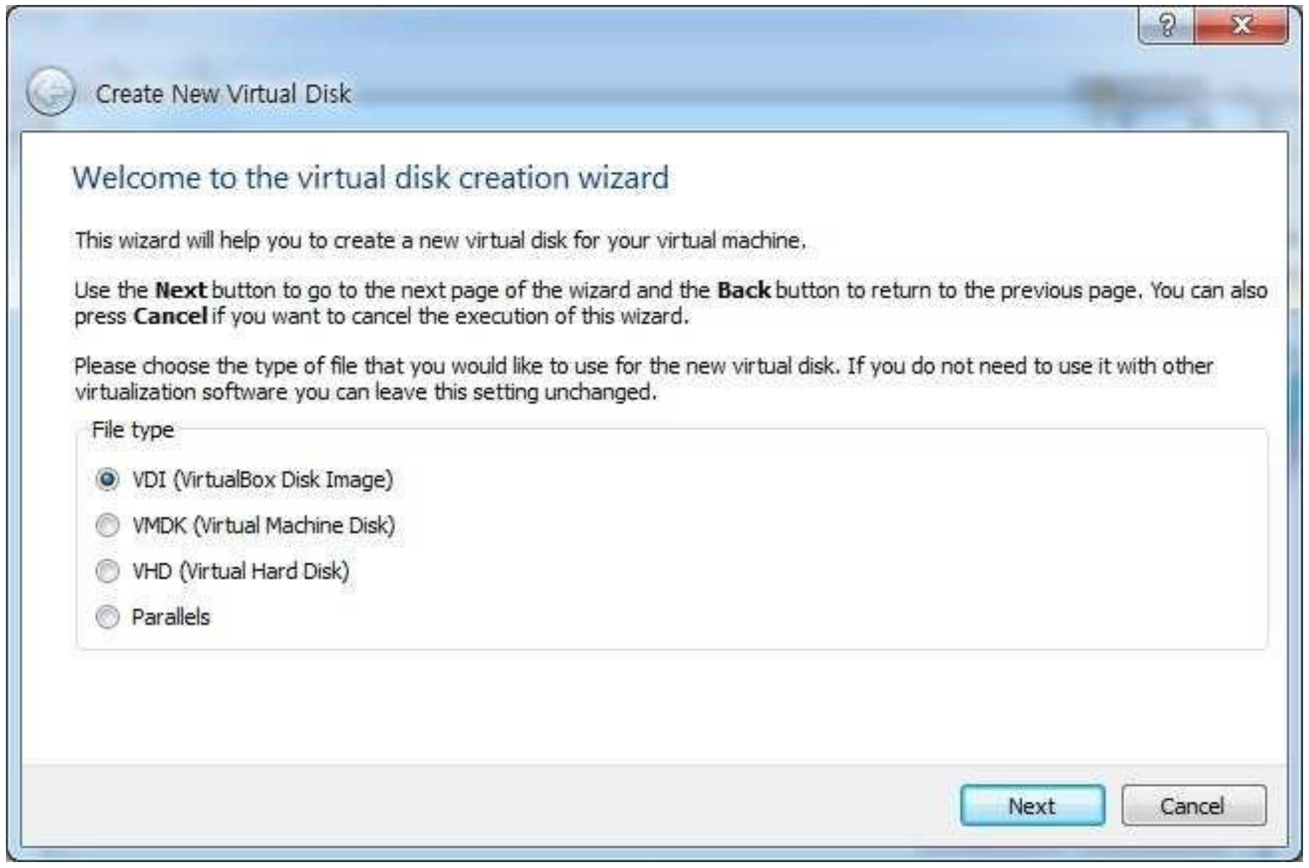
Use the **Continue** button to go to the next page of the wizard and the **Go Back** button to return to the previous page. You can also press **Esc** if you want to cancel the execution of this wizard.

Please choose the type of file that you would like to use for the new virtual disk. If you do not need to use it with other virtualization software you can leave this setting unchanged.

File type

[✓ VDI (VirtualBox Disk Image)]
VMDK (Virtual Machine Disk)
VHD (Virtual Hard Disk)
Parallels

PC



Create New Virtual Disk

Welcome to the virtual disk creation wizard

This wizard will help you to create a new virtual disk for your virtual machine.

Use the **Next** button to go to the next page of the wizard and the **Back** button to return to the previous page. You can also press **Cancel** if you want to cancel the execution of this wizard.

Please choose the type of file that you would like to use for the new virtual disk. If you do not need to use it with other virtualization software you can leave this setting unchanged.

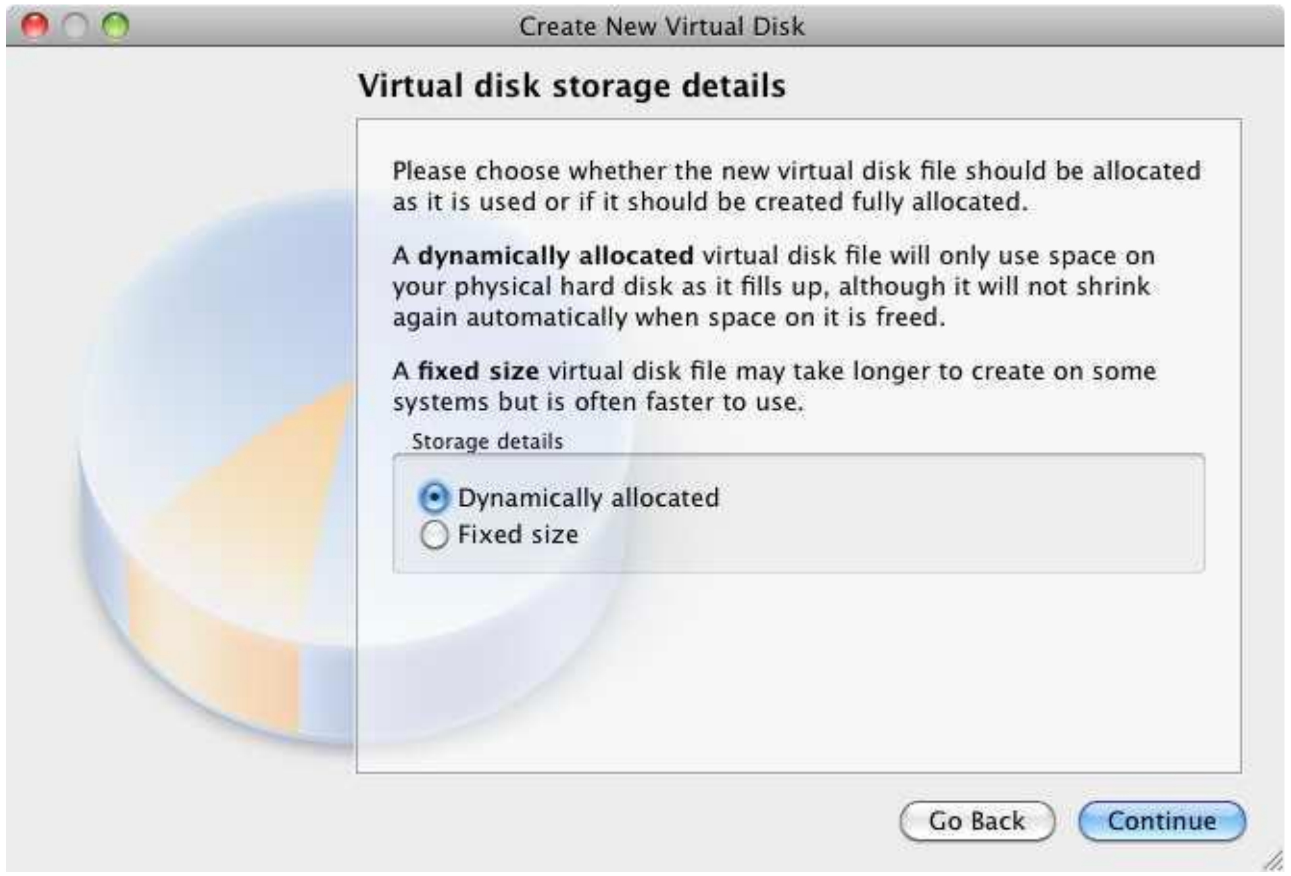
File type

[✓ VDI (VirtualBox Disk Image)]
VMDK (Virtual Machine Disk)
VHD (Virtual Hard Disk)
Parallels

Install Linux using Virtual Box

II. Choose "Dynamically Allocated" click continue. This way, the size of your Virtual Hard Disk will grow as you use.

MAC



Create New Virtual Disk

Virtual disk storage details

Please choose whether the new virtual disk file should be allocated as it is used or if it should be created fully allocated.

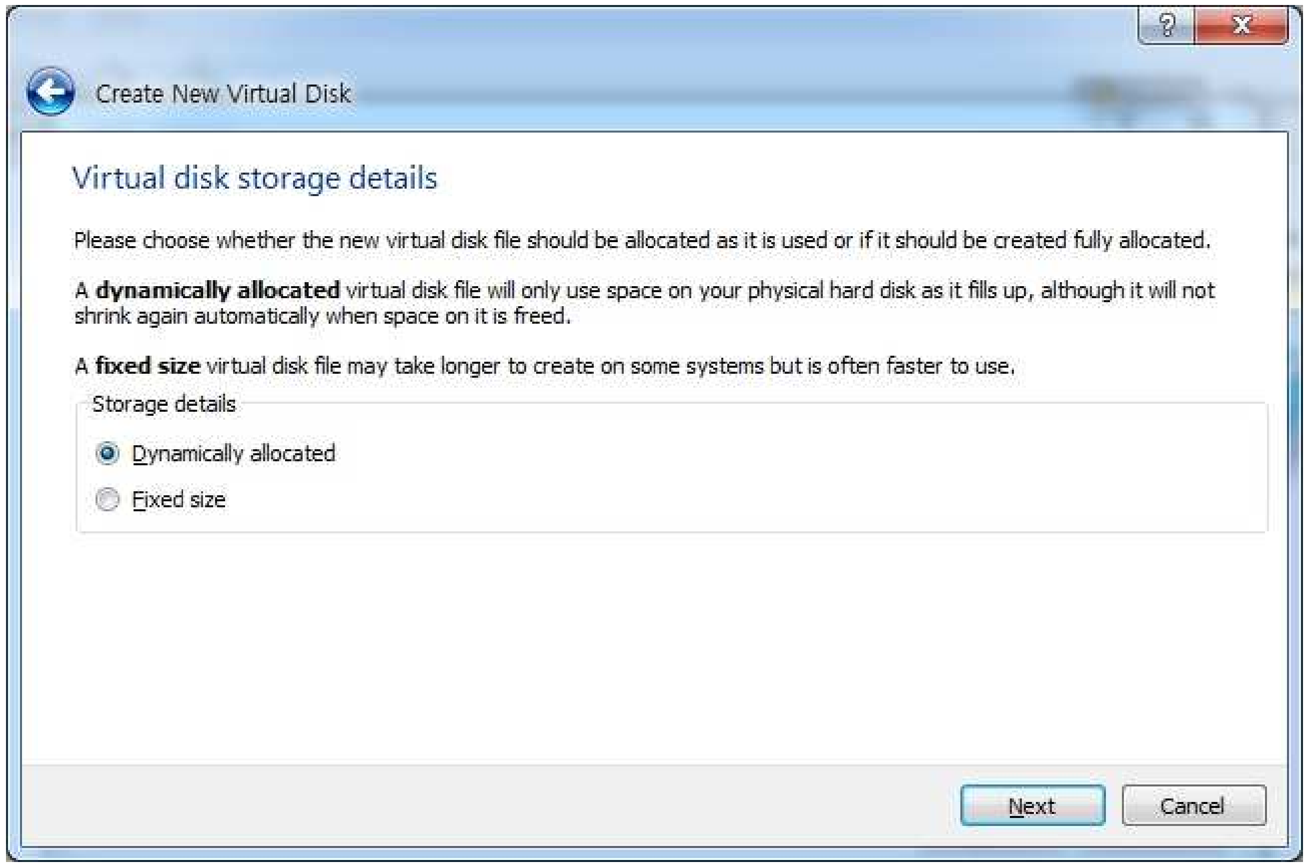
A **dynamically allocated** virtual disk file will only use space on your physical hard disk as it fills up, although it will not shrink again automatically when space on it is freed.

A **fixed size** virtual disk file may take longer to create on some systems but is often faster to use.

Storage details

(●) Dynamically allocated
( ) Fixed size

PC



Create New Virtual Disk

Virtual disk storage details

Please choose whether the new virtual disk file should be allocated as it is used or if it should be created fully allocated.

A **dynamically allocated** virtual disk file will only use space on your physical hard disk as it fills up, although it will not shrink again automatically when space on it is freed.

A **fixed size** virtual disk file may take longer to create on some systems but is often faster to use.

Storage details

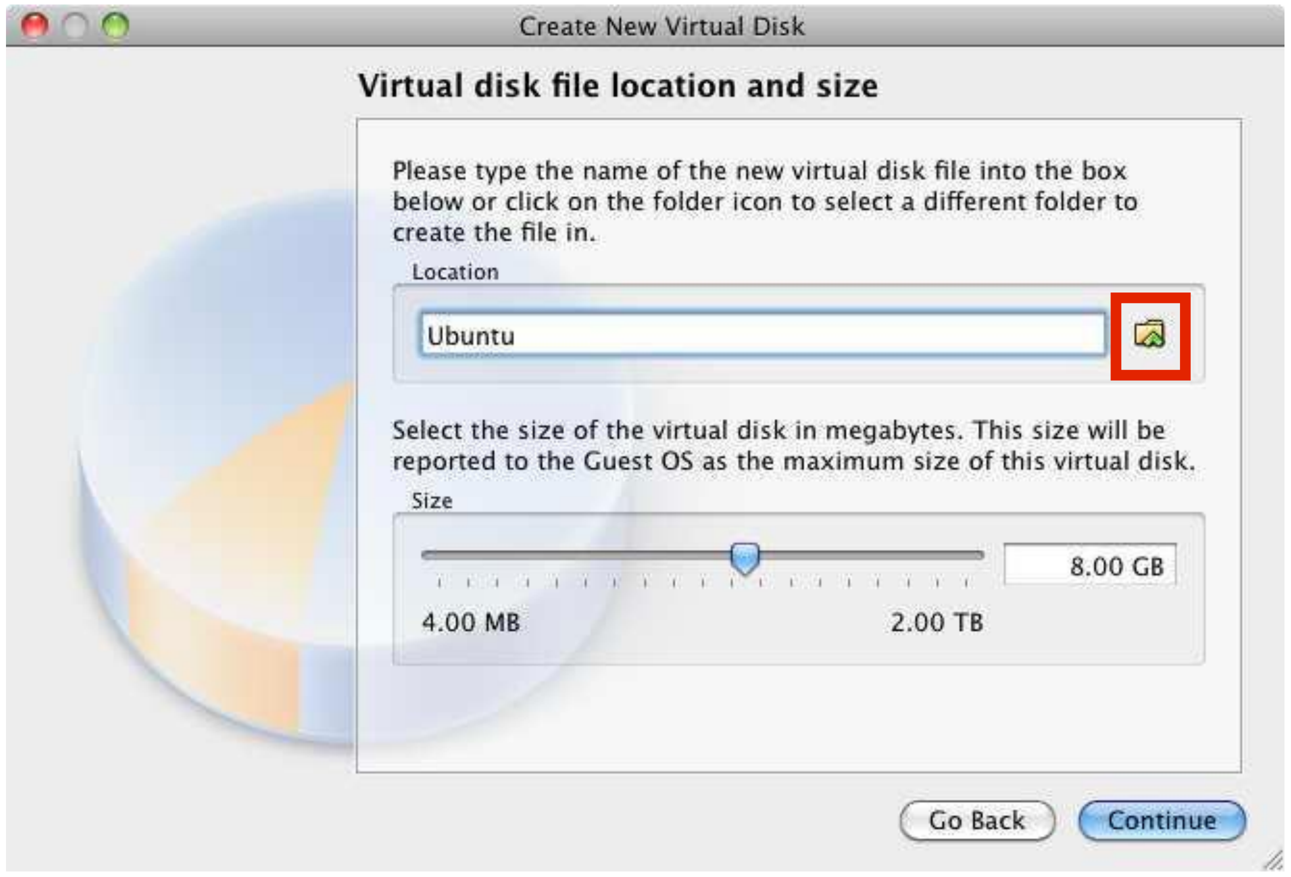
(●) Dynamically allocated
( ) Fixed size

Install Linux using Virtual Box

12. Click the folder icon and choose the ubuntu iso file you downloaded.

13. Select the size of the Virtual Disk (I recommend choosing 8 GB) and click continue

MAC



Create New Virtual Disk

Virtual disk file location and size

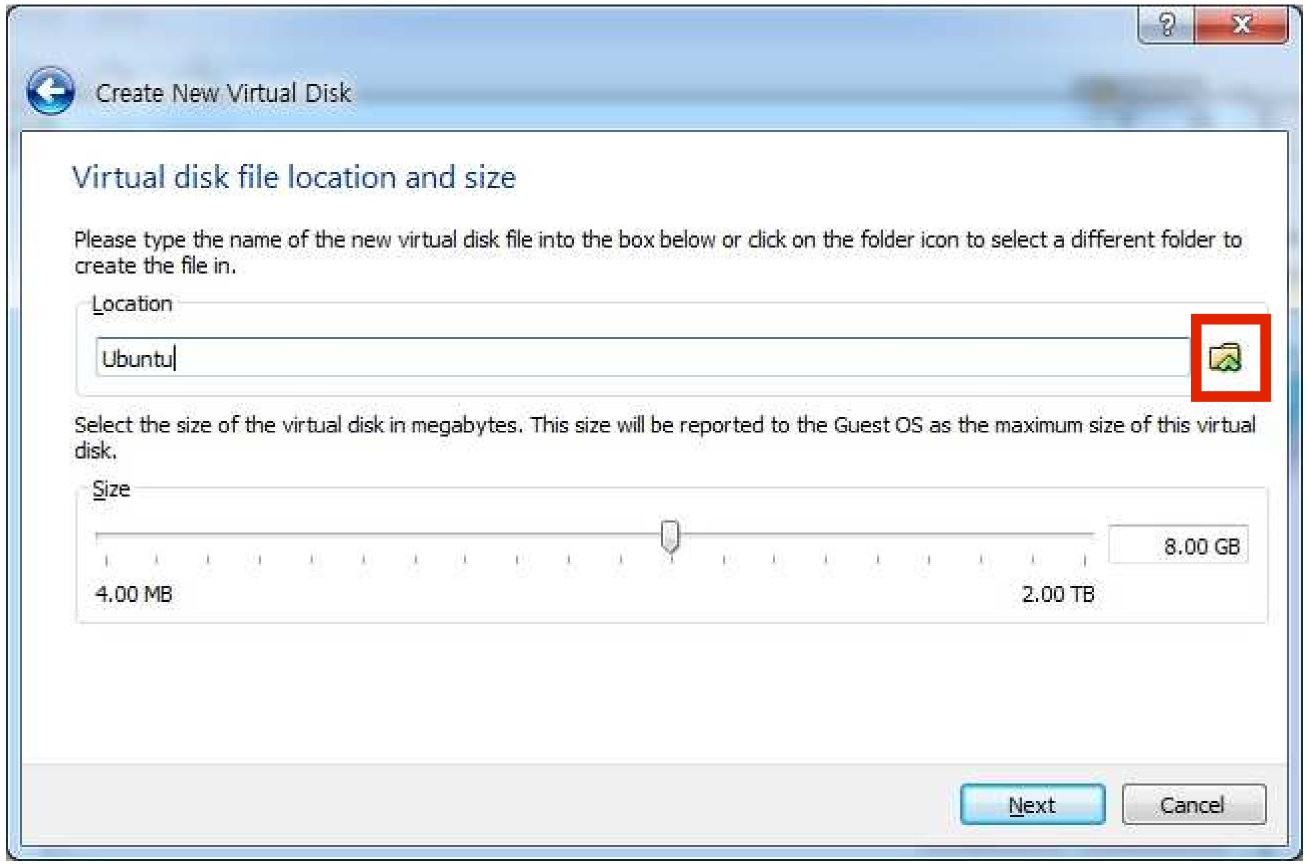
Please type the name of the new virtual disk file into the box below or click on the folder icon to select a different folder to create the file in.

Location: Ubuntu

Select the size of the virtual disk in megabytes. This size will be reported to the Guest OS as the maximum size of this virtual disk.

