

Virtual Machine Ubuntu Test Setup 01

Resources

- [Linux Video Tutorials - YouTube - How to Install Ubuntu 16.04.1 LTS + VMware Tools on Vmware Workstation/Player Easy Tutorial](#)
- [Linux Video Tutorials - YouTube - How to Install VMware Tools on Ubuntu 16.04 LTS Step by Step HD](#)
- [Black Hat Studio - YouTube - install Ubuntu 16.04 On VMware workstation 12](#)
- [Brunolauri - YouTube - How to install Ubuntu 16.04 \(64-Bit\) on VMware workstation 12](#)
- [blacktechhack - YouTube How to step by step install VMware Tools in Ubuntu 14.04 LTS](#)
- [Tech Blog - YouTube - How to Install VMware tools Ubuntu 16.04.1 LTS - Using VMware](#)

Download Ubuntu

- [Main Ubuntu Page](#)
- Next Go to Download Dropdown and select [Desktop](#)
- Ubuntu 20.xx.x LTS - LTS = Long Term Support- which means five years of free security and maintenance updates, guaranteed.
- Select and actuate Download
- Select and "*actuate Not now, take me to the download*"
- Save the ISO file to disk. ISO files are stored at **D:\VMIsolimages**

Create a New Virtual Machine

VMware Preferences

- Before staring VM configuration go to **Edit->Preferences**.
- Under **Workspace** make certain **Default location for virtual machines** is properly configured.
- Under **USB -> Connections**, check **Ask me what to do**.
- Under **Memory->Reserved memory**, select **13773MB**.
- Under **Memory->Additional memory**, select **Allow most virtual machine memory to be swapped**.
- Under **Devices**, check **Enable virtual printers**.

Create The Virtual Machine Files Directory

Virtual Machine files will be housed in a directory of your choosing. The default path was set above under, **VMware Preferences**. Before beginning the VM creation process, create the directory where the files for this particular VM will be stored.

It is recommended that you create to sub-directories to differentiate between operational files and support files.

Example:

D:\VirtualMachines\UbuntuBase-3\Operations
D:\VirtualMachines\UbuntuBase-3\Support

Virtual NIC Option

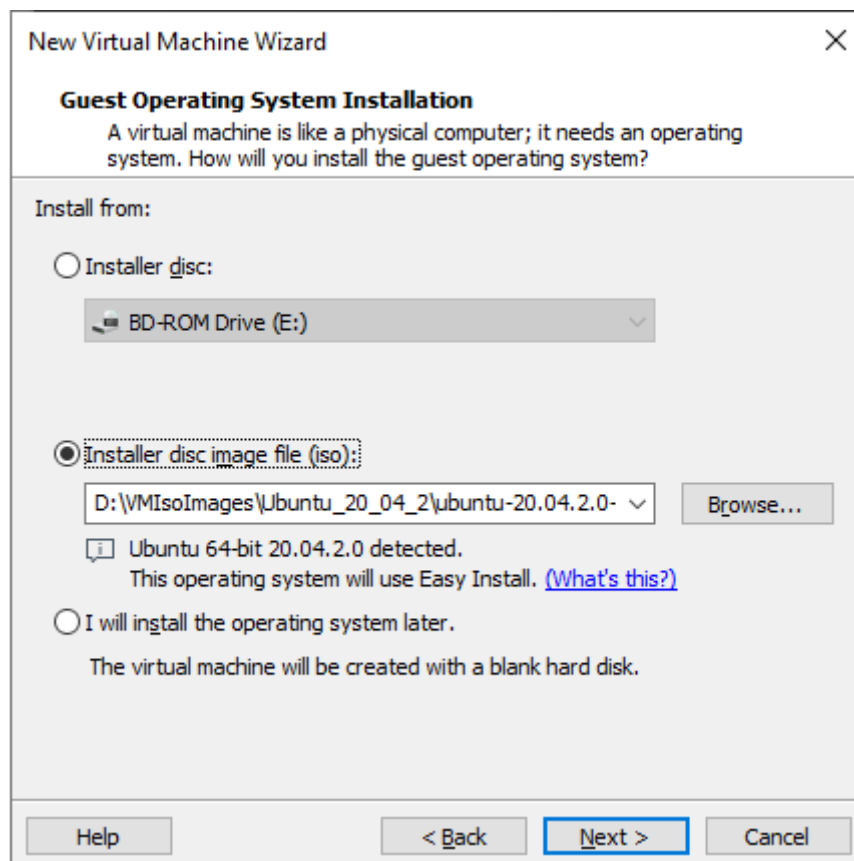
If required, add a virtual Network Interface Card (NIC) to your virtual machine.

Reference:

[Add a Virtual Network Adapter to a Virtual Machine](#)

Configure the Virtual Machine

- Inside VMWare Workstation Initiate - **Create a New Virtual Machine**
- Display: **Welcome to the New Virtual Machine Wizard**
- Check box: **Typical(recommended)**
 - Click Button: Next
- **Install From disc image file(iso)**
 - Select location of iso file



- Check **I will install the operating system later.**

New Virtual Machine Wizard

Guest Operating System Installation

A virtual machine is like a physical computer; it needs an operating system. How will you install the guest operating system?

Install from:

☐ Installer disc:

BD-ROM Drive (E:)

☐ Installer disc image file (iso):

D:\VMisoImages\Ubuntu_20_04_3_LTS\ubuntu-20.04 Browse...

☒ I will install the operating system later.

The virtual machine will be created with a blank hard disk.