Size: 8.00 GB (4.00 MB – 2.00 TB)

PC



Create New Virtual Disk

Virtual disk file location and size

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

Location: Ubuntu

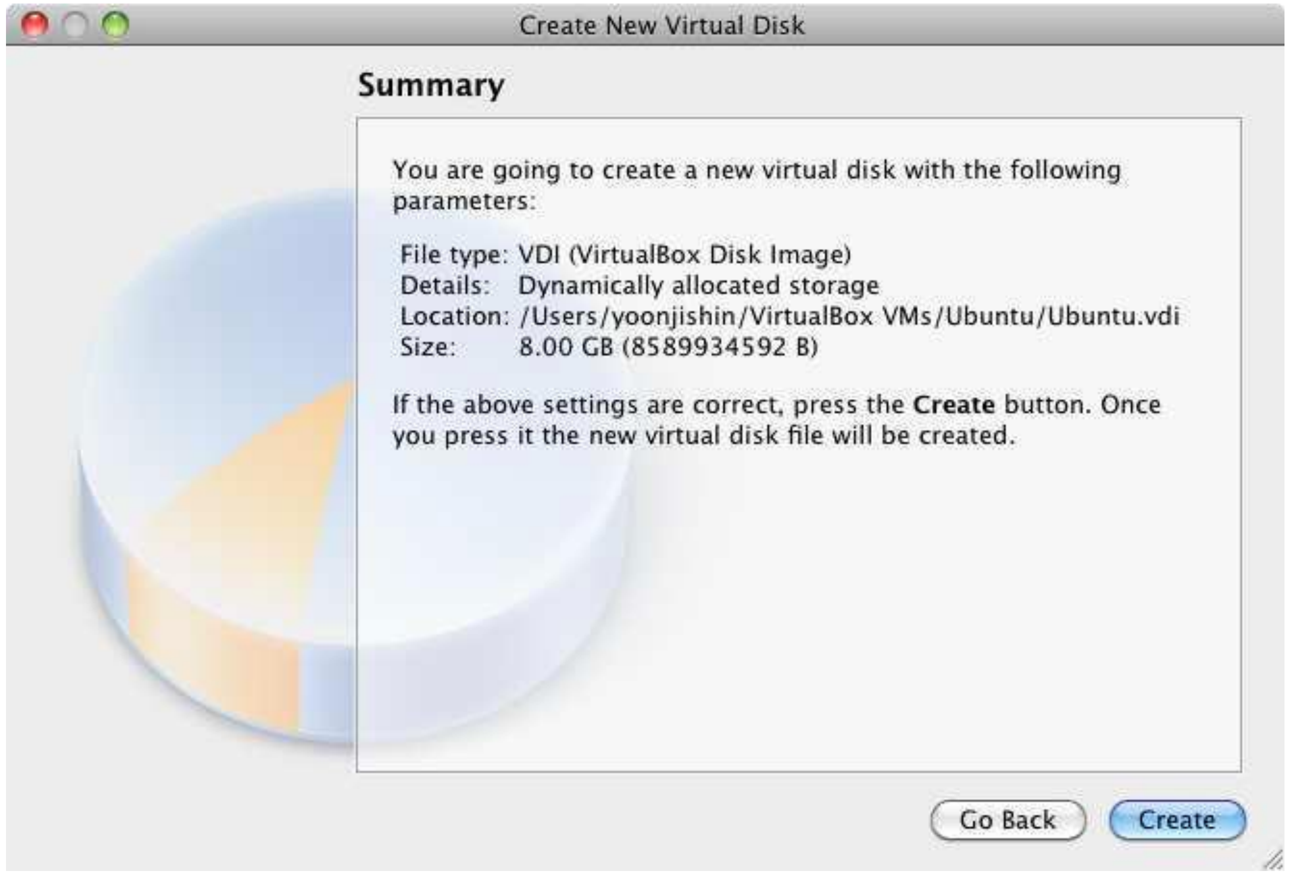
Select the size of the virtual disk in megabytes. This size will be reported to the Guest OS as the maximum size of this virtual disk.

Size: 8.00 GB (4.00 MB – 2.00 TB)

Install Linux using Virtual Box

I4. Click Create

MAC



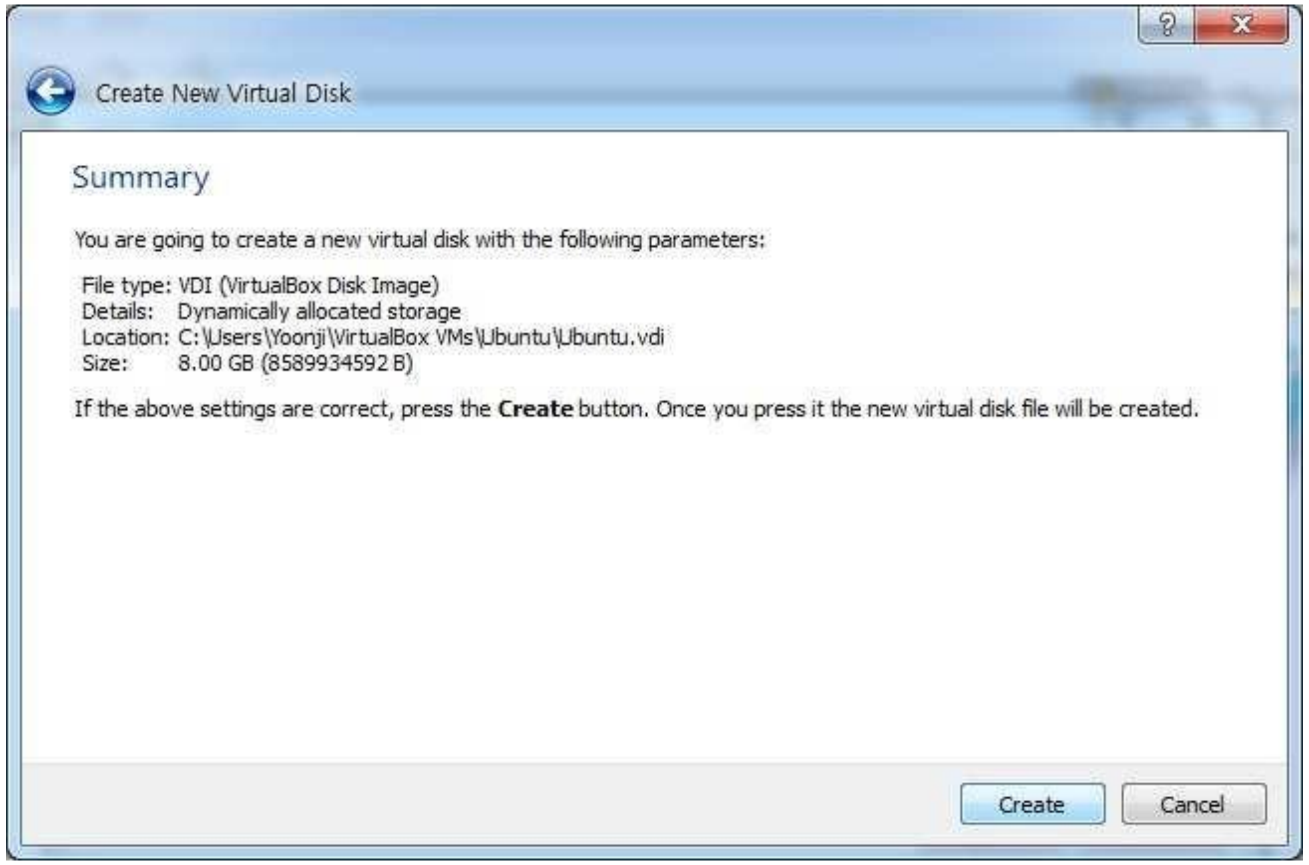
Summary

You are going to create a new virtual disk with the following parameters:

File type: VDI (VirtualBox Disk Image)
Details: Dynamically allocated storage
Location: /Users/yoonjishin/VirtualBox VMs/Ubuntu/Ubuntu.vdi
Size: 8.00 GB (8589934592 B)

If the above settings are correct, press the **Create** button. Once you press it the new virtual disk file will be created.

PC



Summary

You are going to create a new virtual disk with the following parameters:

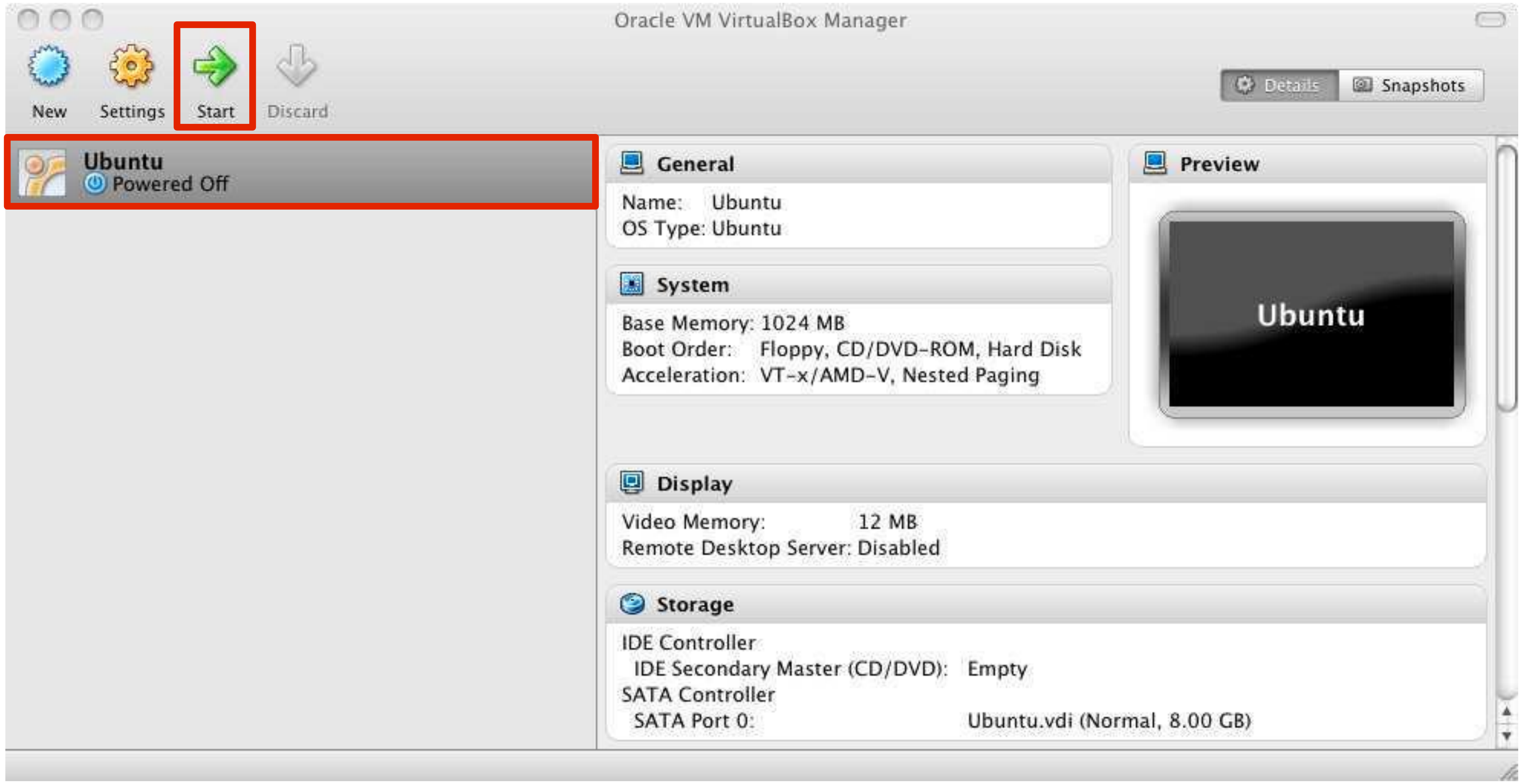
File type: VDI (VirtualBox Disk Image)
Details: Dynamically allocated storage
Location: C:\Users\Yoonji\VirtualBox VMs\Ubuntu\Ubuntu.vdi
Size: 8.00 GB (8589934592 B)

If the above settings are correct, press the **Create** button. Once you press it the new virtual disk file will be created.

Running Linux

I. Choose Ubuntu from left column and click Start

MAC & PC



Name: Ubuntu
OS Type: Ubuntu

Base Memory: 1024 MB
Boot Order: Floppy, CD/DVD-ROM, Hard Disk
Acceleration: VT-x/AMD-V, Nested Paging

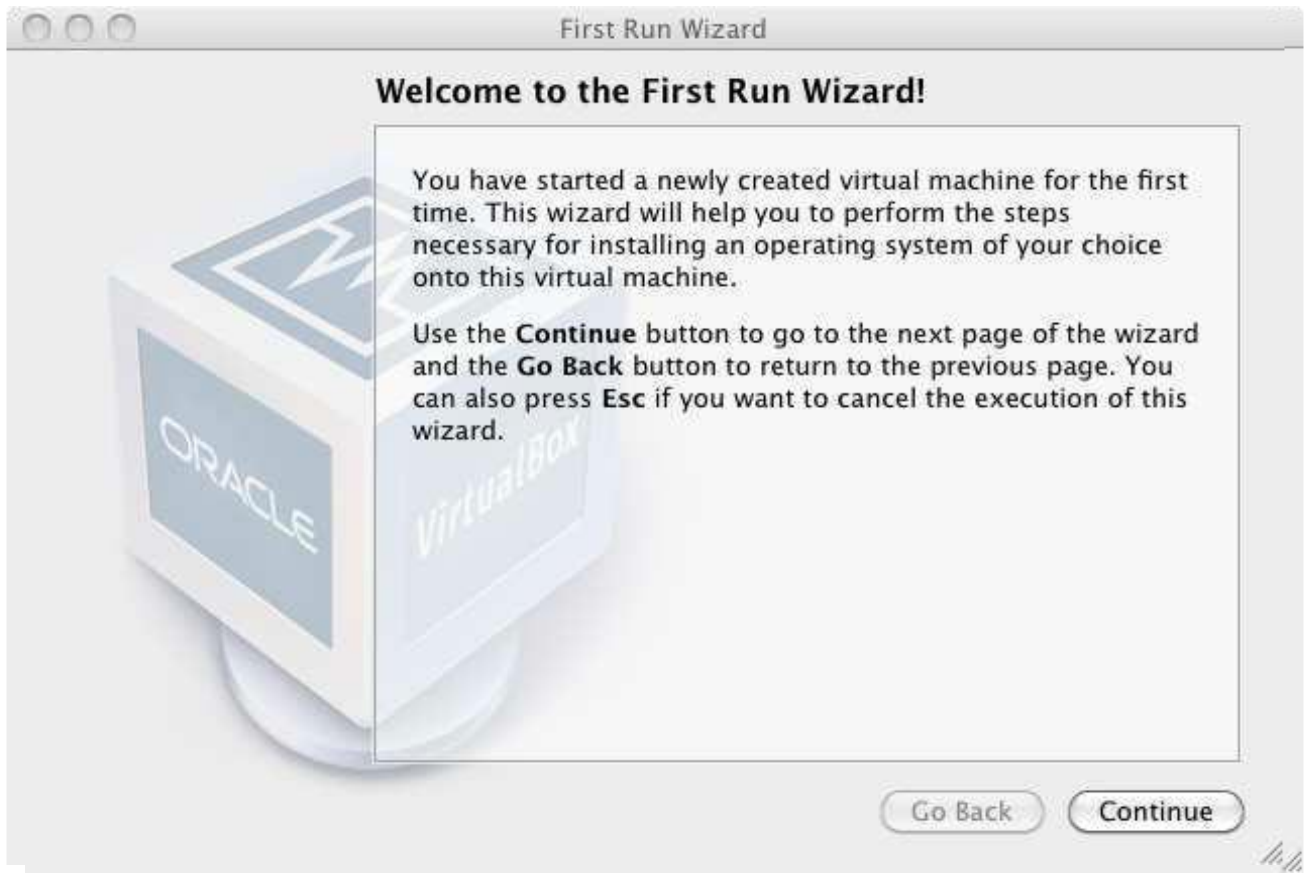
Video Memory: 12 MB
Remote Desktop Server: Disabled

IDE Controller
 IDE Secondary Master (CD/DVD): Empty
SATA Controller
 SATA Port 0: Ubuntu.vdi (Normal, 8.00 GB)

Running Linux

2. Click continue on pop-up window

MAC

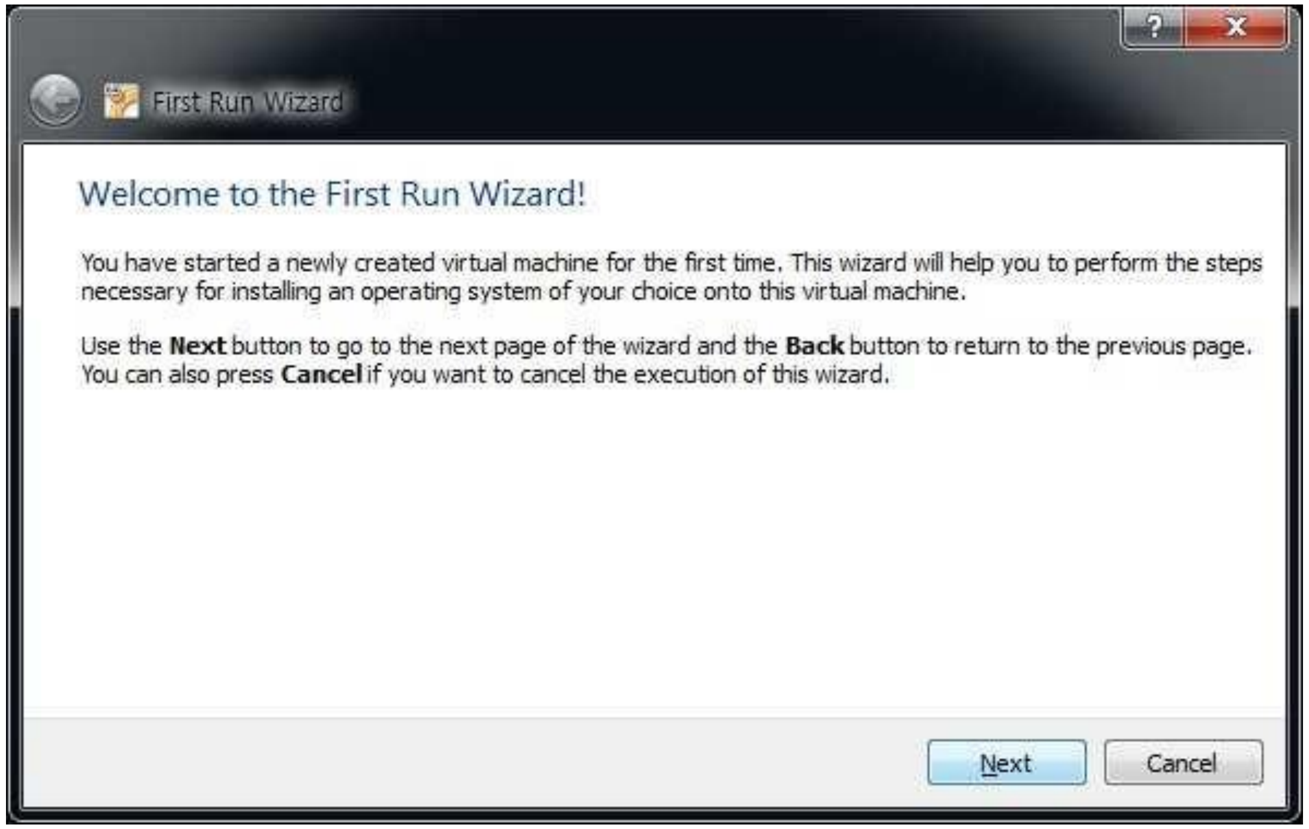


Welcome to the First Run Wizard!

You have started a newly created virtual machine for the first time. This wizard will help you to perform the steps necessary for installing an operating system of your choice onto this virtual machine.

Use the **Continue** button to go to the next page of the wizard and the **Go Back** button to return to the previous page. You can also press **Esc** if you want to cancel the execution of this wizard.

PC



Welcome to the First Run Wizard!

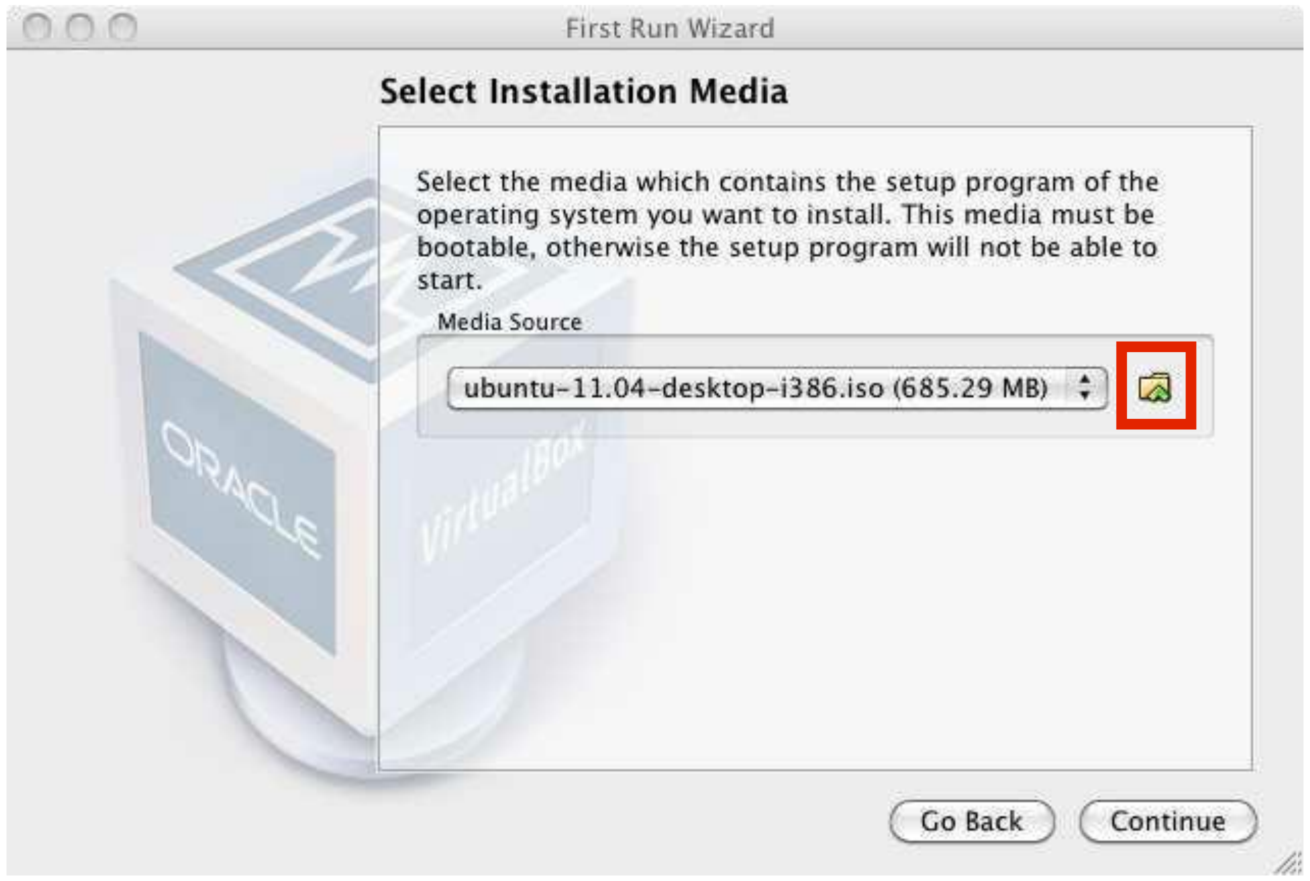
You have started a newly created virtual machine for the first time. This wizard will help you to perform the steps necessary for installing an operating system of your choice onto this virtual machine.

Use the **Next** button to go to the next page of the wizard and the **Back** button to return to the previous page. You can also press **Cancel** if you want to cancel the execution of this wizard.

Running Linux

3. Click the folder icon and choose the ubuntu iso file you downloaded and click continue and start

MAC


First Run Wizard

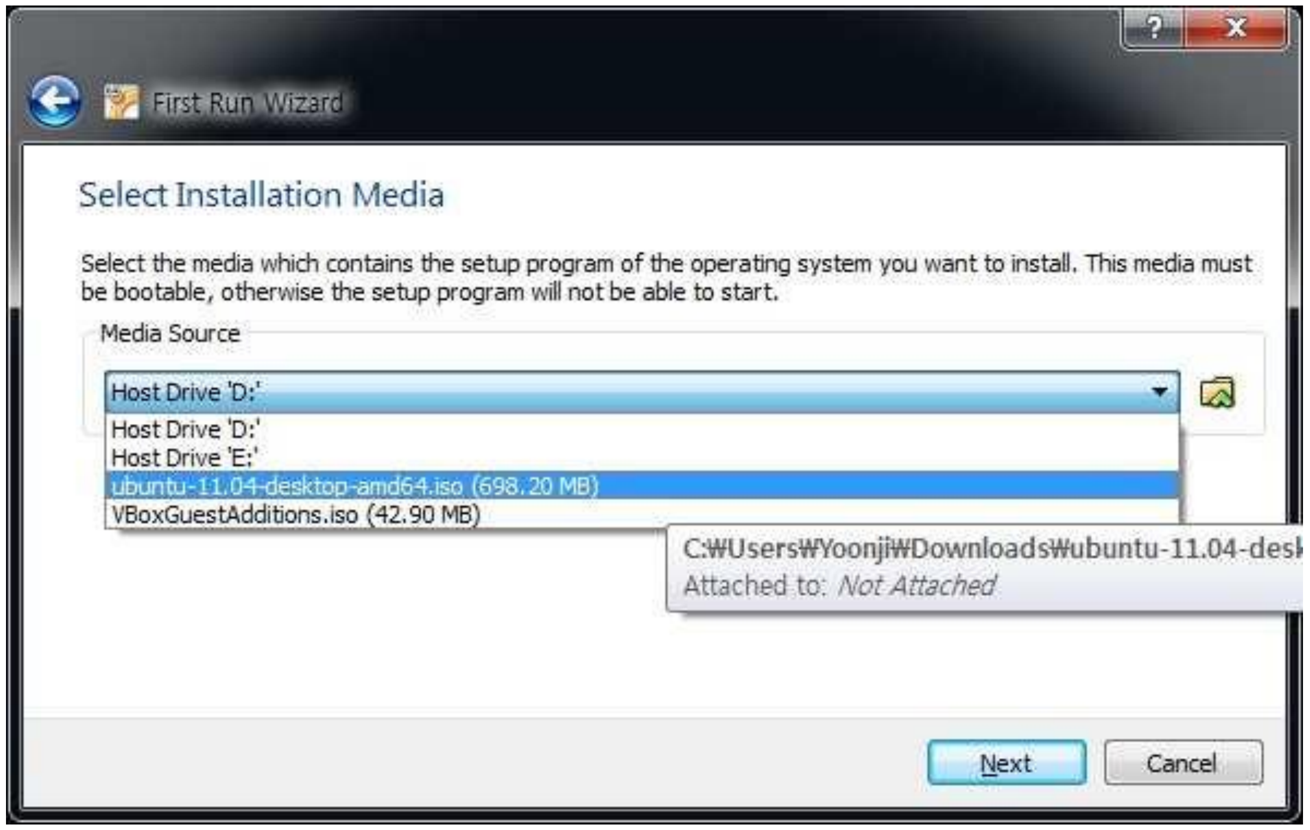
Select Installation Media

Select the media which contains the setup program of the operating system you want to install. This media must be bootable, otherwise the setup program will not be able to start.

Media Source

ubuntu-11.04-desktop-i386.iso (685.29 MB)

PC


First Run Wizard

Select Installation Media

Select the media which contains the setup program of the operating system you want to install. This media must be bootable, otherwise the setup program will not be able to start.

Media Source

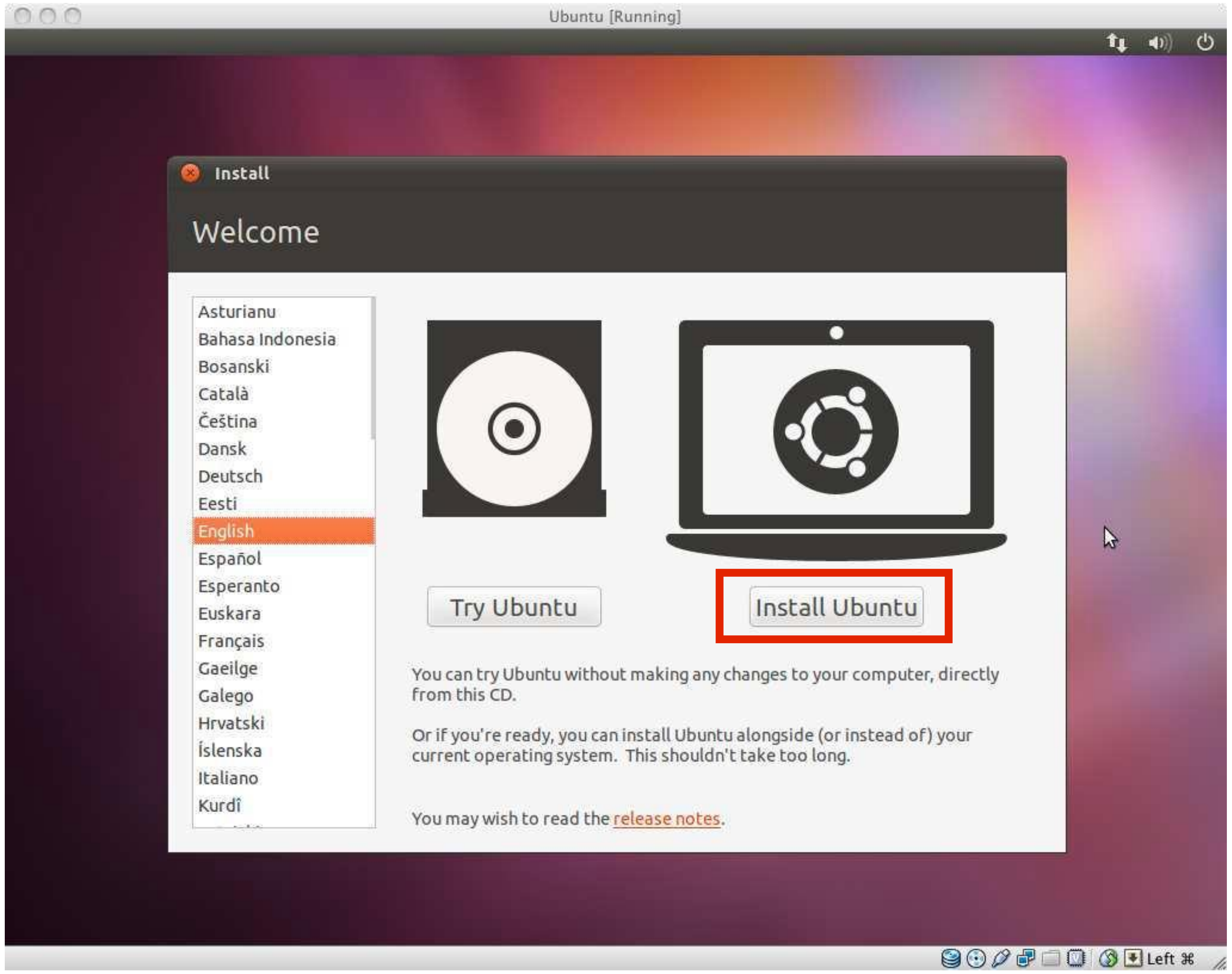
Host Drive 'D:'

Host Drive 'D:'
Host Drive 'E:'
ubuntu-11.04-desktop-amd64.iso (698.20 MB)
VBoxGuestAdditions.iso (42.90 MB)

C:₩Users₩Yoonji₩Downloads₩ubuntu-11.04-desk
Attached to: *Not Attached*

Running Linux

4. Click Install Ubuntu



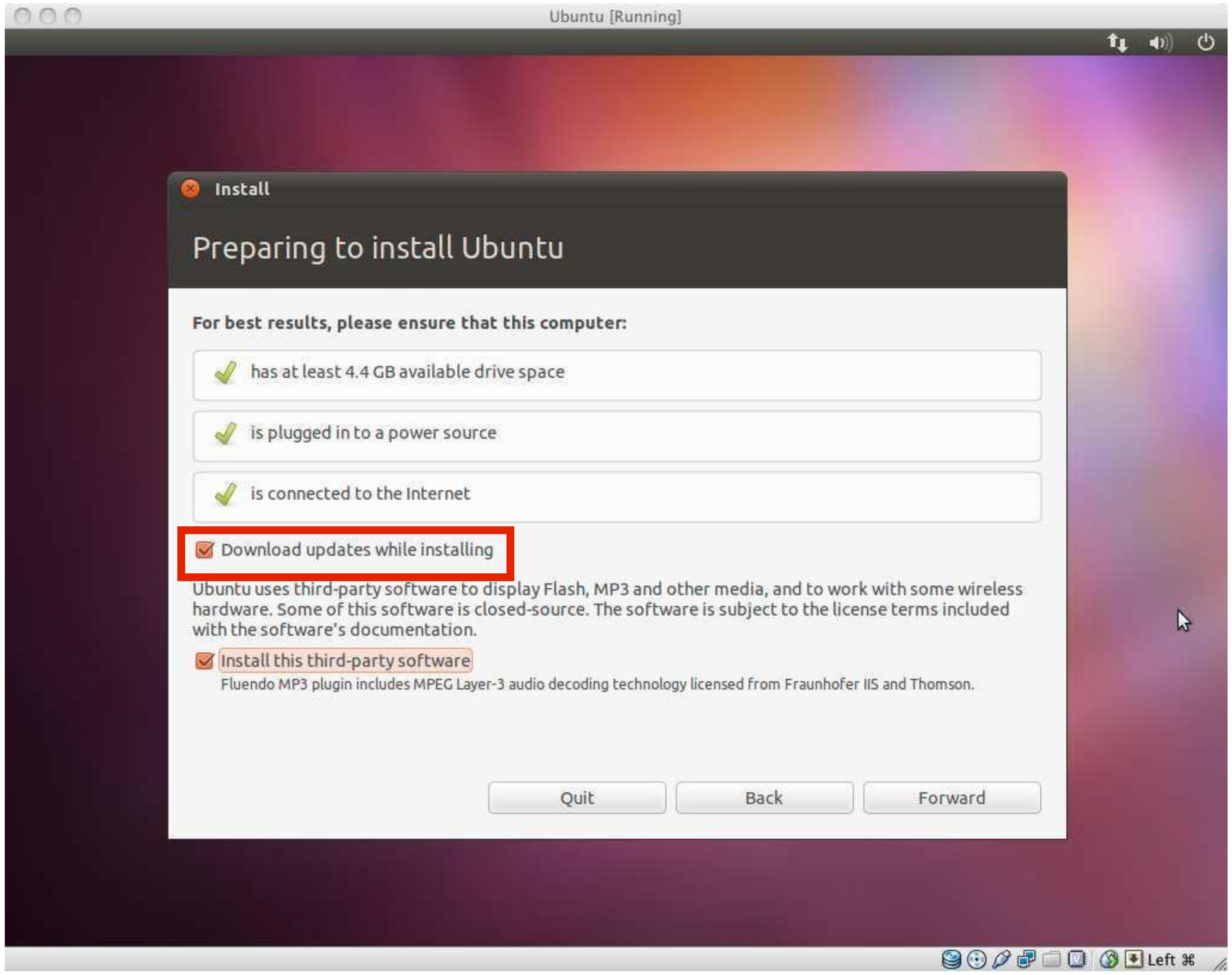
You can try Ubuntu without making any changes to your computer, directly from this CD.

Or if you're ready, you can install Ubuntu alongside (or instead of) your current operating system. This shouldn't take too long.

You may wish to read the release notes.

Running Linux

4. Check “Download updates” and click Forward



Preparing to install Ubuntu

For best results, please ensure that this computer:

- ✓ has at least 4.4 GB available drive space
- ✓ is plugged in to a power source
- ✓ is connected to the Internet

[✓] Download updates while installing

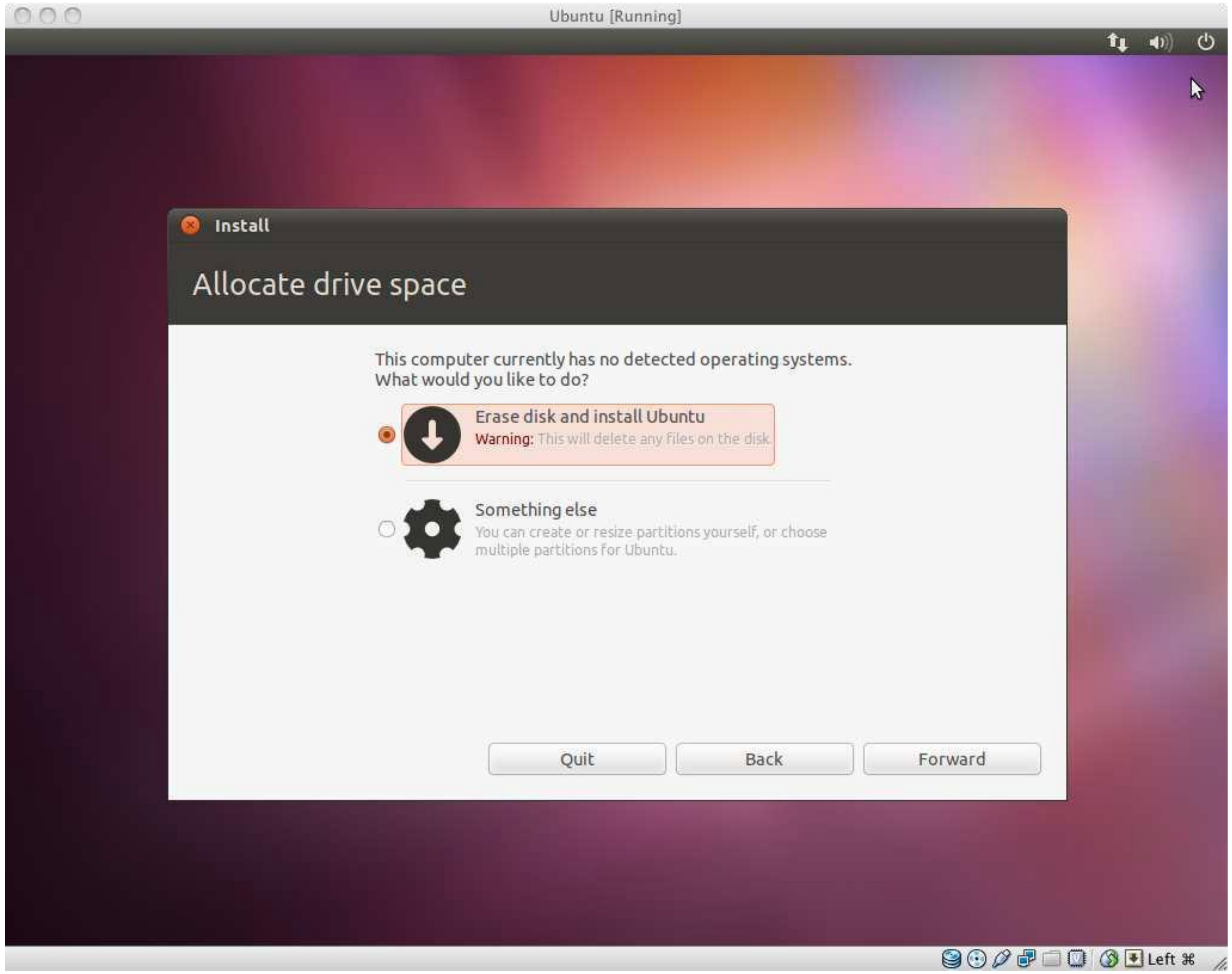
Ubuntu uses third-party software to display Flash, MP3 and other media, and to work with some wireless hardware. Some of this software is closed-source. The software is subject to the license terms included with the software's documentation.

[✓] Install this third-party software

Fluendo MP3 plugin includes MPEG Layer-3 audio decoding technology licensed from Fraunhofer IIS and Thomson.