Help < Back Next > Cancel

Click Button: **Next**

- Select a Guest Operating System
- Check Box: **Linux**
- Under Version Drop Down Box Select: **Ubuntu 64-bit**
- Click Button: **Next**
- **Name the Virtual Machine**
 - Virtual Machine Name: **UbuntuBase-3**
 - Location: Location means the location of the virtual machine operations files on your disk. Browse and select the appropriate location.
 - Location: **D:\VirtualMachines\UbuntuBase-3\Operations**

- Click Button: **Next**

- **Specify Disk Capacity:**

- Maximum disk size is 60Gb. VMware will suggest a recommended size for your operating system.
- Select and check the box:

Split virtual disk into multiple files

- Click Button: **Next**

- **Configure Hardware**

- Click Button: **Configure Hardware**
- Set Memory: **4-Gigs (4096Mb)**
 - Note: Some suggest 2-gigs = 2048Mb
- Set Processors: (2 x 2 = 4 Processor Cores)
 - Number of Processors: 2
- Number of cores per processor: 2
 - Note: Some suggest 1 Processors and 4 cores
- Virtualization Engine
 - Take Default: Automatic
- New CD/DVD Auto Detect

- Check Device Status box: **Connect at power on.**
 - **Check Button: Use ISO image file**
 - **Browse for ISO Image File**
- **Configure Networking**
 - Set Network to:
 - **Bridged**
 - **Replicate physical network connection state**
 - Physical Network Config will allow this VM access to your network and Router. The Router will then assign a separate IP address to this VM.
- Configure USB Controller
 - You may want to pass through to USB Stick depending on setup
 - Uncheck and De-select: **Show all USB input devices**
 - Check: **Share Bluetooth devices with the virtual machine**
- Remove Sound Card
 - Check: **Connect at power on**
 - Check: **Use default host sound card**
- Printer
 - Device Status: Check **Connect at power on**
- Configure: Display
 - Check Accelerate 3D graphics
 - Use host setting for monitors
 - Take Defaults: Set Maximum Graphics Memory: 8Gb
- Click Button: Close
- **Return to Dialog: Ready To Create Virtual Machine Dialog**
- **Click Finish**
- Set Working Directory

Under **VM->Settings->Options->Working directory**, browse and select the working directory where Suspend, snapshots and other administrative files associated with this virtual machine will be saved.

D:\VirtualMachines\UbuntuBase-3\Support

Select **Ok** to save.

Before You Power-On the VM to Install Operating System

Edit the VM Configuration and make certain the CD-ROM is pointing and the ISO Directory and File Name.

- From main VM menu choose **Edit virtual machine settings.**
- Under Connection
 - Check Use ISO image file.
 - Browse to ISO location and store location in dialog box.
- Click **Ok**

Install Ubuntu Operating System

Note: Installation of Ubuntu may take 30-minutes.

From Left Panel of main Display: Click **Power On This Virtual Machine**

Beware!!! The screen may be cut off at the bottom left. You may not be able to see the continue buttons. If this is the case, you will need to tab to the continue. Be advised that the first button on bottom left of screen is back button. You may need to tab to the button on the immediate right which is off screen and not visible. That is the Continue button.

Power On The Virtual Machine

- From top of VM menu page select: **Power on this virtual machine**
- At the very bottom of the screen a Yellow Dialog Box will display a message:

When you are done and the operating system boots up, click "I Finished Installing"

- This is an informational message. Wait until the installation is completely finished and the system reboots, before clicking the "I Finished Installing" button.

Install

Select your language (**English**) then Click Button **Install Ubuntu**

Keyboard Layout

- Select Language from Left Panel: English
- Select Language from Right Panel: English(US)
- Click: Continue

Updates and other software

- What apps would you like to install to start with?
 - Select: **Normal Installation**
- Other Options
 - Select: **Download updates while installing Ubuntu**
- Click Button: **Continue**

Installation Type

- Check box: **Erase disk and install Ubuntu**
- Click and actuate button: **Install Now**
- Pop Up Displays - Write the changes to disks?
 - Click Button: **Continue**

Where Are you?

- Graphically select your time zone: Chicago
- Click Button: Continue

Who Are You Page

Your name: John Smith

Your computer's name: ubuntu16

Pick a user name: john

Choose a password: pass!WORD497

Confirm a password: pass!WORD497

Check Box: ***Require my password to log in***

Click Button: ***Continue***

Primary Installation Sequence

- By default, you should be configured as superuser
- At this point the primary installation process continues with no further user input required.
- After Installation process completes, you will be asked to reboot the machine. Execute the Reboot procedure.

Log In to New Machine and Complete Installation

- Enter your password
- Explore Left Panel and familiarize yourself with primary operating system applications.
- Top button is the search function.

Perform Updates

Open Terminal (Ctrl+Alt+T)

- At Command Prompt - Execute the update operation
- Execute update command:

```
sudo apt update <ENTER>
```

- Execute upgrade command:

```
sudo apt upgrade -y <ENTER>
```

Return To GUI

- Click Left Panel Icon A = Applications = Ubuntu Software
 - Click Top Menu Item: Updates
 - Update all software items
- Execute update command again:

Open Terminal (Ctrl+Alt+T)

- At Command Prompt - Execute the update operation
- Execute update command:

```
sudo apt update <ENTER>
```

- Execute upgrade command:

```
sudo apt-get upgrade <ENTER>
```

- After completion enter command clear to clear the screen.

```
clear <ENTER>
```

Install VMware Tools

VMware Tools are probably already installed.

To verify installation, run command:

```
sudo apt search open-vm-tools
```

if the tools are already installed, then there is no need to re-install them. In this case, skip this installation and proceed to the next step.

Method-1 Install From Operating System (Preferred Method)

Ubuntu distributes VMware tools with the operating system.

Updates are likewise distributed with the operating system. Therefore, VMware recommends that operating system be used to install VMware tools from the command line.

See this VMware Knowledge Base Article.

[VMware support for open-vm-tools \(2073803\)](#)

- Execute get VM tools command

```
sudo apt install open-vm-tools-desktop -y
```

- Execute update command:


```
sudo apt update
```

- Execute upgrade command again:

```
sudo apt upgrade -y
```

Method #2

- From VMWare IDE menu choose VM.
- Choose Re-Install VMWare Tools
- VMWare Tools is on the CD-ROM Drive
- Execute **blkid** command to list available drives

```
blkid '<ENTER>'
```

Probable CD-ROM

```
/dev/sr0: UUID="2020-10-13-18-03-38-00" LABEL="VMware Tools" TYPE="iso9660"
```

- Create Mount Point

Next, create a mount point for where you'd like to mount the CD to. Or, just pick an empty directory somewhere on your computer, if you already have a place in mind.

```
$ sudo mkdir /mnt/cdrom
```

- Configure Mount Point

Now we can use the [mount command](#) to map the device file to the directory we've just created.

```
$ sudo mount /dev/sr0 /mnt/cdrom
mount: /mnt/cdrom: WARNING: device write-protected, mounted read-only.
```

- Access The CD-ROM

Your CD or DVD should now be accessible in the directory which it was mounted.

```
$ cd /mnt/cdrom
```

- Verify CD-ROM Access

You can also verify that the mount was successful by using the `mount` command without further options.

```
$ mount | grep cdrom
```

- Copy VMwareTools-10.3.23-17030940.tar.gz To Working Directory

```
sudo mkdir -p ~/work/vmwaretools/
```

```
sudo cp /mnt/cdrom/VMwareTools-10.3.23-17030940.tar.gz ~/work/vmwaretools
```

- Unzip the tarball to Working Directory

```
cd ~/work/vmwaretools
```

```
sudo tar zxvf VMwareTools-10.3.23-17030940.tar.gz
```

Unmount the CD-ROM

```
umount /mnt/cdrom
```

- Install VMWare Tools

```
cd cd ./vmware-tools-distrib
```

```
$ sudo /vmware-install.pl
```

Power Off The Computer

- Click the Power Button in top Right-corner of GUI
- Select Power Off/Log Out from Drop Down
- Select: **Log Out**
- Again, Click the Power Button in top Right-corner of GUI
- Select Power Off/Log Out from Drop Down
- Select Power Off and shut down the VM

Linux Tooling

Startup

- In Linux use **Ctrl+Shift+C** and **Ctrl+Shift+V** for copy and paste.

Update The Linux System

```
sudo apt update && sudo apt full-upgrade -y  
sudo apt autoremove
```

Useful Commands

Access *root* Directory

```
cd /
```

Install Networking Tools

Networking Tools Install Command

```
sudo apt install net-tools
```

Verify Networking Tools Installation

This command should display the IP address and MAC address for this VM Machine.

```
ip a
```

Note The IP Address

Make a note of the IP Address and MAC Address resulting from the following command:

```
ip a
```

Time Zone

Query Current Time and Zone Information

```
timedatectl<ENTER>
```

This will display current time and time zone data. It will also indicate whether the NTP time synchronization service is installed and active.

```
~ timedatectl
          Local time: Sun 2021-05-23 16:24:31 PDT
          Universal time: Sun 2021-05-23 23:24:31 UTC
            RTC time: Sun 2021-05-23 23:24:32
            Time zone: America/Los_Angeles (PDT, -0700)
System clock synchronized: yes
            NTP service: active
            RTC in local TZ: no
```

If the NTP service is active, skip the step below and DO NOT INSTALL NTP.

Install NTP

[How to setup NTP server on Ubuntu 20.04 - Eldernode Blog](#)

If NTP is not installed, install it now. This will coordinate machine time with an atomic clock.

```
sudo apt install ntpdate
```

After installation, link to a server clock for continued updates.

Reference: [How to sync time on Ubuntu 20.04 Focal Fossa Linux - LinuxConfig.org](#)

Set the Time Zone

Executing the command below will bring up a series of prompts which allow you to select your home continent, country and time zone. Make the appropriate selections:

```
cd /usr/share/zoneinfo <ENTER>

tzselect<ENTER>
```

After you set the time zone, log off, power off and log back in. This will activate the new time zone.

Change User Data

It may be necessary to change the user data on your VM.

Change hostname

[How to Change Your Hostname \(Computer Name\) on Ubuntu Linux \(howtogeek.com\)](https://howtogeek.com/2012/06/how-to-change-your-hostname-computer-name-on-ubuntu-linux/)

Host names can only contain letters (a through z), digits (0 through 9), and the hyphen character (-), and the period character (.). A hostname must begin and end with a letter or number — not a hyphen or period. Letters are also case-insensitive, so “COMPUTER” is equivalent to “computer.” The hostname must be between 2 and 63 characters long, although you’ll probably find shorter hostnames more convenient.

Query and Display Host Name

Display the current host name:

```
hostnamectl<ENTER>
```

Edit Your /etc/hostname File

```
sudo subl /etc/hostname
```

Replace the old hostname with the new hostname.

Edit Your /etc/hosts File

```
sudo subl /etc/hosts
```

The old hostname will be styled like this:

```
127.0.1.1 your-old-hostname
```

Simply replace the old host name but do not disturb the number (127.0.1.1 in this example).

Change Your Hostname Immediately

```
sudo hostname your-new-hostname
```

To complete the hostname change and make it permanent, log off and log back on again.