Running Linux

5. Choose “Erase disk and install Ubuntu” and click Forward (Don’t worry, it won’t wipe your computer)



Ubuntu [Running]

Install

Allocate drive space

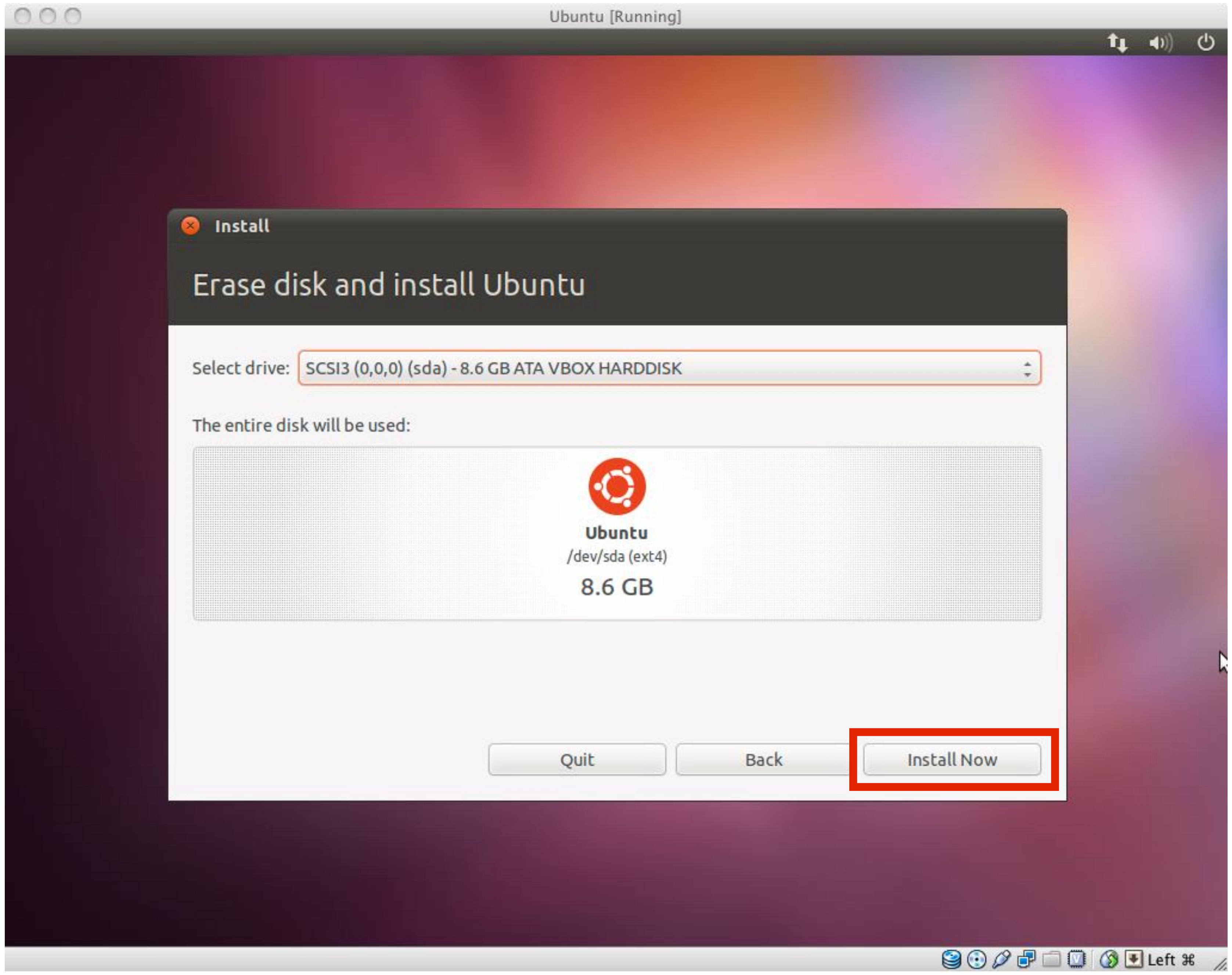
This computer currently has no detected operating systems.
What would you like to do?

[✓ Erase disk and install Ubuntu
Warning: This will delete any files on the disk.]

Something else
You can create or resize partitions yourself, or choose multiple partitions for Ubuntu.

Running Linux

6. Click "Install Now" and wait. Maybe grab a snack.
7. When finished, click Restart and press Enter.



Erase disk and install Ubuntu

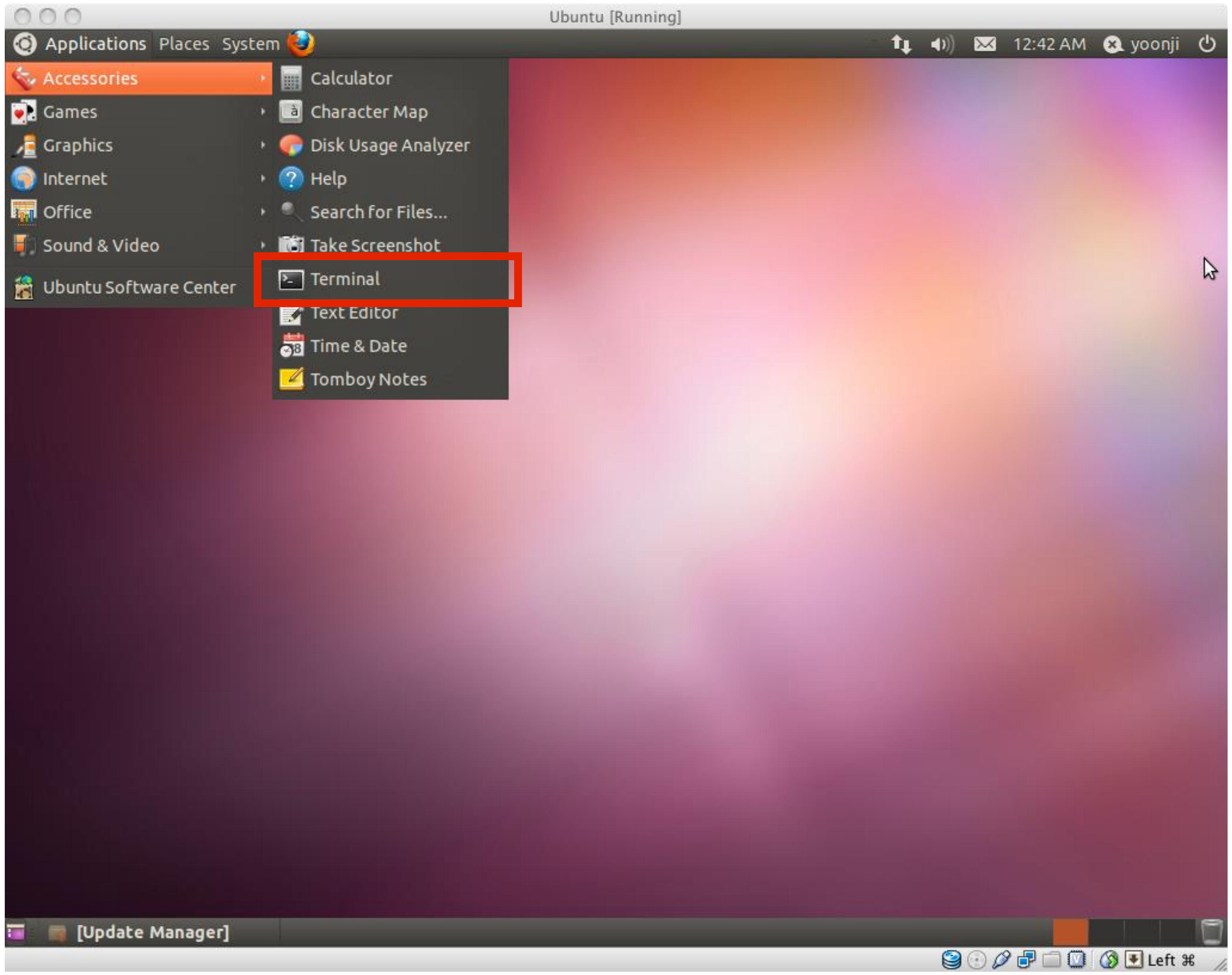
Select drive: SCSI3 (0,0,0) (sda) - 8.6 GB ATA VBOX HARDDISK

The entire disk will be used:

Ubuntu
/dev/sda (ext4)
8.6 GB

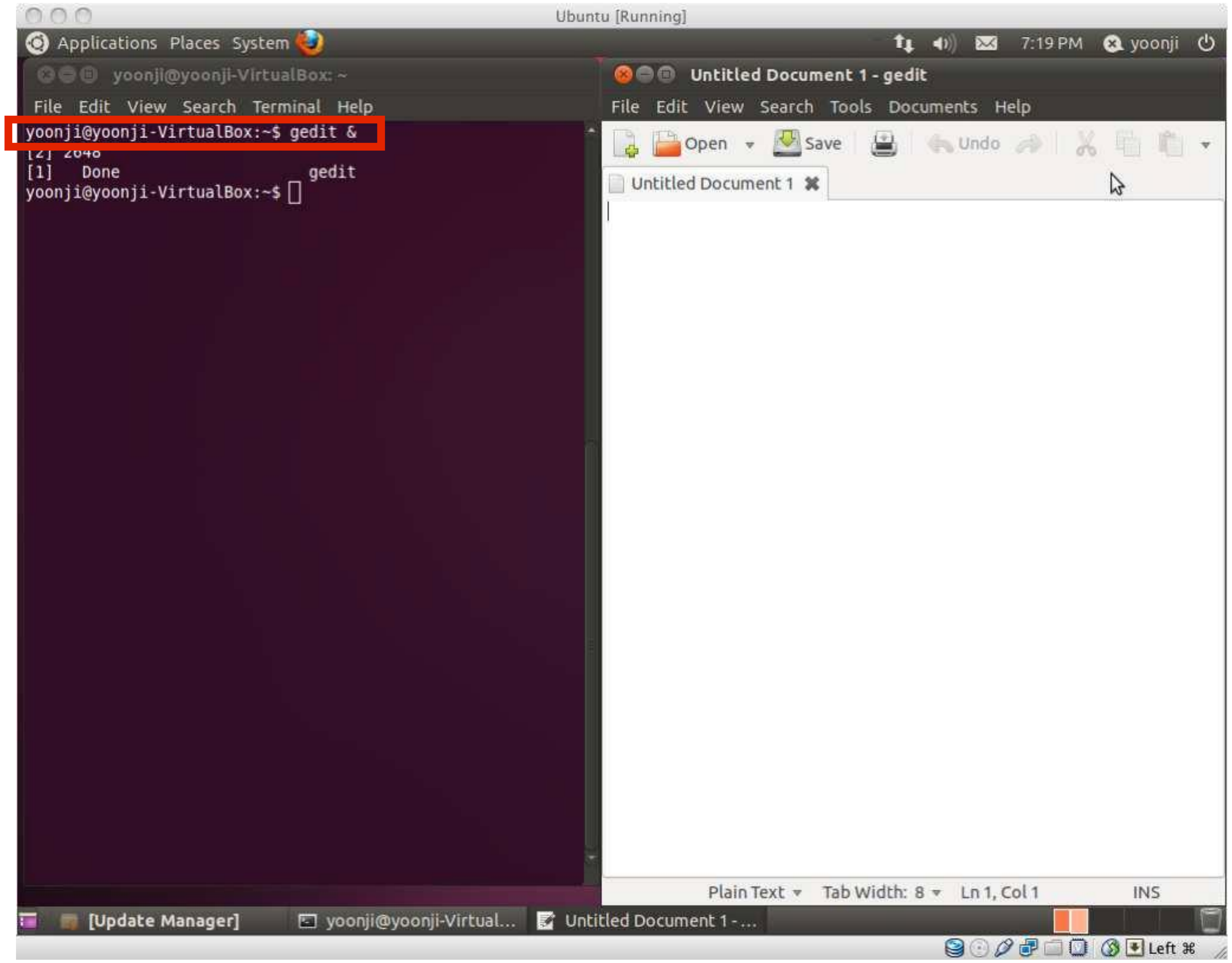
C Programming on Linux

I. Open Terminal (Applications-Accessories-Terminal)



C Programming on Linux

2. Open gedit by typing “gedit &” on terminal
   (You can also use any other Text Editor application)



```
yoonji@yoonji-VirtualBox:~$ gedit &
[2] 2048
[1]   Done                    gedit
yoonji@yoonji-VirtualBox:~$ 
```

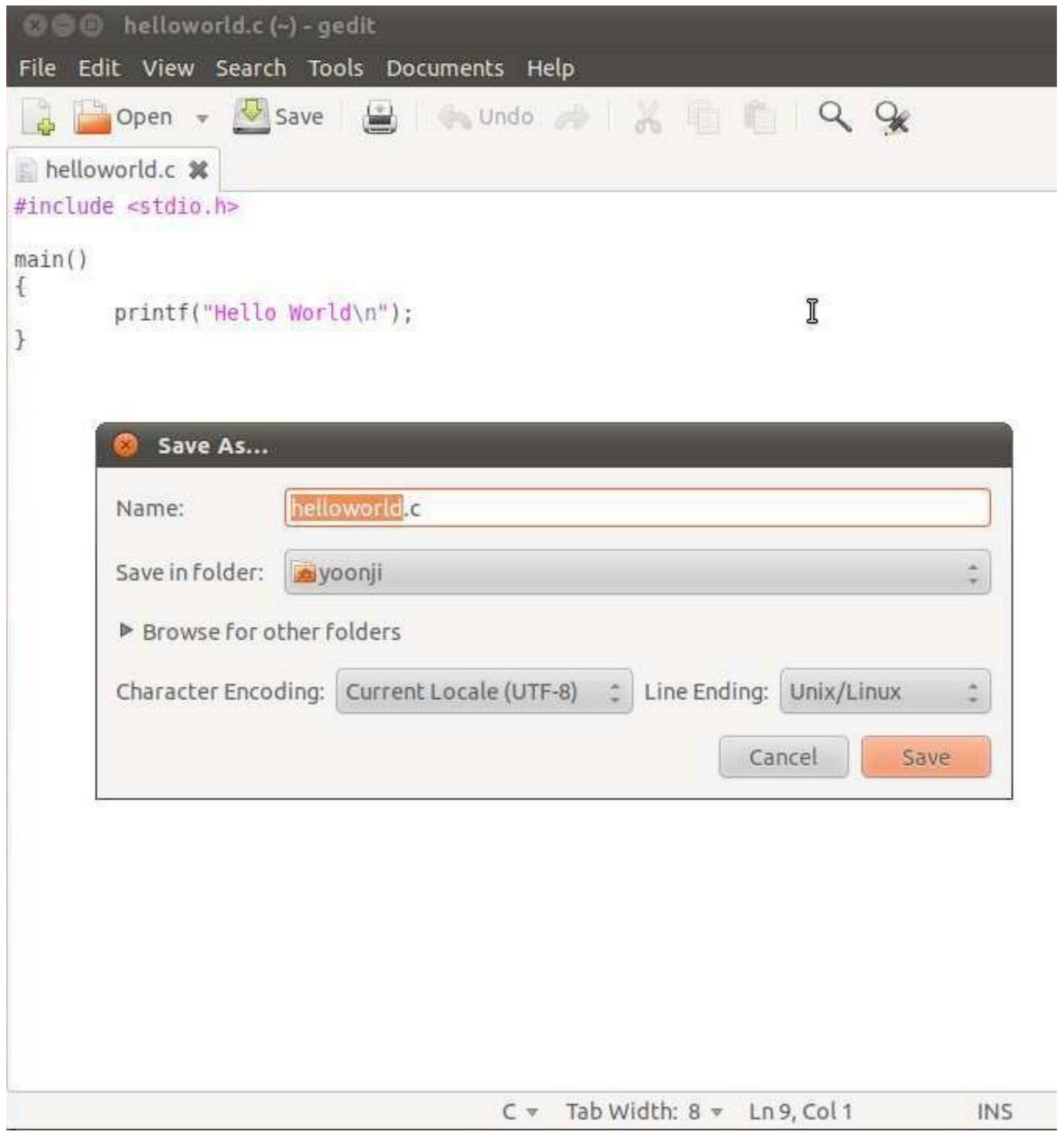
Ubuntu [Running]
Applications Places System
7:19 PM yoonji
yoonji@yoonji-VirtualBox: ~
File Edit View Search Terminal Help
Untitled Document 1 - gedit
File Edit View Search Tools Documents Help
Open Save Undo
Untitled Document 1
Plain Text Tab Width: 8 Ln 1, Col 1 INS
[Update Manager] yoonji@yoonji-Virtual... Untitled Document 1 - ...
Left ⌘

C Programming on Linux

3. Type the following on gedit (or any other text editor)

```
#include<3tdio.h>

main()
{
    printf("Hello World\n");
}
```

4. Save this file as "helloworld.c"



```
#include <stdio.h>

main()
{
        printf("Hello World\n");
}
```

Name: helloworld.c

Save in folder: yoonji

Character Encoding: Current Locale (UTF-8)   Line Ending: Unix/Linux

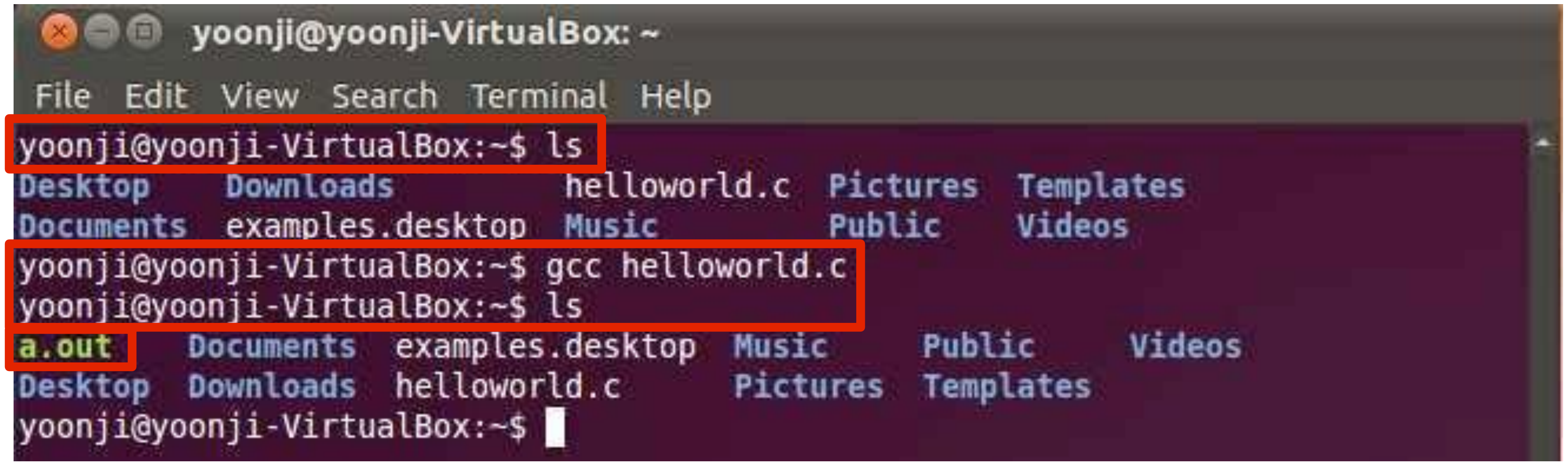
C   Tab Width: 8   Ln 9, Col 1   INS

C Programming on Linux

5. Type “ls” on Terminal to see all files under current folder

6. Confirm that “helloworld.c” is in the current directory. If not, type cd DIRECTORY_PATH to go to the directory that has “helloworld.c”

7. Type “gcc helloworld.c” to compile, and type “ls” to confirm that a new executable file “a.out” is created


yoonji@yoonji-VirtualBox: ~
File Edit View Search Terminal Help

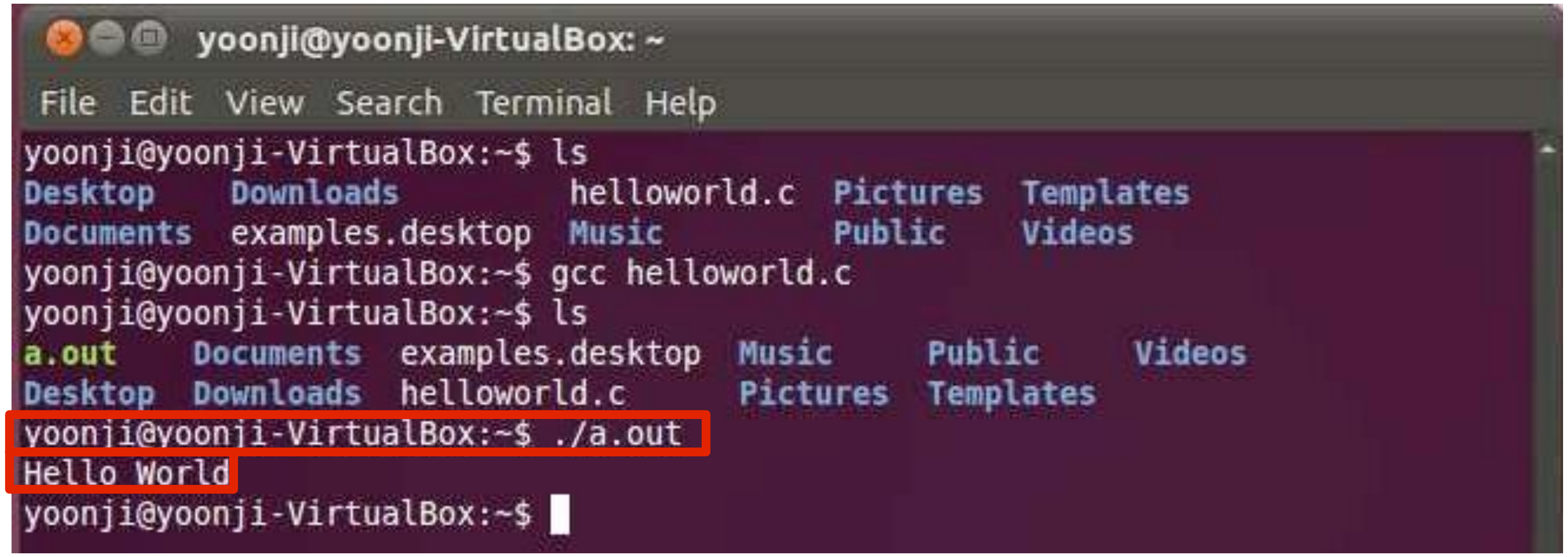
```
yoonji@yoonji-VirtualBox:~$ ls
Desktop    Downloads         helloworld.c  Pictures  Templates
Documents  examples.desktop  Music         Public    Videos
yoonji@yoonji-VirtualBox:~$ gcc helloworld.c
yoonji@yoonji-VirtualBox:~$ ls
a.out    Documents  examples.desktop  Music     Public     Videos
Desktop  Downloads  helloworld.c      Pictures  Templates
yoonji@yoonji-VirtualBox:~$ 
```

C Programming on Linux

8. Type "./a.out" on Terminal to run the program

9. If you see "Hello World" on the next line,
   you just successfully ran your first C program!

10. Try other codes from "A Shotgun Introduction to C" on professor Edwards's webpage. You can also find many C programing guides online. (just google it!)   Enjoy :)


yoonji@yoonji-VirtualBox: ~
File Edit View Search Terminal Help

```
yoonji@yoonji-VirtualBox:~$ ls
Desktop    Downloads         helloworld.c  Pictures  Templates
Documents  examples.desktop  Music         Public    Videos
yoonji@yoonji-VirtualBox:~$ gcc helloworld.c
yoonji@yoonji-VirtualBox:~$ ls
a.out    Documents  examples.desktop  Music     Public     Videos
Desktop  Downloads  helloworld.c      Pictures  Templates
yoonji@yoonji-VirtualBox:~$ ./a.out
Hello World
yoonji@yoonji-VirtualBox:~$ 
```

Ex.No. 3 Installing and Running the Google App Engine On Windows

This document describes the installation of the Google App Engine Software Development Kit (SDK) on a Microsoft Windows and running a simple "hello world" application.