Change GECOS (User) Data

1. Install *finger*:

```
sudo apt update && sudo apt full-upgrade -y  
sudo apt install finger
```

2. List all GECOS information about your account (Example Account: 'mike'):

```
finger mike -l
```

3. Change the "Full Name" field in the GECOS record for account 'mike':

```
sudo chfn -f UbuntuVM02 mike
```

DO NOT put any punctuation marks like quotation marks or tick marks around the new name (Example: UbuntuVM02-Mike).

cURL

The cURL [Linux command](#) can use various network protocols to download and upload data on a [Linux system](#). It's a tool that proves handy for downloading files, in which it functions similarly to [wget](#), as well as debugging, performing downloads with authentication, uploading via FTP, and more.

Install **curl**:

```
sudo apt update  
  
sudo apt install -yqq curl
```

Wget

The free, open-source wget command-line tool downloads content from the Web through a variety of protocols, including HTTP, secure HTTPS and FTP. Wget works in the background, even when you're not logged on, to run complicated or long download scripts without the need for user input. Wget also follows links and monitors FTP servers for updates. Wget isn't installed by default on most editions of Ubuntu, but you can add it with Ubuntu's apt-get utility.

Determine If Wget is already installed:

```
wget --version
```

If Wget is already installed, you can skip the installation.

Install Wget

```
sudo apt update  
  
sudo apt install wget
```

Verify that wget was downloaded properly by entering the following command:

```
wget --version
```

Git

Resources

[GitHub - DevInsideYou/install-git](#)

[How To Install Git On Ubuntu 20.04 - Ubuntu-Server.com](#)

Install Git

1. Install **Git** before you install anything else.
2. Install the software-properties-common package on your system, which contains add-apt-repository command.

```
sudo apt install -y software-properties-common
```

3. Add the PPA (Personal Packages Archive) to your system using the following command:

```
sudo add-apt-repository -y ppa:git-core/ppa
```

4. Next, install **Git** on your system:

```
sudo apt install git -y
```

5. Finally, setup and configure **Git**.

```
git config --global user.name "Amarillo Mike"  
git config --global user.email "mike@paladinacs.com"
```

6. Verify the **Git** installation:

```
git --version  
# Expected Result: git version 2.31.1 (or later)
```

zsh Shell Installation

Resources

1. [Working with Linux - Terminal, Zsh & Oh My Zsh - YouTube](#)
2. [Learn the basics of the ZSH shell](#)

Install zsh

```
sudo apt update
sudo apt install -y zsh
```

Verify Installation:

```
zsh --version
# Expected Result: zsh 5.8 (x86_64-ubuntu-linux-gnu) (or later)
```

Check to ensure **zsh** is in your authorized shells list:

```
cat /etc/shells
```

Make zsh Your Default Shell

```
sudo chsh -s $(which zsh)
```

Log Out and Log Back In Again

Log out through the GUI, don't execute 'shutdown' command.

nano

GNU nano is an easy to use command line text editor for Unix and Linux operating systems. It includes all the basic functionality you'd expect from a regular text editor, like syntax highlighting, multiple buffers, search and replace with regular expression support, spellchecking, UTF-8 encoding, and more.

Resources

[nano \(nano-editor.org\)](#)

[Cheatsheet for nano \(nano-editor.org\)](#)

[How to Install and Use Nano Text Editor in Linux - TecNStuff](#)

Verify/Install nano

1. The **nano** editor is probably already installed.
2. Run these commands to ensure installation of latest version.

```
nano --version
```

If nano is installed, you should see something like this:

```
nano --version
# Expected Result: GNU nano, version 4.8 (or later)
```

If nano IS installed, skip this installation and proceed to the next step.

3. If nano is NOT installed, install it now:

```
sudo apt update
sudo apt install nano
```

4. Test the **nano** installation:

```
nano --version
# Expected Result: GNU nano, version 4.8 (or later)
```

Oh My Zsh (Enhanced Shell)

Remember, you **MUST** install Zsh successfully, before installing Oh My Zsh.

Resources

1. [GitHub - DevInsideYou/install-zsh](#)
2. [Install Home · ohmyzsh/ohmyzsh Wiki · GitHub](#)
3. [Working with Linux - Terminal, Zsh & Oh My Zsh - YouTube](#)
4. [Learn the basics of the ZSH shell](#)

Install Oh My Zsh

Taken from [ohmyzsh](#) website.

1. Install Oh My Zsh

```
sh -c "$(curl -fsSL
https://raw.githubusercontent.com/ohmyzsh/ohmyzsh/master/tools/install.sh)"
```

2. Edit file `~/.zshrc` to see Themes and Plugins.
3. Sticking with default theme: **robbyrussell**

4. Did not select any Plugins
5. Log Out and Log On again.
6. Test Zsh Installation:

```
echo $SHELL  
# Expected Result: /bin/zsh or similar
```

6. Test Zsh Version:

```
$SHELL--version  
# Expected Result: 'zsh 5.4.2' or similar
```

Set Screen Time-Out

Extent the Time-Out period or eliminate it altogether in order to streamline the package installation process.

Control Time Out Through GUI

Select the System Settings from the icon located on the far right in your top panel. Once there select the **Power** settings. It will look as I've shown below. Change the "Power Saving" to **Never**, and change the "Suspend & Power Button" to "**Automatic Suspend - OFF**".

Control Screen Lock Through Command Line Interface (CLI)

If you want to turn off the screen lock, run:

```
gsettings set org.gnome.desktop.screensaver lock-enabled false
```

Consolidating Bash Aliases

[How to create a permanent Bash alias on Linux/Unix - nixCraft \(cyberciti.biz\)](https://www.cyberciti.biz/faq/how-to-create-a-permanent-bash-alias-on-linux-unix/)

1. Modify rc File

A. If You Have A ~/.zshrc File (And you certainly should!)

Use **nano** to perform file edits.

Example:

```
nano /home/mike/.zshrc
```

The following lines must be present in the **~/.zshrc** file:

```
if [ -f ~/.bash_aliases ]; then
. ~/.bash_aliases
fi
```

B. If You DO NOT HAVE a ~/.zshrc File

The following lines must be present in the ~/.bashrc file:

```
if [ -f ~/.bash_aliases ]; then
. ~/.bash_aliases
fi
```

C. OR - You may want to put these lines in both ~/.bashrc and ~/.zshrc

```
if [ -f ~/.bash_aliases ]; then
. ~/.bash_aliases
fi
```

2. Create the .bash_aliases File

Create the file ~/.bash_aliases. Elevated privileges are required.

```
sudo touch ~/.bash_aliases
sudo chmod a+rw ~/.bash_aliases
```

Mike's List of Aliases

1. Add these aliases to the ~/.bash_aliases file.

```
alias aliases="cat ~/.bash_aliases"

alias cls="clear"

alias CLS="clear"
```

2. Activate these aliases immediately by running this command:

```
source ~/.zshrc
```

Install p7-Zip for Linux

[p7zip](#) - the port of the command line version of 7-Zip to Linux/Posix.

Resources:

[7-Zip for Linux — Download Best Archiver for your Linux Distro](#)

[How To Install 7-Zip On Linux](#)

[7-Zip Web Site - Download](#)

p7zip Usage

```
usage: /usr/bin/p7zip [options] [--] [ name ... ]
```

Help

For help execute the command:

```
p7zip -h
```

p7zip Options

```
Options:
  -c --stdout --to-stdout      output data to stdout
  -d --decompress --uncompress decompress file
  -f --force                   do not ask questions
  -k --keep                    keep original file
  -h --help                    print this help
  --                           treat subsequent arguments
                              as file names, even if they
                              start with a dash
```

Installation

Determine if p7Zip is available

```
sudo dpkg -s p7zip-full | grep Status
or
sudo dpkg -s p7Zip-full | grep Status
```

If 7-Zip IS installed you will received the response:

```
Status: install ok installed
```

Otherwise the package is NOT installed and you will need to install it.

Installation Procedure

1. On the Terminal, type in this command:

```
sudo apt update
```

2. You will be asked to enter your password. Please do so as this first command will update your system.
3. Once the system update finishes, you will need to type in this command:

```
sudo apt-get install p7zip-full p7zip-rar
```

p7zip-rar will allow you to extract **rar** files.

4. Again, you will need to enter your system password to confirm the command request.
5. The command will now start to download the necessary files from the repository
6. As soon as all the files are downloaded, you will be asked if you want to continue or not with the installation. Type "**Y**" for yes and "**N**" for no. In this case, just type "**Y**" and press the Enter key.
7. This will now initiate the unpacking of the downloaded files and start setting them up for installation.
8. When the command line appears again, it means the installation is through.
9. To make sure that the **7Zip** really is installed on your Linux computer you will need to type in this command:

```
sudo dpkg -s p7zip-full | grep Status
```

The next line should say:

```
status: install ok installed
```

10. That's it. You now have **p7Zip** installed on your Ubuntu system.

htop

A Linux utility for displaying crucial information about the system's processes. It can be considered as a Linux counterpart of Windows Task Manager.

[A Guide to the htop command in Linux - LinuxForDevices](#)

Note: snap will install a later version

1. Determine if **htop** is installed:

```
htop <Enter>
```

2. Install **htop**:

```
sudo apt install -y htop  
or  
sudo snap install htop
```

2. Test **htop**:

```
htop<ENTER>
```

3. Click **F10** or press **Q** to quit **htop**.

Python

[How to Install Python PIP Tool on Ubuntu 20.04 – Linux Hint](#)

1. Update and upgrade your **apt**:

```
sudo apt update && sudo apt full-upgrade -y
```

2. Add Universe Repository:

```
sudo add-apt-repository universe
```

3. Install python2:

```
sudo apt install -y python2
```

4. Download get-pip.py:

```
sudo curl https://bootstrap.pypa.io/pip/2.7/get-pip.py --output get-pip.py
```

5. Process get-pip.py:

```
sudo python2 get-pip.py
```

6. Install PIP for Python 3:

```
sudo apt install -y python3-pip
```

7. Verify Python2 Installation:

```
pip --version  
# Expected Result: python 2.7
```

8. Verify Python3 Installation:

```
pip3 --version  
# Expected Result: python 3.8 (or later)
```

9. List and identify the installed python packages which will be added to your path:

```
ls /usr/bin/python*
```

10. Use **nano** to edit and add the following directories to your \$PATH in ~/.profile:

```
/usr/bin/python2  
/usr/bin/python2.7  
/usr/bin/python3  
/usr/bin/python3-futurize  
/usr/bin/python3-pasteurize  
/usr/bin/python3-config  
/usr/bin/python3.8  
/usr/bin/python3.8-config
```

Example:

```
PATH="$PATH:/usr/bin/python2:/usr/bin/python2.7:/usr/bin/python3:/usr/bin/py  
thon3-futurize:/usr/bin/python3-pasteurize:/usr/bin/python3-  
config:/usr/bin/python3.8:/usr/bin/python3.8-config"
```

11. Add the items to your path immediately:

Add the "PATH" to your **~/.profile** file. Then activate the "PATH" with the following command.

```
source ~/.profile
```

12. Verify the path:

```
echo $PATH
```

Install Node.js

Resources

[distributions/README.md at master · nodesource/distributions · GitHub](#)

[Installing Node.js](#)

[Install Node.js Locally with Node Version Manager \(nvm\)](#)

Run update and upgrade

```
sudo apt update && sudo apt full-upgrade -y
```

Install Node.js:

Note: The Latest Stable Release is the 16.x series. The current release is the 17.x series.

```
curl -fsSL https://deb.nodesource.com/setup_17.x | sudo -E bash -  
  
sudo apt install -y nodejs
```

Verify the installation:

```
node -v  
# Expected Result: v16.1.0 (or later)
```

Install *npm*

Node Package Manager (*npm*) is Node's official package manager, used for installing and managing package dependencies.