The App Engine SDK allows you to run Google App Engine Applications on your local computer. It simulates the run---time environment of the Google App Engine infrastructure.

Pre--Requisites: Python 2.5.4

If you don't already have Python 2.5.4 installed in your computer, download and Install Python 2.5.4 from:

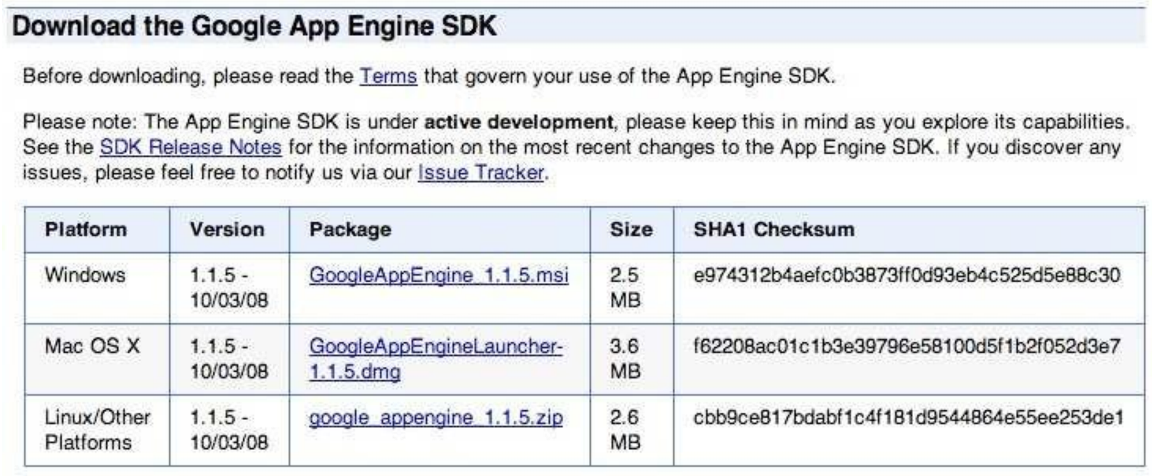
```
http://www.python.org/download/releases/2.5.4/
```

Download and Install

You can download the Google App Engine SDK by going to:

```
http://code.google.com/appengine/downloads.html
```

and download the appropriate install package.



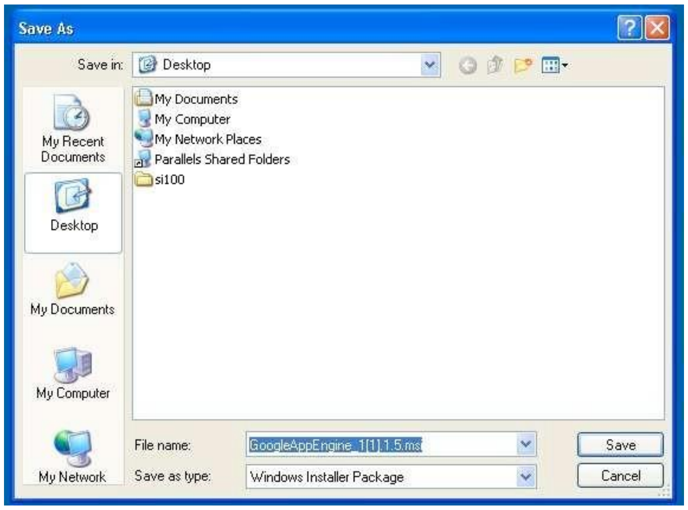
Download the Google App Engine SDK

Before downloading, please read the Terms that govern your use of the App Engine SDK.

Please note: The App Engine SDK is under **active development**, please keep this in mind as you explore its capabilities. See the SDK Release Notes for the information on the most recent changes to the App Engine SDK. If you discover any issues, please feel free to notify us via our Issue Tracker.

Platform	Version	Package	Size	SHA1 Checksum
Windows	1.1.5 - 10/03/08	GoogleAppEngine_1.1.5.msi	2.5 MB	e974312b4aefc0b3873ff0d93eb4c525d5e88c30
Mac OS X	1.1.5 - 10/03/08	GoogleAppEngineLauncher-1.1.5.dmg	3.6 MB	f62208ac01c1b3e39796e58100d5f1b2f052d3e7
Linux/Other Platforms	1.1.5 - 10/03/08	google_appengine_1.1.5.zip	2.6 MB	cbb9ce817bdabf1c4f181d9544864e55ee253de1

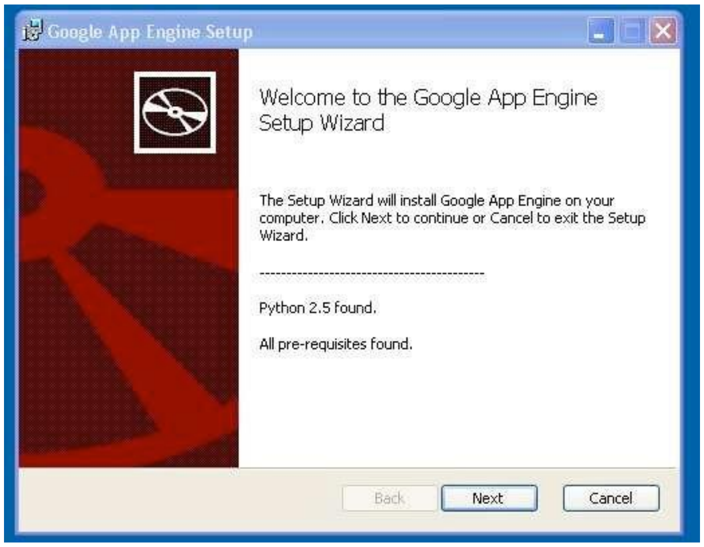
Download the Windows installer – the simplest thing is to download it to your Desktop or another folder that you remember.



File name: GoogleAppEngine_1[1].1.5.msi

Save as type: Windows Installer Package

Double Click on the **GoogleApplicationEngine** installer.



Welcome to the Google App Engine Setup Wizard

The Setup Wizard will install Google App Engine on your computer. Click Next to continue or Cancel to exit the Setup Wizard.

Python 2.5 found.

All pre-requisites found.

Click through the installation wizard, and it should install the App Engine. If you do not have Python 2.5, it will install Python 2.5 as well.

Once the install is complete you can discard the downloaded installer

Making your First Application

Now you need to create a simple application. We could use the "+" option to have the launcher make us an application – but instead we will do it by hand to get a better sense of what is going on.

Make a folder for your Google App Engine applications. I am going to make the Folder on my Desktop called "**apps**" – the path to this folder is:

C:\Documents and Settings\csev\Desktop\apps

And then make a sub-folder in within **apps** called "**ae-01-trivial**" – the path to this folder would be:

C:\ Documents and Settings \csev\Desktop\apps\ae--01--trivial

Using a text editor such as JEdit (www.jedit.org), create a file called **app.yaml** in the **ae--01--trivial** folder with the following contents:

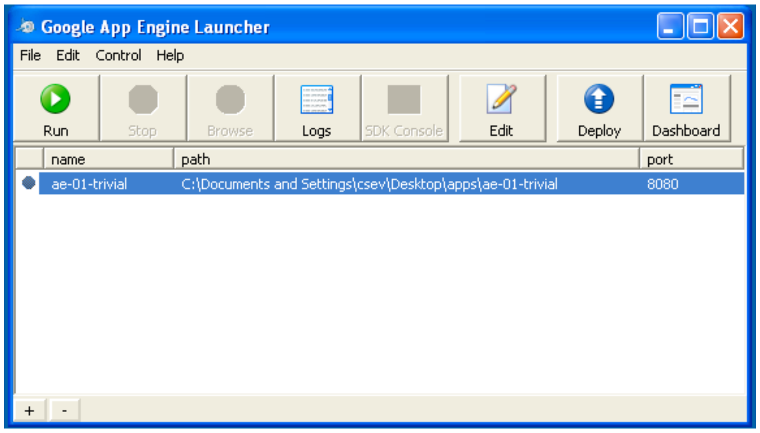
```
application: ae-01-trivial
version: 1
runtime: python
api_version: 1

handlers:
- url: /.*
  script: index.py
```

Note: Please do not copy and paste these lines into your text editor – you might end up with strange characters – simply type them into your editor.

Then create a file in the **ae--01--trivial** folder called **index.py** with three lines in it:

```
print 'Content-Type: text/plain'
print ' '
print 'Hello there Chuck'
```

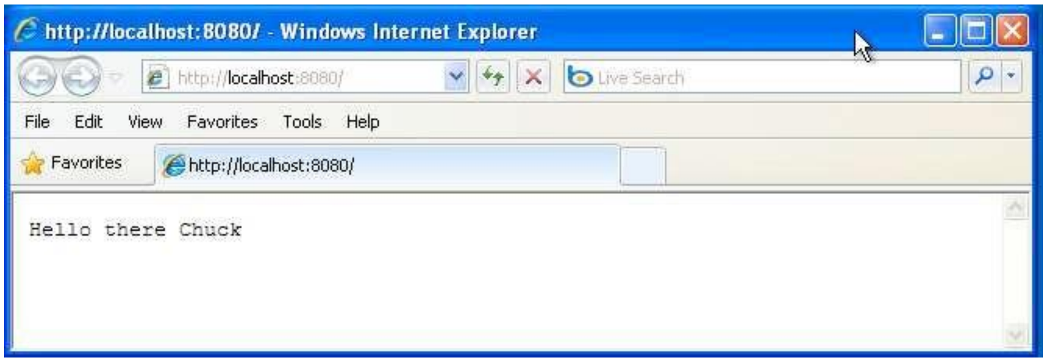
Then start the **GoogleAppEngineLauncher** program that can be found under **Applications**. Use the **File --> Add Existing Application** command and navigate into the **apps** directory and select the **ae--01--trivial** folder. Once you have added the application, select it so that you can control the application using the launcher.



name	path	port
ae-01-trivial	C:\Documents and Settings\csev\Desktop\apps\ae-01-trivial	8080

Once you have selected your application and press **Run**. After a few moments your application will start and the launcher will show a little green icon next to your application. Then press **Browse** to open a browser pointing at your application which is running at **http://localhost:8080/**

Paste **http://localhost:8080** into your browser and you should see your application as follows:

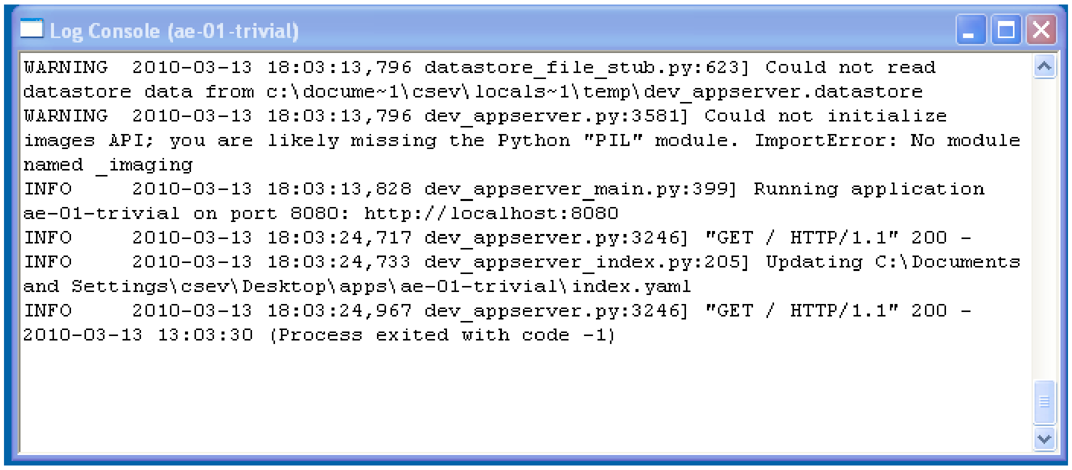

http://localhost:8080/ - Windows Internet Explorer
http://localhost:8080/
Live Search
File Edit View Favorites Tools Help
Favorites
http://localhost:8080/

```
Hello there Chuck
```

Just for fun, edit the **index.py** to change the name “Chuck” to your own name and press Refresh in the browser to verify your updates.

Watching the Log

You can watch the internal log of the actions that the web server is performing when you are interacting with your application in the browser. Select your application in the Launcher and press the **Logs** button to bring up a log window:

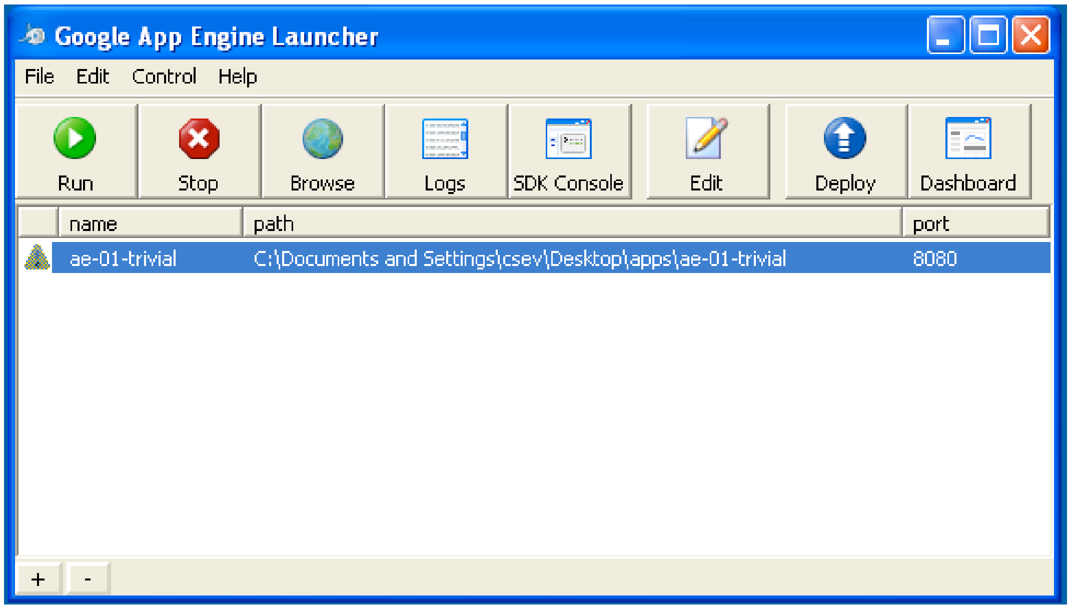


```
WARNING  2010-03-13 18:03:13,796 datastore_file_stub.py:623] Could not read
datastore data from c:\docume~1\csev\locals~1\temp\dev_appserver.datastore
WARNING  2010-03-13 18:03:13,796 dev_appserver.py:3581] Could not initialize
images API; you are likely missing the Python "PIL" module. ImportError: No module
named _imaging
INFO     2010-03-13 18:03:13,828 dev_appserver_main.py:399] Running application
ae-01-trivial on port 8080: http://localhost:8080
INFO     2010-03-13 18:03:24,717 dev_appserver.py:3246] "GET / HTTP/1.1" 200 -
INFO     2010-03-13 18:03:24,733 dev_appserver_index.py:205] Updating C:\Documents
and Settings\csev\Desktop\apps\ae-01-trivial\index.yaml
INFO     2010-03-13 18:03:24,967 dev_appserver.py:3246] "GET / HTTP/1.1" 200 -
2010-03-13 13:03:30 (Process exited with code -1)
```

Each time you press **Refresh** in your browser – you can see it retrieving the output with a **GET** request.

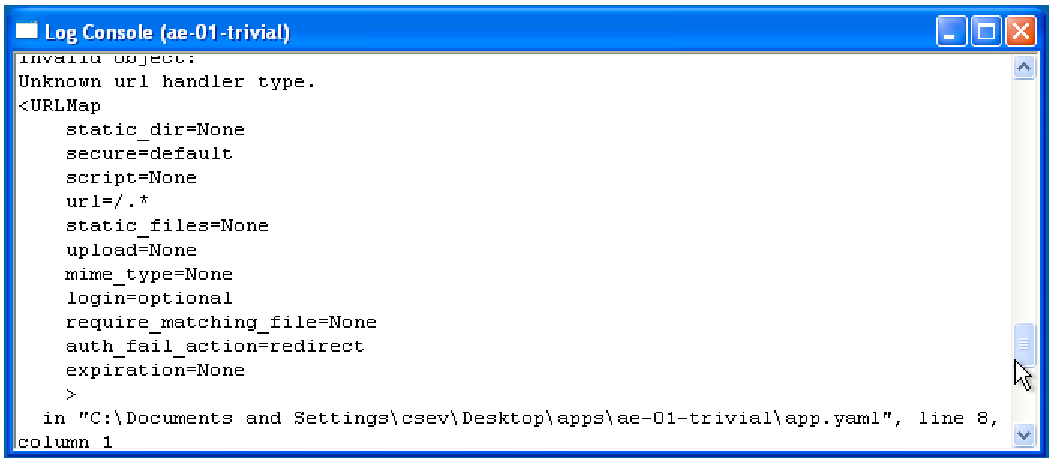
Dealing With Errors

With two files to edit, there are two general categories of errors that you may encounter. If you make a mistake on the **app.yaml** file, the App Engine will not start and your launcher will show a yellow icon near your application:



name	path	port
ae-01-trivial	C:\Documents and Settings\csev\Desktop\apps\ae-01-trivial	8080

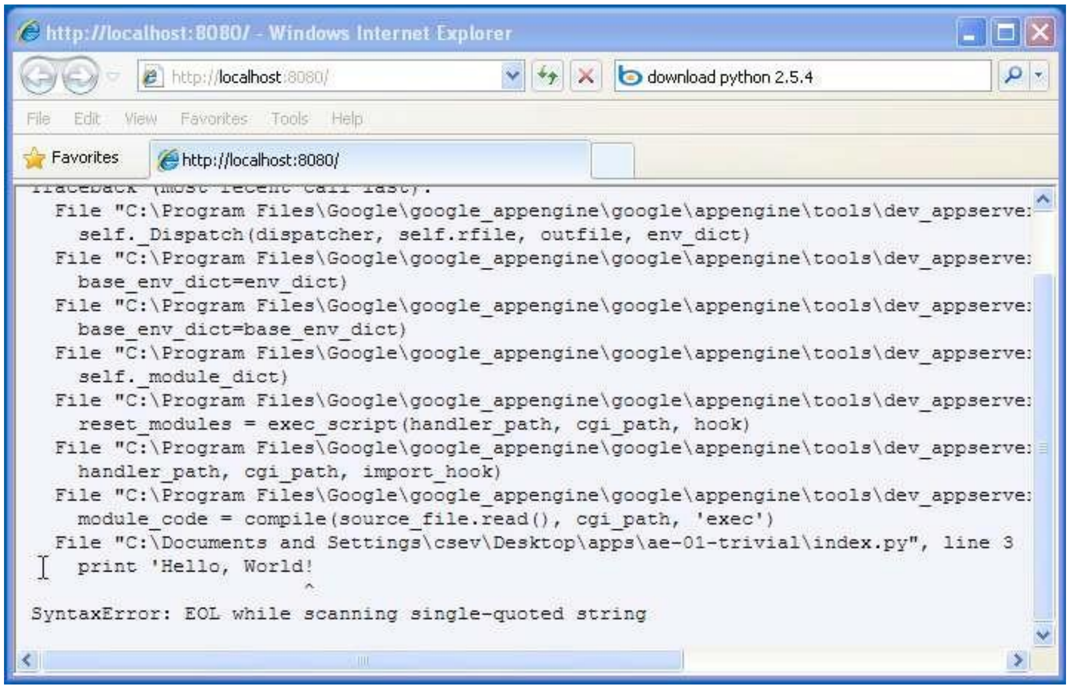
To get more detail on what is going wrong, take a look at the log for the application:



```
invalid object:
Unknown url handler type.
<URLMap
    static_dir=None
    secure=default
    script=None
    url=/.*
    static_files=None
    upload=None
    mime_type=None
    login=optional
    require_matching_file=None
    auth_fail_action=redirect
    expiration=None
    >
  in "C:\Documents and Settings\csev\Desktop\apps\ae-01-trivial\app.yaml", line 8,
column 1
```

In this instance – the mistake is mis-indenting the last line in the **app.yaml** (line 8).