Resources

[How to update npm package to latest version](#)

Determine If *npm* is Installed

```
npm -v  
# Expected Result: v8.1.1 (or later)
```

If *npm* is already installed, skip this installation and proceed to the next step.

Install Commands

The following commands will install and update the latest stable release of *npm*.

```
sudo npm cache clean -f
sudo npm install -g n
sudo n stable
```

Verify Installation

```
npm -v
# Expected Result: v8.1.1 (or later)
```

Install Node Version Manager NVM

The [Node Version Manager \(NVM\)](#) is an open source version manager for [Node.js \(Node\)](#). NVM is easy to install and understand, and works on any POSIX-compliant shell (for example, sh or bash). NVM allows you to easily install different versions of Node and switch between them on a per-shell basis. This guide describes how to install NVM, and how to use it to install and run different versions of Node.

Resources

[Node Version Manager - Github](#)

Installing and Configuring NVM

These instructions are generally valid for most Linux distributions, although some of the `.bashrc` directives might vary slightly on different shells. You can install and use NVM regardless of whether you have installed Node already. NVM alters `path` variables to select different versions of Node, so it works with pre-existing installations.

Install NVM

We recommend you install NVM using either `curl` or `wget`.

To install NVM using `curl`, run the following command:

```
curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.39.0/install.sh | bash
```

To install NVM using `wget`, run the following command:

```
wget -qO- https://raw.githubusercontent.com/nvm-sh/nvm/v0.39.0/install.sh | bash
```

Either of these commands may produce a message requiring another command. Be sure to read the output messages. You may be asked to run the following command:

```
export NVM_DIR="$HOME/.nvm"
[ -s "$NVM_DIR/nvm.sh" ] && \. "$NVM_DIR/nvm.sh" # This loads nvm
[ -s "$NVM_DIR/bash_completion" ] && \. "$NVM_DIR/bash_completion"
```

As an alternative, you could simply Log Off and Log On again.

Confirm you have successfully installed NVM.

```
command -v nvm
```

If the installation was successful, NVM returns `nvm`. If you see the message `nvm: command not found`, confirm the original installation completed successfully.

You can confirm the version of NVM that is running with the following command:

```
nvm --version
```

Add Node and *npm* Directories to Your Path

List and identify the Node.js and NPM installation directories. You will need to add these to your path.

- List Node Directories:

```
ls /usr/bin/node*
```

- List Node Package Manager (NPM) Directories

```
ls /usr/bin/npm*
```

- The directories to be added to your \$PATH:

```
/usr/bin/node:/usr/bin/npm
```

- User **nano** to edit and add these directories to your \$PATH in the file: `~/.profile`:

```
PATH="$PATH:/usr/bin/python2:/usr/bin/python2.7:/usr/bin/python3:/usr/bin/python3-futurize:/usr/bin/python3-pasteurize:/usr/bin/python3-config:/usr/bin/python3.8:/usr/bin/python3.8-config:/usr/bin/node:/usr/bin/npm"
```

- Add the items to your path immediately:

```
source ~/.profile
```

- Verify the path:

```
echo $PATH
```

Install Ruby and Gem

[How to Install Ruby on Ubuntu 20.04](#)

1. Run update and upgrade:

```
sudo apt update && sudo apt full-upgrade -y
```

2. Install Ruby:

```
sudo apt install -y ruby-full
```

3. Verify the installation. List the version:

```
ruby --version
# Expected result should look something like this:
ruby 2.7.0p0 (2019-12-25 revision 647ee6f091) [x86_64-linux-gnu]
# You may see a later version number
```

4. List and identify the **ruby** and **gem** installation directories. You will need to add them to your \$PATH:

```
ls /usr/bin/ruby*
ls /usr/bin/gem*
```

5. You should see directories that are similar to this:

```
/usr/bin/ruby
/usr/bin/ruby2.7
/usr/bin/gem
/usr/bin/gem2.7
```

6. Use **nano** to edit and add these directories to your \$PATH in the file: ~/.profile:

```
PATH="$PATH:/usr/bin/python2:/usr/bin/python2.7:/usr/bin/python3:/usr/bin/python3-config:/usr/bin/python3.8:/usr/bin/python3.8-config:/usr/bin/python3-futurize:/usr/bin/python3-pasteurize:/usr/bin/node:/usr/bin/node-gyp:/usr/bin/nodejs:/usr/bin/npm:/usr/bin/ruby:/usr/bin/ruby2.7:/usr/bin/gem:/usr/bin/gem2.7"
```

7. Add the items to your path immediately:

```
source ~/.profile
```

8. Verify the path:

```
echo $PATH
```

xclip

1. **xclip** website: [Ubuntu Manpage: xclip - command line interface to X selections \(clipboard\)](#)

2. Install **xclip**:

```
sudo apt update && sudo apt full-upgrade -y  
  
sudo apt install -y xclip
```

3. Test **xclip** installation:

```
xclip -version  
# Expected Result: xclip version 0.13 (or later)
```

neovim

Resources

Neovim website: [Home - Neovim](#)

[Neovim Help Docs - Short Cut Keys](#)

[Josh NeoVim Setup](#)

[Neovim CheatSheet](#)

Vlad's install script: [GitHub - DevInsideYou/install-neovim](#)

Use This Instead of Vlad's: [Installing Neovim · neovim/neovim Wiki · GitHub](#)

See Python Setup if checkhealth reveals problems: [Setting up Python for Neovim · deoplete-plugins/deoplete-jedi Wiki · GitHub](#)

Install neovim

Run Neovim's commands. This is a combination of Neovim commands and those from Vlad's DevInsideYou script.

```
sudo curl -L
https://github.com/neovim/neovim/releases/latest/download/nvim.appimage -o
/usr/local/bin/nvim

sudo chmod 755 /usr/local/bin/nvim

nvim --version | head -1
# Expected Result: NVIM v0.4.4 (or later)
```

Go to directory `~/.config/nvim`. If any part of the directory does not yet exist, create it.

```
mkdir -p ~/.config/nvim/
cd ~/.config/nvim/
```

Create and edit a file named `~/.config/nvim/init.vim`:

```
cd ~/.config/nvim/
nvim init.vim
```

Note: If you already installed xclip, you may not need to add the clipboard setting. Run **checkhealth** to evaluate clipboard status.

```
nvim
:checkhealth<ENTER>
```

If you installed xclip you shouldn't have to do the following set parameter procedure. Just skip this step.

However, if necessary, add the following text to the init.nvim file beginning on the first line:

```
set clipboard=unnamedplus
```

Save and exit. This is the only line you need to fix the clipboard if **xclip** is NOT installed.

Run **checkhealth** to evaluate the installation:

```
nvim<ENTER>
# Follow the menu and run 'checkhealth'
:checkhealth<ENTER>
```

checkhealth will probably list errors and warnings associated with Python 3, Python 2, **ruby** and **gem** and Node.js. Follow the instructions and eliminate the errors.

It takes work, but nvim will eventually show all green no errors. My personal installations require the following fixes:

a. Python2 Fix:

```
python2 -m pip install --user --upgrade pynvim
```

b. Python3 Fix:

```
python3 -m pip install --user --upgrade pynvim
```

c. Add the following commands to the file **~/.config/nvim/init.vim** :

```
let g:python3_host_prog = '/usr/bin/python3'
let g:python_host_prog = '/usr/bin/python2'
```

d. Ruby Fix:

```
sudo gem install neovim
sudo gem environment
```

e. Node.js Fix:

```
sudo npm install -g neovim
```

f. Perl Error

"Neovim::Ext" appears to be broken. Ignore this error.

```
## Perl provider (optional)
- ERROR: perl provider error:
  - ADVICE:
    - "Neovim::Ext" cpan module is not installed
```

g. Python 2 Error

You may have to run the following to eliminate a python2 error:

```
/usr/bin/python2 -m pip install pynvim
```

10. Log Out and Log On again.

11. Run **checkhealth** again to evaluate the installation:

```
nvim<ENTER>
# Follow the menu and run 'checkhealth'
:checkhealth<ENTER>
```

Everything except the Perl Error should be green. No errors.

Yarn

yarn is a JavaScript package manager compatible with npm that helps you automate the process of installing, updating, configuring, and removing npm packages. It caches every download package and speeds up the installation process by parallelizing operations.

Resource

[Yarn Install](#)

Determine If yarn Is Already Installed

yarn may have been installed as part of the **nodejs** installation.

```
yarn --version
```

If **yarn** is already installed, skip this installation and proceed to the next step.

Add Key

To add the Yarn repository with the GPG key, enter the following commands:

```
sudo apt update

curl -ss https://dl.yarnpkg.com/debian/pubkey.gpg | sudo apt-key add
```

Add Repository

```
echo "deb https://dl.yarnpkg.com/debian/ stable main" |
sudo tee /etc/apt/sources.list.d/yarn.list
```

```
(base) younis@younisx:~$ echo "deb https://dl.yarnpkg.com/debian/ stable ma
in" | sudo tee /etc/apt/sources.list.d/yarn.list
deb https://dl.yarnpkg.com/debian/ stable main
(base) younis@younisx:~$
```

The system is now ready to add Yarn to its active programs.

Install Yarn

Check whether your system has Node.js on it. If Node.js is preinstalled, you can custom install yarn by running the following command:

```
sudo apt install --no-install-recommends yarn
```

Otherwise, if Node.js is not pre-installed, you will need to download the Node.js dependencies with Yarn.

```
sudo apt update && sudo apt upgrade -y
```

```
sudo apt install yarn
```

Verify Install

```
yarn --version
```

unzip:

You will need a utility to unzip zip files.

1. Determine if **unzip** is already installed.

```
unzip --h  
# Expected Result: Should show help.
```

If **unzip** is already installed, skip this installation and proceed to the next step.

2. On the Linux distro, install the Linux application, **unzip**.

```
sudo apt install -y unzip
```

3. Test the **unzip** installation:

```
unzip --h  
# Expected Result: Should show help.
```

Nautilus

A file manager for Linux.