If you make a syntax error in the **index.py** file, a Python trace back error will appear in your browser.



```
Traceback (most recent call last):
  File "C:\Program Files\Google\google_appengine\google\appengine\tools\dev_appserve:
    self._Dispatch(dispatcher, self.rfile, outfile, env_dict)
  File "C:\Program Files\Google\google_appengine\google\appengine\tools\dev_appserve:
    base_env_dict=env_dict)
  File "C:\Program Files\Google\google_appengine\google\appengine\tools\dev_appserve:
    base_env_dict=base_env_dict)
  File "C:\Program Files\Google\google_appengine\google\appengine\tools\dev_appserve:
    self._module_dict)
  File "C:\Program Files\Google\google_appengine\google\appengine\tools\dev_appserve:
    reset_modules = exec_script(handler_path, cgi_path, hook)
  File "C:\Program Files\Google\google_appengine\google\appengine\tools\dev_appserve:
    handler_path, cgi_path, import_hook)
  File "C:\Program Files\Google\google_appengine\google\appengine\tools\dev_appserve:
    module_code = compile(source_file.read(), cgi_path, 'exec')
  File "C:\Documents and Settings\csev\Desktop\apps\ae-01-trivial\index.py", line 3
    print 'Hello, World!
                       ^

SyntaxError: EOL while scanning single-quoted string
```

The error you need to see is likely to be the last few lines of the output – in this case I made a Python syntax error on line one of our one-line application.

Reference: http://en.wikipedia.org/wiki/Stack_trace

When you make a mistake in the **app.yaml** file – you must the fix the mistake and attempt to start the application again.

If you make a mistake in a file like i**ndex.py**, you can simply fix the file and press refresh in your browser – there is no need to restart the server.

Shutting Down the Server

To shut down the server, use the Launcher, select your application and press the **Stop** button.

Ex.No. 5 How to use CloudSim in Eclipse

CloudSim is written in Java. The knowledge you need to use CloudSim is basic Java programming and some basics about cloud computing. Knowledge of programming IDEs such as Eclipse or NetBeans is also helpful. It is a library and, hence, CloudSim does not have to be installed. Normally, you can unpack the downloaded package in any directory, add it to the Java classpath and it is ready to be used. Please verify whether Java is available on your system.
To use CloudSim in Eclipse:
1. Download CloudSim installable files
from *https://code.google.com/p/cloudsim/downloads/list and unzip*
2. Open Eclipse
3. Create a new Java Project: File -> New
4. Import an unpacked CloudSim project into the new Java Project
5. The first step is to initialise the CloudSim package by initialising the CloudSim library, as follows:

```
CloudSim.init(num_user, calendar, trace_flag)
```

6. Data centres are the resource providers in CloudSim; hence, creation of data centres is a second step. To create Datacenter, you need the DatacenterCharacteristics object that stores the properties of a data centre such as architecture, OS, list of machines, allocation policy that covers the time or spaceshared, the time zone and its price:

```
Datacenter datacenter9883 = new Datacenter(name, characteristics, new VmAllocationPolicySimple(hostList), s
```

7. The third step is to create a broker:

```
DatacenterBroker broker = createBroker();
```

8. The fourth step is to create one virtual machine unique ID of the VM, userId ID of the VM's owner, mips, number Of Pes amount of CPUs, amount of RAM, amount of bandwidth, amount of storage, virtual machine monitor, and cloudletScheduler policy for cloudlets:

```
Vm vm = new Vm(vmid, brokerId, mips, pesNumber, ram, bw, size, vmm, new
CloudletSchedulerTimeShared())
```

9. Submit the VM list to the broker:

```
broker.submitVmList(vmlist)
```

10. Create a cloudlet with length, file size, output size, and utilisation model:

```
Cloudlet cloudlet = new Cloudlet(id, length, pesNumber, fileSize, outputSize, utilizationModel, utilizationMode
```

11. Submit the cloudlet list to the broker:

```
broker.submitCloudletList(cloudletList)
```

12. Start the simulation:

```
CloudSim.startSimulation()
```

Sample Output from the Existing Example:

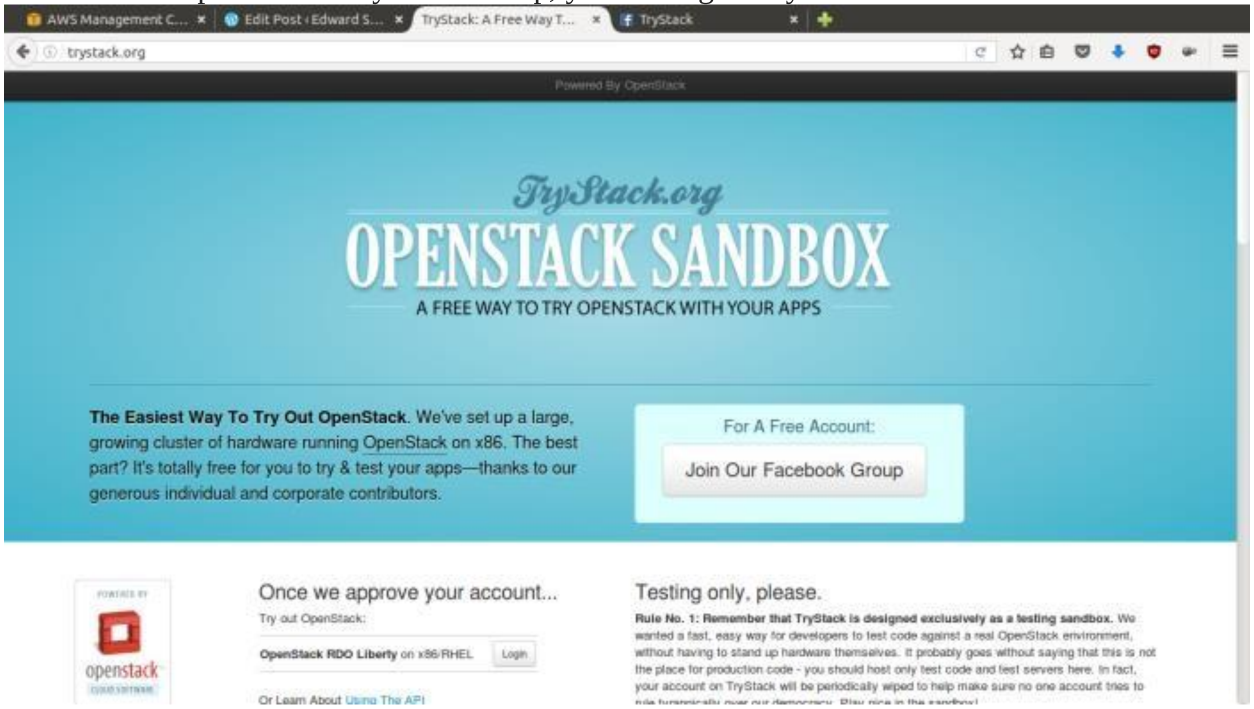
```
Starting CloudSimExample1...
Initialising...
Starting CloudSim version 3.0
Datacenter_0 is starting...
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>null
Broker is starting...
Entities started.
: Broker: Cloud Resource List received with 1 resource(s)
0.0: Broker: Trying to Create VM #0 in Datacenter_0
    : Broker: VM #0 has been created in Datacenter #2, Host #0
0.1: Broker: Sending cloudlet 0 to VM #0
400.1: Broker: Cloudlet 0 received
: Broker: All Cloudlets executed. Finishing...
400.1: Broker: Destroying VM #0
Broker is shutting down...
Simulation: No more future events
CloudInformationService: Notify all CloudSim entities for shutting down.
Datacenter_0 is shutting down...
Broker is shutting down...
Simulation completed.
Simulation completed.
========== OUTPUT ==========
Cloudlet ID  STATUS   Data center ID   VM ID   Time   Start Time   Finish Time
     0         SUCCESS     2            0      400                  0.1            400.1
*****Datacenter: Datacenter_0*****
User id      Debt
    3          35.6

CloudSim Example1 finished!
```

1. You can copy few (or more) lines with ***copy & paste*** mechanism.
   For this you need to share clipboard between host OS and guest OS, installing **Guest Addition** on both the virtual machines (probably setting *bidirectional* and restarting them).
   You *copy* from *guest OS* in the clipboard that is shared with the *host OS*.
   Then you *paste* from the *host OS* to the second *guest OS*.
2. You can enable **drag and drop** too with the same method (Click on the machine, settings, general, advanced, drag and drop: set to *bidirectional*)
3. You can have **common *Shared Folders*** on both virtual machines and use one of the directory shared as buffer to copy.
   Installing **Guest Additions** you have the possibility to set Shared Folders too. As you put a file in a shared folder from *host OS* or from *guest OS*, is immediately visible to the other. (Keep in mind that can arise some problems for date/time of the files when there are different clock settings on the different virtual machines).
   If you use the same folder shared on more machines you can exchange files directly copying them in this folder.
4. You can use **usual method to copy files between 2 different computer** with client-server application. (e.g. scp with sshd active for linux, winscp... you can get some info about SSH servers e.g. here)
   You need an active server (sshd) on the receiving machine and a client on the sending machine. Of course you need to have the authorization setted (via password or, better, via an automatic authentication method).
   Note: many Linux/Ubuntu distribution install sshd by default: you can see if it is running with pgrep sshd from a shell. You can install with sudo apt-get install openssh-server.
5. You can **mount part of the file system** of a virtual machine via NFS or SSHFS on the other, or you can **share file and directory** with Samba.
   You may find interesting the article Sharing files between guest and host without VirtualBox shared folders with detailed step by step instructions.

You should remember that you are dialling with a little network of machines with different operative systems, and in particular:

- Each virtual machine has its own operative system running on and acts as a physical machine.
- Each virtual machine is an instance of a program *owned* by an *user* in the hosting operative system and should undergo the restrictions of the *user* in the *hosting OS*.
  E.g Let we say that Hastur and Meow are users of the hosting machine, but they did not allow each other to see their directories (no read/write/execute authorization). When each of them run a virtual machine, for the hosting OS those virtual machine are two normal programs owned by Hastur and Meow and cannot see the private directory of the other user. This is a restriction due to the *hosting OS*. It's easy to overcame it: it's enough to give authorization to read/write/execute to a directory or to chose a different directory in which both users can read/write/execute.
- Windows likes mouse and Linux fingers. :-)
  I mean I suggest you to enable *Drag & drop* to be cosy with the Windows machines and the *Shared folders* or to be cosy with Linux.

OpenStack is an open-source software cloud computing platform. OpenStack is primarily used for deploying an infrastructure as a service (IaaS) solution like Amazon Web Service (AWS). In other words, you can *make your own AWS* by using OpenStack. If you want to try out OpenStack, **TryStack** is the easiest and free way to do it.
In order to try OpenStack in TryStack, you must register yourself by joining TryStack Facebook Group. The acceptance of group needs a couple days because it's approved manually. After you have been accepted in the TryStack Group, you can log in TryStack.

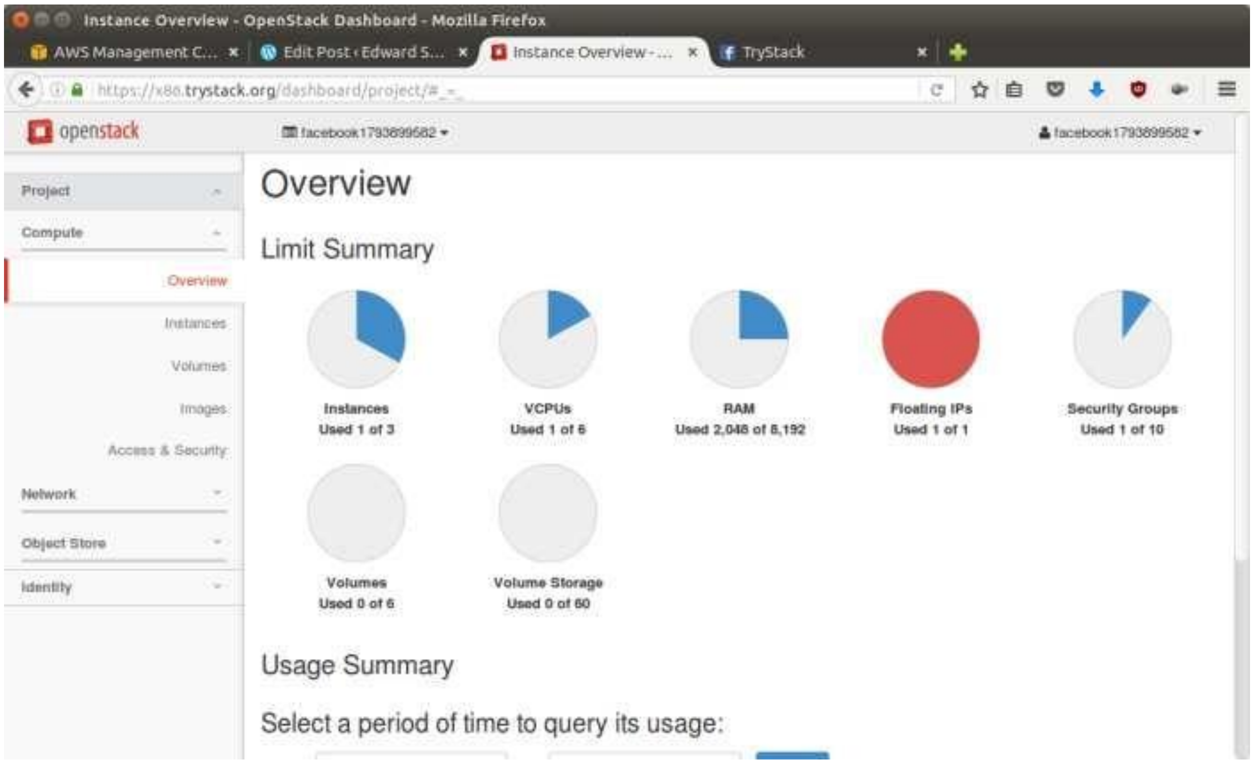


The Easiest Way To Try Out OpenStack. We've set up a large, growing cluster of hardware running OpenStack on x86. The best part? It's totally free for you to try & test your apps—thanks to our generous individual and corporate contributors.

Once we approve your account...
Try out OpenStack:
OpenStack RDO Liberty on x86/RHEL Login
Or Learn About Using The API

Testing only, please.
Rule No. 1: Remember that TryStack is designed exclusively as a testing sandbox. We wanted a fast, easy way for developers to test code against a real OpenStack environment, without having to stand up hardware themselves. It probably goes without saying that this is not the place for production code - you should host only test code and test servers here. In fact, your account on TryStack will be periodically wiped to help make sure no one account tries to rule tyrannically over our democracy. Play nice in the sandbox!

TryStack.org Homepage
I assume that you already join to the Facebook Group and login to the dashboard. After you log in to the TryStack, you will see the Compute Dashboard like:



Overview

Limit Summary

Instances
Used 1 of 3

VCPUs
Used 1 of 6

RAM
Used 2,048 of 8,192

Floating IPs
Used 1 of 1

Security Groups
Used 1 of 10

Volumes
Used 0 of 6

Volume Storage
Used 0 of 60

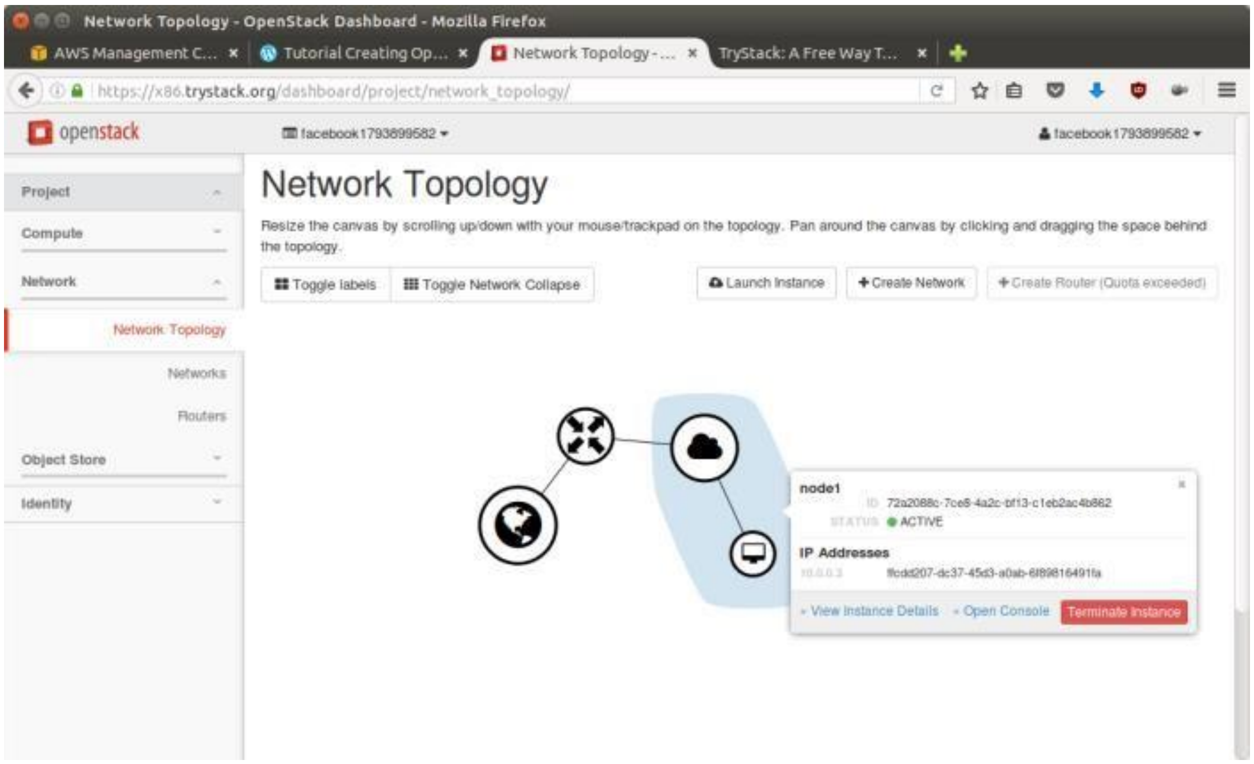
Usage Summary

Select a period of time to query its usage:

OpenStack Compute Dashboard

Overview: What we will do?

In this post, I will show you how to run an OpenStack instance. The instance will be accessible through the internet (have a public IP address). The final topology will like:


Network Topology - OpenStack Dashboard - Mozilla Firefox
AWS Management C...
Tutorial Creating Op...
Network Topology - ...
TryStack: A Free Way T...
https://x86.trystack.org/dashboard/project/network_topology/
openstack
facebook1793899582
Project
Compute
Network
Network Topology
Networks
Routers
Object Store
Identity
Network Topology
Resize the canvas by scrolling up/down with your mouse/trackpad on the topology. Pan around the canvas by clicking and dragging the space behind the topology.
Toggle labels
Toggle Network Collapse
Launch Instance
Create Network
Create Router (Quota exceeded)
node1
ID 72a2088c-7ce8-4a2c-bf13-c1eb2ac4b862
STATUS ACTIVE
IP Addresses
10.0.0.3 ffcdd207-dc37-45d3-a0ab-6f898164916a
View Instance Details
Open Console
Terminate Instance
Network topology
As you see from the image above, the instance will be connected to a local network and the local network will be connected to internet.

Step 1: Create Network

Network? Yes, the network in here is our own local network. So, your instances will be not mixed up with the others. You can imagine this as your own LAN (Local Area Network) in the cloud.

1. Go to **Network > Networks** and then click **Create Network**.
2. In **Network** tab, fill **Network Name** for example internal and then click **Next**.
3. In **Subnet** tab,
    1. Fill **Network Address** with appropriate CIDR, for example 192.168.1.0/24. Use private network CIDR block as the best practice.
    2. Select **IP Version** with appropriate IP version, in this case IPv4.
    3. Click **Next**.
4. In **Subnet Details** tab, fill **DNS Name Servers** with 8.8.8.8 (Google DNS) and then click **Create**.

Step 2: Create Instance

Now, we will create an instance. The instance is a virtual machine in the cloud, like AWS EC2. You need the instance to connect to the network that we just created in the previous step.

1. Go to **Compute > Instances** and then click **Launch Instance**.
2. In **Details** tab,
   1. Fill **Instance Name**, for example Ubuntu 1.
   2. Select **Flavor**, for example m1.medium.
   3. Fill **Instance Count** with **1**.
   4. Select **Instance Boot Source** with **Boot from Image**.
   5. Select **Image Name** with **Ubuntu 14.04 amd64 (243.7 MB)** if you want install Ubuntu 14.04 in your virtual machine.
3. In **Access & Security** tab,
   1. Click [+] button of **Key Pair** to import key pair. This key pair is a public and private key that we will use to connect to the instance from our machine.
   2. In **Import Key Pair** dialog,
      1. Fill **Key Pair Name** with your machine name (for example Edward-Key).
      2. Fill **Public Key** with your **SSH public key** (usually is in ~/.ssh/id_rsa.pub). See description in Import Key Pair dialog box for more information. If you are using Windows, you can use **Puttygen** to generate key pair.
      3. Click **Import key pair**.
   3. In **Security Groups**, mark/check **default**.
4. In **Networking** tab,
   1. In **Selected Networks**, select network that have been created in Step 1, for example internal.
5. Click **Launch**.
6. If you want to create multiple instances, you can repeat step 1-5. I created one more instance with instance name Ubuntu 2.

Step 3: Create Router

I guess you already know what router is. In the step 1, we created our network, but it is isolated. It doesn't connect to the internet. To make our network has an internet connection, we need a router that running as the gateway to the internet.

1. Go to **Network > Routers** and then click **Create Router**.
2. Fill **Router Name** for example router1 and then click **Create router**.
3. Click on your **router name link**, for example router1, **Router Details** page.
4. Click **Set Gateway** button in upper right:
   1. Select **External networks** with **external**.
   2. Then **OK**.
5. Click **Add Interface** button.
   1. Select **Subnet** with the network that you have been created in Step 1.
   2. Click **Add interface**.
6. Go to **Network > Network Topology**. You will see the network topology. In the example, there are two network, i.e. external and internal, those are bridged by a router. There are instances those are joined to internal network.

Step 4: Configure Floating IP Address

Floating IP address is public IP address. It makes your instance is accessible from the internet. When you launch your instance, the instance will have a private network IP, but no public IP. In OpenStack, the public IPs is collected in a pool and managed by admin (in our case is TryStack). You need to request a public (floating) IP address to be assigned to your instance.

1. Go to **Compute > Instance**.
2. In one of your instances, click **More > Associate Floating IP**.
3. In **IP Address**, click Plus [+].
4. Select **Pool** to **external** and then click **Allocate IP**.
5. Click **Associate**.
6. Now you will get a public IP, e.g. 8.21.28.120, for your instance.

Step 5: Configure Access & Security

OpenStack has a feature like a firewall. It can whitelist/blacklist your in/out connection. It is called *Security Group*.

1. Go to **Compute > Access & Security** and then open **Security Groups** tab.
2. In **default** row, click **Manage Rules**.
3. Click **Add Rule**, choose **ALL ICMP** rule to enable ping into your instance, and then click **Add**.
4. Click **Add Rule**, choose **HTTP** rule to open HTTP port (port 80), and then click **Add**.
5. Click **Add Rule**, choose **SSH** rule to open SSH port (port 22), and then click **Add**.
6. You can open other ports by creating new rules.

Step 6: SSH to Your Instance

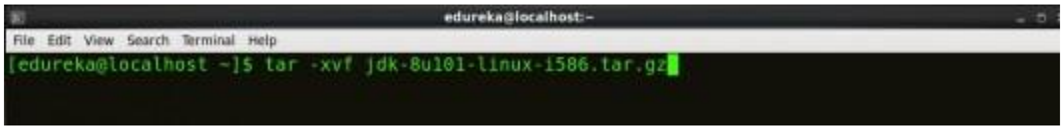
Now, you can SSH your instances to the floating IP address that you got in the step 4. If you are using Ubuntu image, the SSH user will be ubuntu.

Install Hadoop

Step 1: Click here to download the Java 8 Package. Save this file in your home directory.

Step 2: Extract the Java Tar File.

***Command*:** tar -xvf jdk-8u101-linux-i586.tar.gz

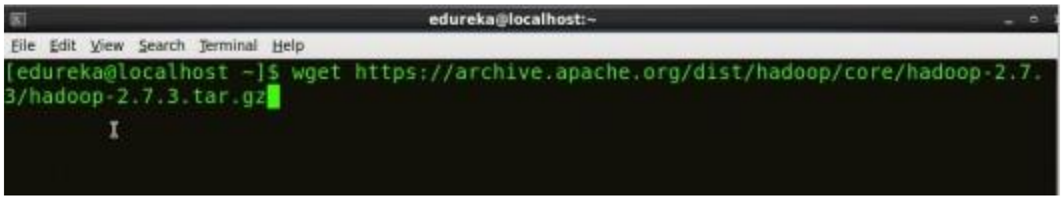


```
[edureka@localhost ~]$ tar -xvf jdk-8u101-linux-i586.tar.gz
```

Fig: Hadoop Installation – Extracting Java Files

Step 3: Download the Hadoop 2.7.3 Package.

***Command*:** wget https://archive.apache.org/dist/hadoop/core/hadoop-2.7.3/hadoop-2.7.3.tar.gz

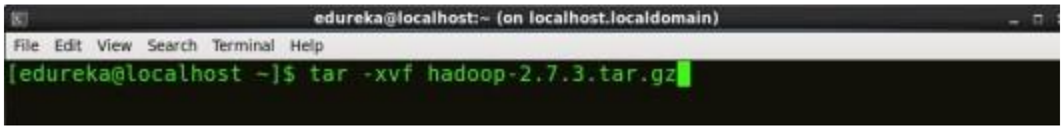


```
[edureka@localhost ~]$ wget https://archive.apache.org/dist/hadoop/core/hadoop-2.7.
3/hadoop-2.7.3.tar.gz
```

Fig: Hadoop Installation – Downloading Hadoop

Step 4: Extract the Hadoop tar File.

Command: tar -xvf hadoop-2.7.3.tar.gz



edureka@localhost:~ (on localhost.localdomain)
File Edit View Search Terminal Help

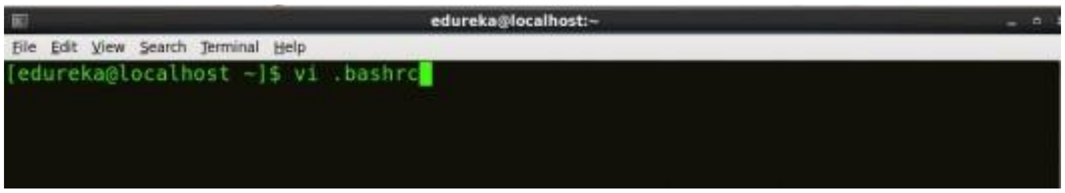
```
[edureka@localhost ~]$ tar -xvf hadoop-2.7.3.tar.gz
```

Fig: Hadoop Installation – Extracting Hadoop Files

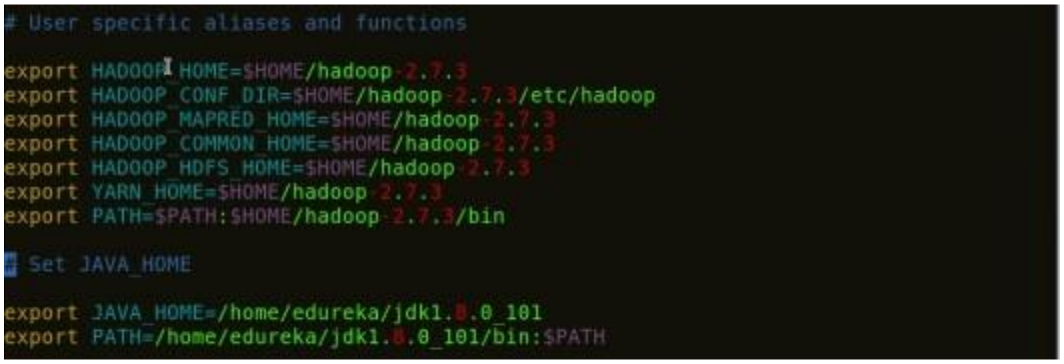
Step 5: Add the Hadoop and Java paths in the bash file (.bashrc).

Open**. bashrc** file. Now, add Hadoop and Java Path as shown below.

***Command*:** vi .bashrc



```
[edureka@localhost ~]$ vi .bashrc
```



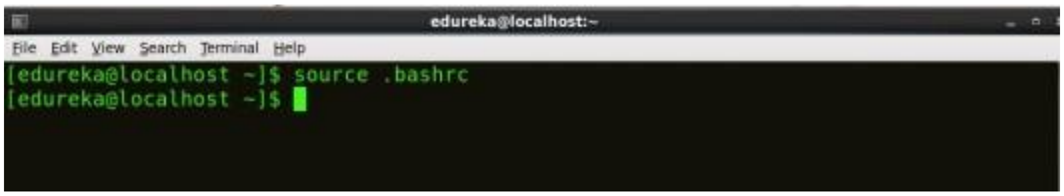
```
# User specific aliases and functions

export HADOOP_HOME=$HOME/hadoop-2.7.3
export HADOOP_CONF_DIR=$HOME/hadoop-2.7.3/etc/hadoop
export HADOOP_MAPRED_HOME=$HOME/hadoop-2.7.3
export HADOOP_COMMON_HOME=$HOME/hadoop-2.7.3
export HADOOP_HDFS_HOME=$HOME/hadoop-2.7.3
export YARN_HOME=$HOME/hadoop-2.7.3
export PATH=$PATH:$HOME/hadoop-2.7.3/bin

# Set JAVA_HOME

export JAVA_HOME=/home/edureka/jdk1.8.0_101
export PATH=/home/edureka/jdk1.8.0_101/bin:$PATH
```

Fig: Hadoop Installation – Setting Environment Variable

Then, save the bash file and close it.

For applying all these changes to the current Terminal, execute the source command.

Command**:** source .bashrc

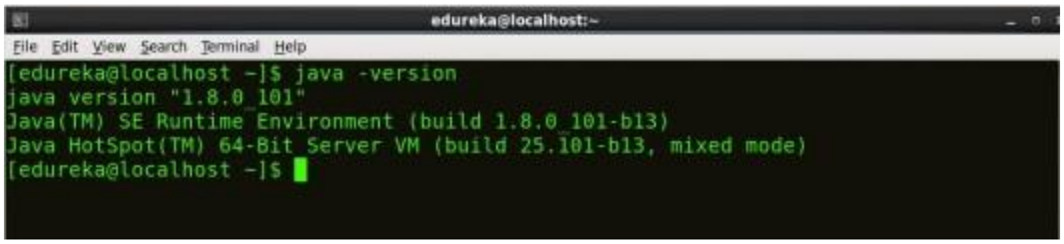


```
[edureka@localhost ~]$ source .bashrc
[edureka@localhost ~]$
```

Fig: Hadoop Installation – Refreshing environment variables

To make sure that Java and Hadoop have been properly installed on your system and can be accessed through the Terminal, execute the java -version and hadoop version commands.

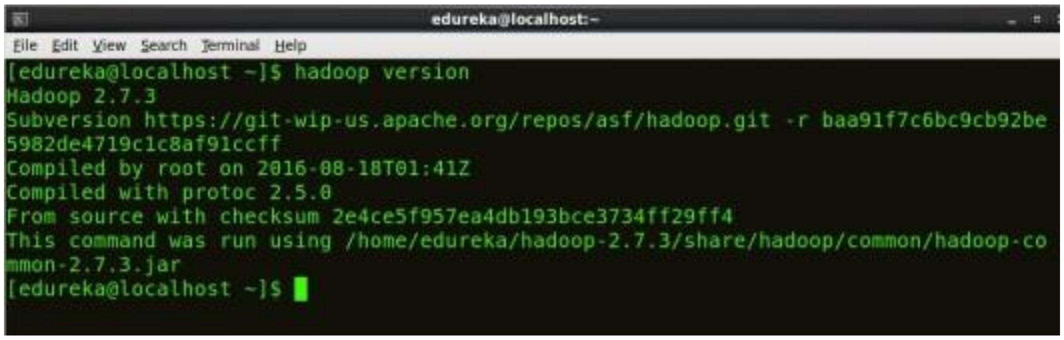
Command**:** java -version



```
[edureka@localhost ~]$ java -version
java version "1.8.0_101"
Java(TM) SE Runtime Environment (build 1.8.0_101-b13)
Java HotSpot(TM) 64-Bit Server VM (build 25.101-b13, mixed mode)
[edureka@localhost ~]$
```

Fig: Hadoop Installation – Checking Java Version

Command**:** hadoop version



```
[edureka@localhost ~]$ hadoop version
Hadoop 2.7.3
Subversion https://git-wip-us.apache.org/repos/asf/hadoop.git -r baa91f7c6bc9cb92be
5982de4719c1c8af91ccff
Compiled by root on 2016-08-18T01:41Z
Compiled with protoc 2.5.0
From source with checksum 2e4ce5f957ea4db193bce3734ff29ff4
This command was run using /home/edureka/hadoop-2.7.3/share/hadoop/common/hadoop-co
mmon-2.7.3.jar
[edureka@localhost ~]$ 
```

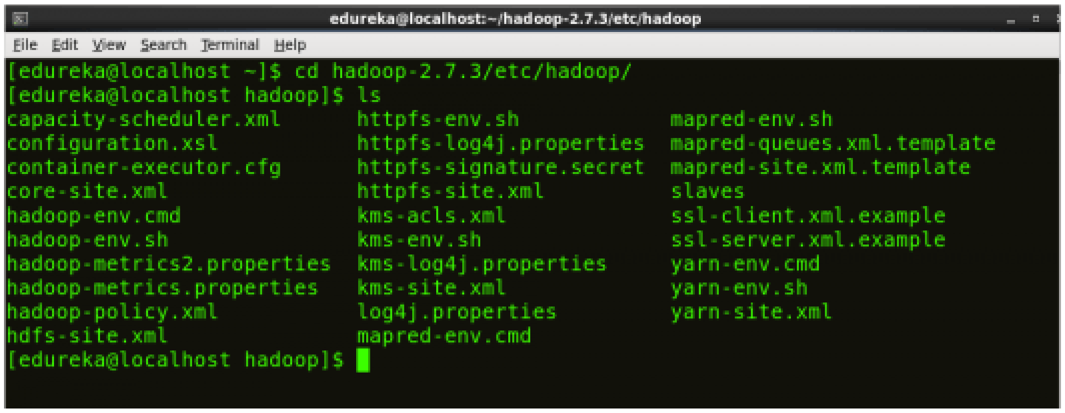
Fig: Hadoop Installation – Checking Hadoop Version

Step 6: Edit the **Hadoop Configuration files**.

Command: cd hadoop-2.7.3/etc/hadoop/

Command: ls

All the Hadoop configuration files are located in **hadoop-2.7.3/etc/hadoop** directory as you can see in the snapshot below:



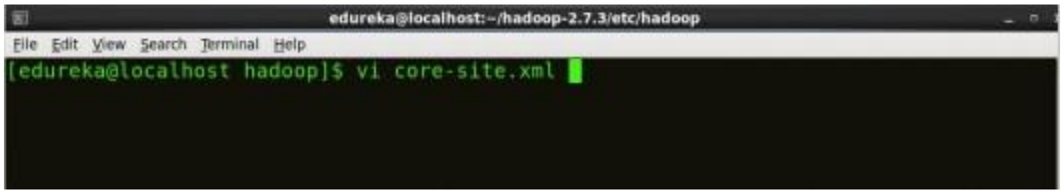
```
edureka@localhost:~/hadoop-2.7.3/etc/hadoop
File Edit View Search Terminal Help
[edureka@localhost ~]$ cd hadoop-2.7.3/etc/hadoop/
[edureka@localhost hadoop]$ ls
capacity-scheduler.xml      httpfs-env.sh              mapred-env.sh
configuration.xsl           httpfs-log4j.properties    mapred-queues.xml.template
container-executor.cfg      httpfs-signature.secret    mapred-site.xml.template
core-site.xml               httpfs-site.xml            slaves
hadoop-env.cmd              kms-acls.xml               ssl-client.xml.example
hadoop-env.sh               kms-env.sh                 ssl-server.xml.example
hadoop-metrics2.properties  kms-log4j.properties       yarn-env.cmd
hadoop-metrics.properties   kms-site.xml               yarn-env.sh
hadoop-policy.xml           log4j.properties           yarn-site.xml
hdfs-site.xml               mapred-env.cmd
[edureka@localhost hadoop]$
```

Fig: Hadoop Installation – Hadoop Configuration Files

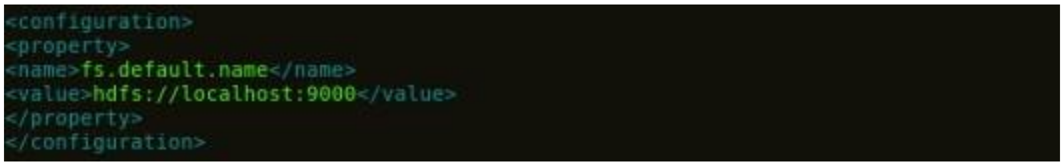
Step 7: Open *core-site.xml* and edit the property mentioned below inside configuration tag:

core-site.xml informs Hadoop daemon where NameNode runs in the cluster. It contains configuration settings of Hadoop core such as I/O settings that are common to HDFS & MapReduce.

Command: vi core-site.xml



```
edureka@localhost:~/hadoop-2.7.3/etc/hadoop
File Edit View Search Terminal Help
[edureka@localhost hadoop]$ vi core-site.xml
```



```
<configuration>
<property>
<name>fs.default.name</name>
<value>hdfs://localhost:9000</value>
</property>
</configuration>
```

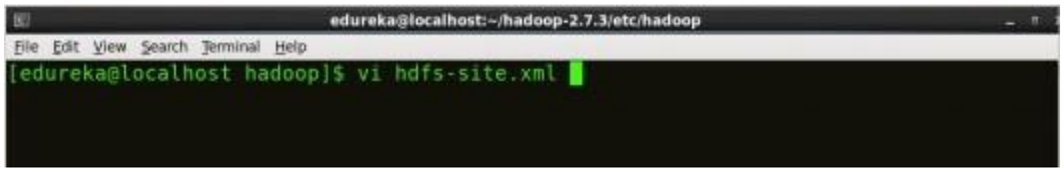
Fig: Hadoop Installation – Configuring core-site.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<configuration>
<property>
<name>fs.default.name</name>
<value>hdfs://localhost:9000</value>
</property>
</configuration>
```

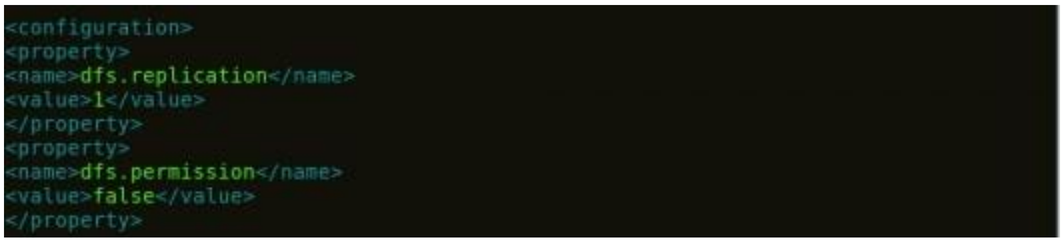
Step 8: Edit *hdfs-site.xml* and edit the property mentioned below inside configuration tag:

hdfs-site.xml contains configuration settings of HDFS daemons (i.e. NameNode, DataNode, Secondary NameNode). It also includes the replication factor and block size of HDFS.

Command: vi hdfs-site.xml


edureka@localhost:~/hadoop-2.7.3/etc/hadoop
File Edit View Search Terminal Help

```
[edureka@localhost hadoop]$ vi hdfs-site.xml
```



```
<configuration>
<property>
<name>dfs.replication</name>
<value>1</value>
</property>
<property>
<name>dfs.permission</name>
<value>false</value>
</property>
```

Fig: Hadoop Installation – Configuring hdfs-site.xml

```xml

<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<configuration>
<property>
<name>dfs.replication</name>
<value>1</value>
</property>
<property>
<name>dfs.permission</name>
<value>false</value>
</property>
</configuration>

```

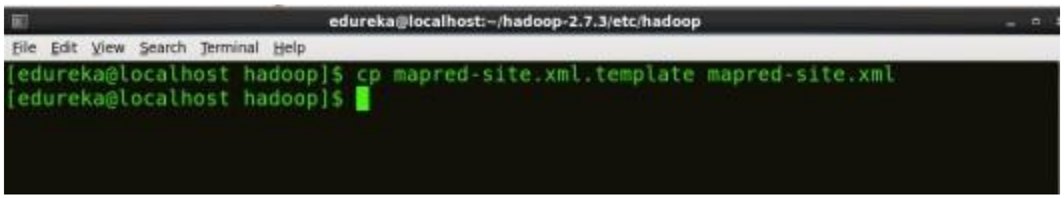
Step 9: Edit the *mapred-site.xml* file and edit the property mentioned below inside configuration tag:

mapred-site.xml contains configuration settings of MapReduce application like number of JVM that can run in parallel, the size of the mapper and the reducer process, CPU cores available for a process, etc.

In some cases, mapred-site.xml file is not available. So, we have to create the mapred-site.xml file using mapred-site.xml template.

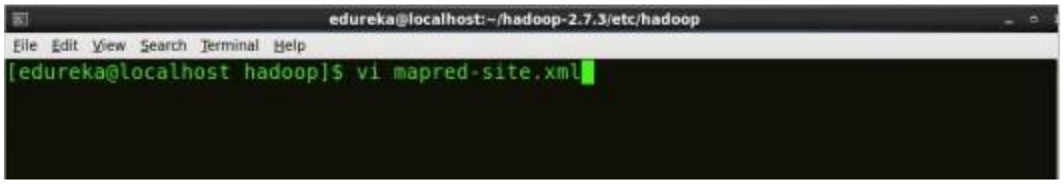
Command: cp mapred-site.xml.template mapred-site.xml

Command**:** vi mapred-site.xml.


edureka@localhost:~/hadoop-2.7.3/etc/hadoop

File Edit View Search Terminal Help

```
[edureka@localhost hadoop]$ cp mapred-site.xml.template mapred-site.xml
[edureka@localhost hadoop]$
```


edureka@localhost:~/hadoop-2.7.3/etc/hadoop

File Edit View Search Terminal Help

```
[edureka@localhost hadoop]$ vi mapred-site.xml
```

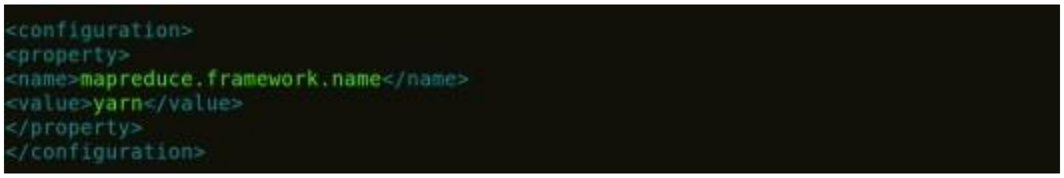

```
<configuration>
<property>
<name>mapreduce.framework.name</name>
<value>yarn</value>
</property>
</configuration>
```

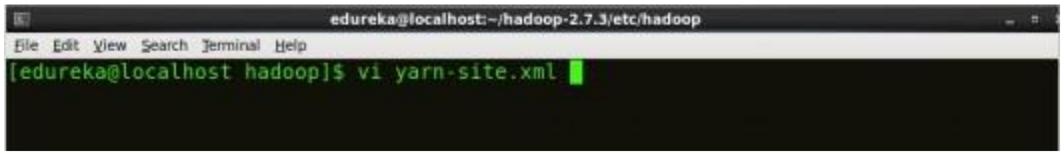
Fig: Hadoop Installation – Configuring mapred-site.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<configuration>
<property>
<name>mapreduce.framework.name</name>
<value>yarn</value>
</property>
</configuration>
```

Step 10: Edit *yarn-site.xml* and edit the property mentioned below inside configuration tag:

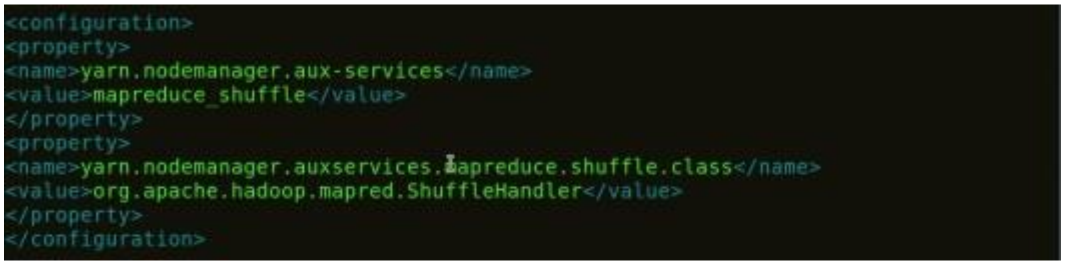
yarn-site.xml contains configuration settings of ResourceManager and NodeManager like application memory management size, the operation needed on program & algorithm, etc.

Command**:** vi yarn-site.xml


edureka@localhost:~/hadoop-2.7.3/etc/hadoop

File Edit View Search Terminal Help

```
[edureka@localhost hadoop]$ vi yarn-site.xml
```



```
<configuration>
<property>
<name>yarn.nodemanager.aux-services</name>
<value>mapreduce_shuffle</value>
</property>
<property>
<name>yarn.nodemanager.auxservices.mapreduce.shuffle.class</name>
<value>org.apache.hadoop.mapred.ShuffleHandler</value>
</property>
</configuration>
```

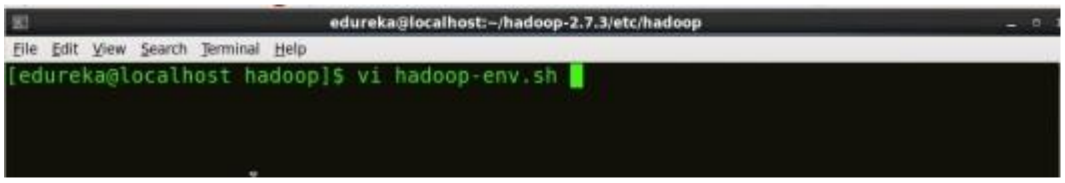
Fig: Hadoop Installation – Configuring yarn-site.xml

```
<?xml version="1.0">
<configuration>
<property>
<name>yarn.nodemanager.aux-services</name>
<value>mapreduce_shuffle</value>
</property>
<property>
<name>yarn.nodemanager.auxservices.mapreduce.shuffle.class</name>
<value>org.apache.hadoop.mapred.ShuffleHandler</value>
</property>
</configuration>
```

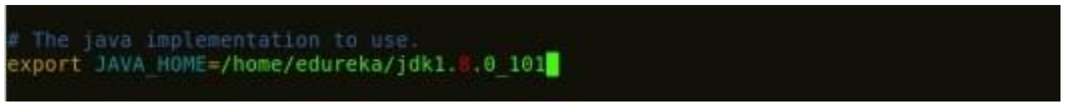
Step 11: Edit *hadoop-env.sh* and add the Java Path as mentioned below:

hadoop-env.sh contains the environment variables that are used in the script to run Hadoop like Java home path, etc.

Command: vi hadoop–env.sh



```
edureka@localhost:~/hadoop-2.7.3/etc/hadoop
File  Edit  View  Search  Terminal  Help
[edureka@localhost hadoop]$ vi hadoop-env.sh
```



```
# The java implementation to use.
export JAVA_HOME=/home/edureka/jdk1.8.0_101
```

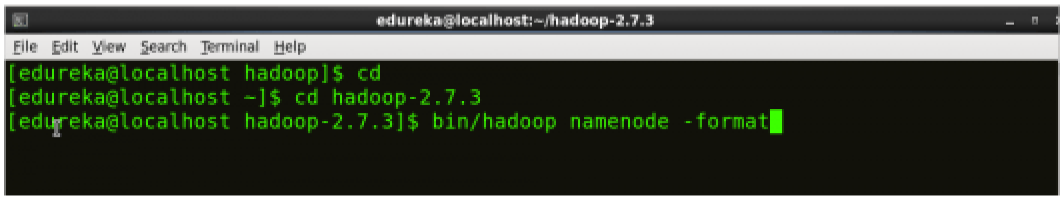
Fig: Hadoop Installation – Configuring hadoop-env.sh

Step 12: Go to Hadoop home directory and format the NameNode.

Command: cd

Command: cd hadoop-2.7.3

Command: bin/hadoop namenode -format


edureka@localhost:~/hadoop-2.7.3
File Edit View Search Terminal Help

```
[edureka@localhost hadoop]$ cd
[edureka@localhost ~]$ cd hadoop-2.7.3
[edureka@localhost hadoop-2.7.3]$ bin/hadoop namenode -format
```

Fig: Hadoop Installation – Formatting NameNode

This formats the HDFS via NameNode. This command is only executed for the first time. Formatting the file system means initializing the directory specified by the dfs.name.dir variable.

Never format, up and running Hadoop filesystem. You will lose all your data stored in the HDFS.

Step 13: Once the NameNode is formatted, go to hadoop-2.7.3/sbin directory and start all the daemons.

Command: cd hadoop-2.7.3/sbin

Either you can start all daemons with a single command or do it individually.

Command: ./start-all.sh

The above command is a combination of ***start-dfs.sh, start-yarn.sh*** & ***mr-jobhistory-daemon.sh***

Or you can run all the services individually as below:

Start NameNode:

The NameNode is the centerpiece of an HDFS file system. It keeps the directory tree of all files stored in the HDFS and tracks all the file stored across the cluster.

Command: ./hadoop-daemon.sh start namenode

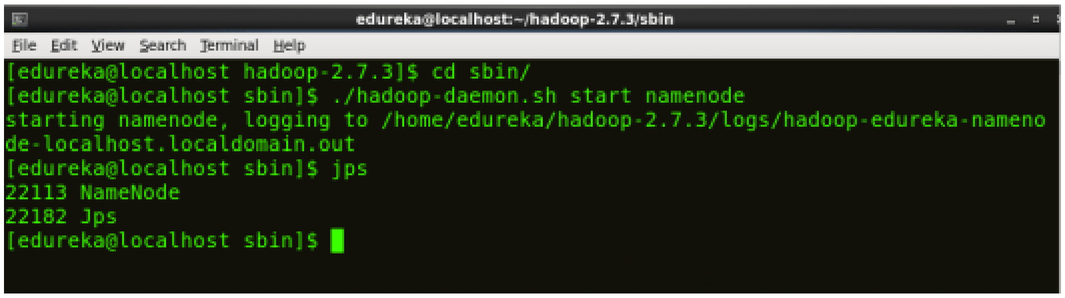

```
[edureka@localhost hadoop-2.7.3]$ cd sbin/
[edureka@localhost sbin]$ ./hadoop-daemon.sh start namenode
starting namenode, logging to /home/edureka/hadoop-2.7.3/logs/hadoop-edureka-nameno
de-localhost.localdomain.out
[edureka@localhost sbin]$ jps
22113 NameNode
22182 Jps
[edureka@localhost sbin]$
```

Fig: Hadoop Installation – Starting NameNode

Start DataNode:

On startup, a DataNode connects to the Namenode and it responds to the requests from the Namenode for different operations.

Command: ./hadoop-daemon.sh start datanode

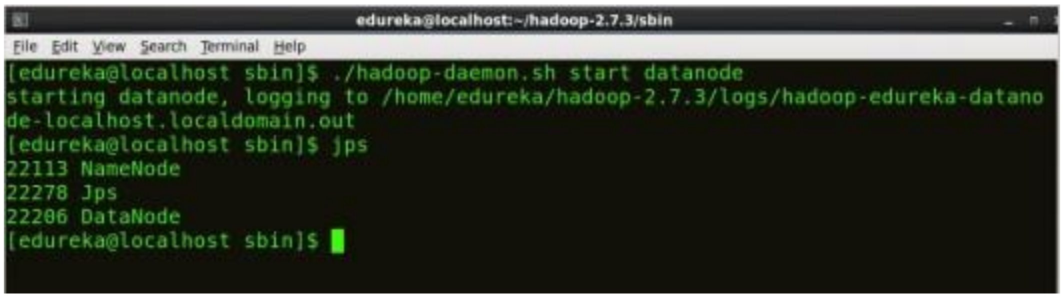

```
[edureka@localhost sbin]$ ./hadoop-daemon.sh start datanode
starting datanode, logging to /home/edureka/hadoop-2.7.3/logs/hadoop-edureka-datano
de-localhost.localdomain.out
[edureka@localhost sbin]$ jps
22113 NameNode
22278 Jps
22206 DataNode
[edureka@localhost sbin]$
```

Fig: Hadoop Installation – Starting DataNode

Start Resource Manager:

ResourceManager is the master that arbitrates all the available cluster resources and thus helps in managing the distributed applications running on the YARN system. Its work is to manage each NodeManagers and the each application's ApplicationMaster.

Command: ./yarn-daemon.sh start resourcemanager

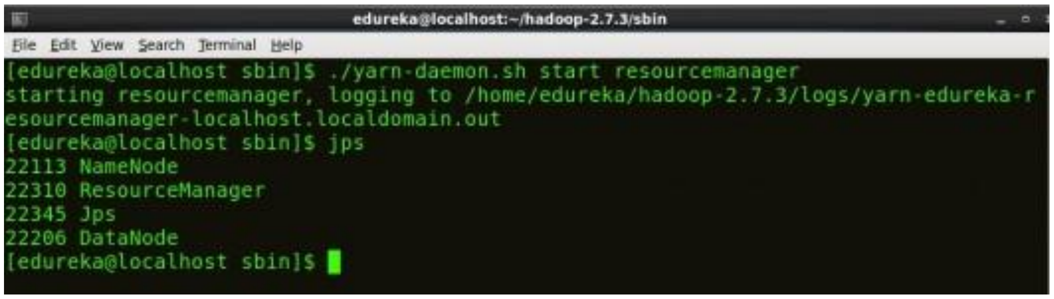
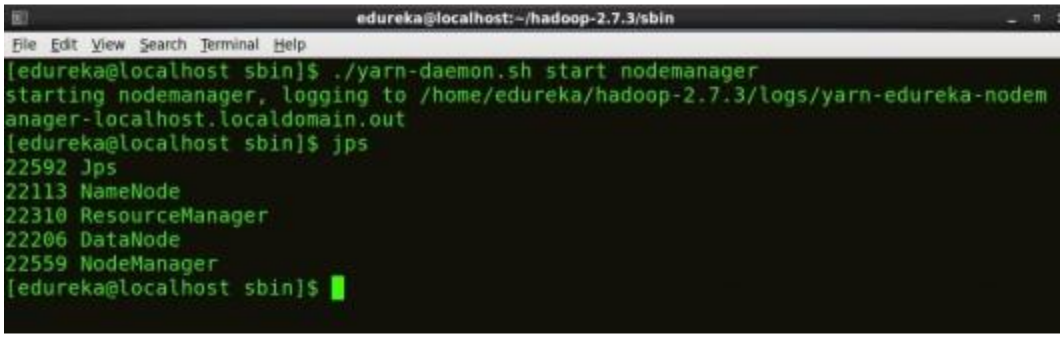

```
[edureka@localhost sbin]$ ./yarn-daemon.sh start resourcemanager
starting resourcemanager, logging to /home/edureka/hadoop-2.7.3/logs/yarn-edureka-r
esourcemanager-localhost.localdomain.out
[edureka@localhost sbin]$ jps
22113 NameNode
22310 ResourceManager
22345 Jps
22206 DataNode
[edureka@localhost sbin]$
```

Fig: Hadoop Installation – Starting ResourceManager

Start NodeManager:

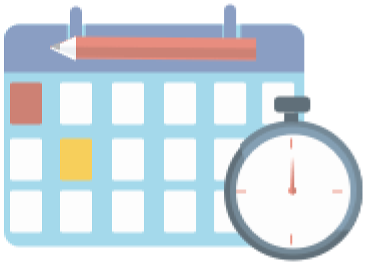
The NodeManager in each machine framework is the agent which is responsible for managing containers, monitoring their resource usage and reporting the same to the ResourceManager.

Command: ./yarn-daemon.sh start nodemanager



edureka@localhost:~/hadoop-2.7.3/sbin
File Edit View Search Terminal Help

```
[edureka@localhost sbin]$ ./yarn-daemon.sh start nodemanager
starting nodemanager, logging to /home/edureka/hadoop-2.7.3/logs/yarn-edureka-nodem
anager-localhost.localdomain.out
[edureka@localhost sbin]$ jps
22592 Jps
22113 NameNode
22310 ResourceManager
22206 DataNode
22559 NodeManager
[edureka@localhost sbin]$
```



See Batch Details

Fig: Hadoop Installation – Starting NodeManager

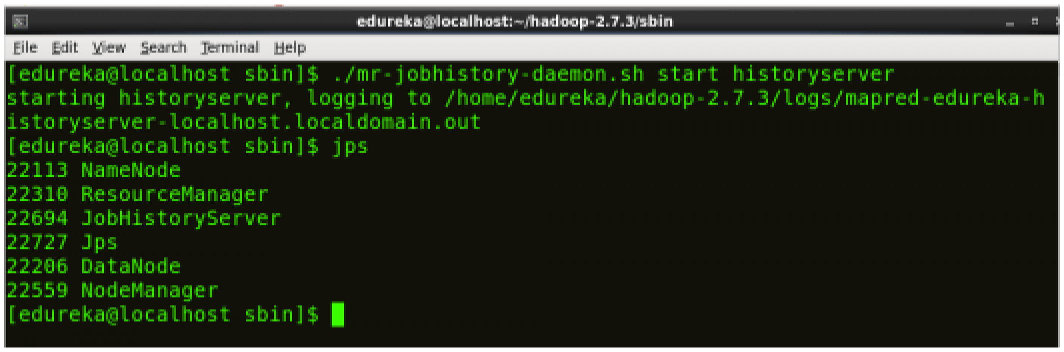
Start JobHistoryServer:

JobHistoryServer is responsible for servicing all job history related requests from client.

Command: ./mr-jobhistory-daemon.sh start historyserver

Step 14: To check that all the Hadoop services are up and running, run the below command.

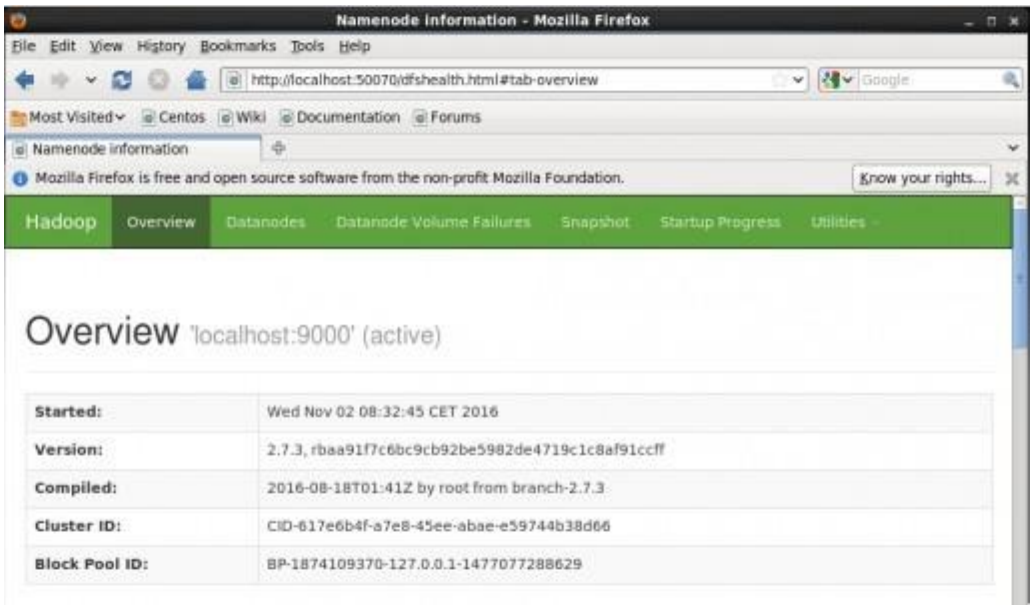
Command: jps



```
[edureka@localhost sbin]$ ./mr-jobhistory-daemon.sh start historyserver
starting historyserver, logging to /home/edureka/hadoop-2.7.3/logs/mapred-edureka-h
istoryserver-localhost.localdomain.out
[edureka@localhost sbin]$ jps
22113 NameNode
22310 ResourceManager
22694 JobHistoryServer
22727 Jps
22206 DataNode
22559 NodeManager
[edureka@localhost sbin]$
```

Fig: Hadoop Installation – Checking Daemons

Step 15: Now open the Mozilla browser and go to **localhost**:**50070/dfshealth.html** to check the NameNode interface.



Namenode information - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost:50070/dfshealth.html#tab-overview
Hadoop
Overview
Datanodes
Datanode Volume Failures
Snapshot
Startup Progress
Utilities

Overview 'localhost:9000' (active)

Started:	Wed Nov 02 08:32:45 CET 2016
Version:	2.7.3, rbaa91f7c6bc9cb92be5982de4719c1c8af91ccff
Compiled:	2016-08-18T01:41Z by root from branch-2.7.3
Cluster ID:	CID-617e6b4f-a7e8-45ee-abae-e59744b38d66
Block Pool ID:	BP-1874109370-127.0.0.1-1477077288629

Fig: Hadoop Installation – Starting WebUI

Congratulations, you have successfully installed a single node Hadoop cluster