Resources

[Nautilus - Debian Wiki](#)

[Best Linux File Managers you can use in 2021 | ComputingForGeeks](#)

[Open Files & Folders as Administrator in Nautilus in Ubuntu \(itsfoss.com\)](#)

Determine if Nautilus is already installed:

```
nautilus --version
```

If Nautilus is NOT installed, install it:

```
sudo apt install nautilus
```

Visual Studio Code - VSCode

[Running Visual Studio Code on Linux](#)

1. The **apt** repository and key can also be installed manually with the following script:

```
wget -qO- https://packages.microsoft.com/keys/microsoft.asc | gpg --dearmor  
> packages.microsoft.gpg  
  
sudo install -o root -g root -m 644 packages.microsoft.gpg  
/etc/apt/trusted.gpg.d/  
  
sudo sh -c 'echo "deb [arch=amd64,arm64,armhf signed-  
by=/etc/apt/trusted.gpg.d/packages.microsoft.gpg]  
https://packages.microsoft.com/repos/code stable main" >  
/etc/apt/sources.list.d/vscode.list'  
  
rm -f packages.microsoft.gpg
```

2. Then update the package cache and install the package using:

```
sudo apt install apt-transport-https  
sudo apt update  
sudo apt install code
```

3. Installed Extensions:

```
Go  
Markdown All in One  
YAML
```

Alacritty

A fast, cross-platform, OpenGL terminal emulator.

Resources

[Alacritty Github Install](#)

[How to Install and Use Alacritty](#)

[Docs 1](#)

[Color Schemes](#)

Install Alacritty

- Install Rust:

```
sudo curl https://sh.rustup.rs -ssf | sh
```

- Log Off and Log Back On Again
- Install Alacritty Software:

```
sudo add-apt-repository ppa:mmstick76/alacritty  
sudo apt update && sudo apt full-upgrade -y  
sudo apt install alacritty
```

- Setup Configuration File
 - Download the default configuration file (*alacritty.yml*) from the releases page (Asset Section): [Releases · alacritty/alacritty · GitHub](#)
- Note: A sample configuration file is maintained at:

```
D:\Programming\Linux\Ubuntu\VMSetup\TestVM01Setup\alacritty\ConfigFiles
```

- Create the directory `~/.config/alacritty/`:

```
sudo mkdir -p ~/.config/alacritty/
```

- Copy the default configuration file to the new directory:

```
~/.config/alacrittyalacritty.yml
```

- Log Off and Log On again.

Alacritty Color Schemes

[Color Schemes](#)

Install Alacritty Color Schemes

alacritty MUST be successfully installed, per instructions above. before executing these commands.

```
sudo npm i -g alacritty-themes
```

Note: You will receive some errors complaining about the *node* version. Ignore the errors, *alacritty-themes* should work fine for our purposes.

Make a Color Selection

Execute the following command and choose a color scheme.

```
alacritty-themes
```

Use the arrow keys to make a color selection.

Note: 3024.dark theme works with *htop*.

tmux

tmux is a terminal multiplexer: it enables a number of terminals to be created, accessed, and controlled from a single screen. tmux may be detached from a screen and continue running in the background, then later reattached.

Resources

[tmux Read Me](#)

[tmux tutorial - phoenixNAP](#)

[Build From Tarball](#)

[Good Install From Source](#)

[Best Install From Source](#)

[tmux documentation](#)

[Installation and Use](#)

Installation

```
sudo apt install tmux
```

Locate Command

The [locate](#) is a command line utility for finding files by name in Linux, just like [find command](#). However, it works more efficiently compared to its counterpart; it uses one or more databases populated by the **updatedb** program and prints file names matching at least one of the patterns (a user provides) to standard output.

Resources

[How to Install 'locate Command' to Find Files in Linux \(tecmint.com\)](#)

Determine If Locate Is Installed

```
locate bash_completion.sh
```

```
rpm -qa | grep findutils
```

Install Locate

```
sudo apt install mlocate
```

After installing **mlocate**, you need to update the **updatedb**, which is used by **locate** command as root user with the [sudo command](#), otherwise you will get an error. The default database storage location is **/var/lib/mlocate/mlocate.db**.

```
sudo updatedb
```

Verify Install

Once the database is updated, now try to run the **locate** command, which should work this time around.

```
locate bash_completion.sh
```

Script Files Directory

1. Create a directory structure to house script (.sh) files:

```
mkdir -p ~/localbashops/scripts  
mkdir -p ~/localbashops/cheatsheets
```

Script files will be stored here.

2. Add the *scripts* and *cheatsheets* directories to your path in **~/.profile**

Cheat Sheets

Load and Store Cheat Sheets

1. Create a directory to house cheat sheets which will be needed to support virtual machine operations:

```
mkdir ~/cheatsheets/
```

2. Add your *cheatsheets* to this directory. These files are usually **html** or **pdf** files.

```
Book_BashNotesForProfessionals.pdf
Book_LinuxNotesForProfessionals.pdf
GitCheatSheet.pdf
NanoCheatSheet.pdf
TestVM01Setup.pdf
tmuxCheatSheet.pdf
UbuntuCheatSheet.pdf
vimCheatSheet.pdf
```

3. Standard Cheat Sheets are located on Mike's Machine at

```
D:\Programming\Linux\Ubuntu\VMSetup\TestVM01Setup\scriptFiles\CheatSheets
```

4. Add the **~/cheatsheets/** directory to your path in **~/.profile**.

Cheat Sheet Scripts

1. Store your Cheat Sheet Scripts in the following directory:

```
~/localbashops/cheatsheets
```

2. Sample Cheat Sheet Script **tmuxCheatSheet.sh**:

```
#!/bin/bash

# Run Asynchronous Fire Fox command
/usr/bin/firefox --new-window ~/cheatsheets/tmuxCheatSheet.pdf &
```

3. Standard Cheat Sheet Scripts are located on Mike's Machine at:

```
D:\Programming\Linux\Ubuntu\VMSetup\TestVM01Setup\scriptFiles\cheatsheets
```

Grant Read/Write Permissions to Script Files

Be sure to give your script files Read and Execute permissions.

```
sudo chmod a+rx ~/localbashops/cheatsheets/Book_BashNotesForProfessionals.sh
sudo chmod a+rx ~/localbashops/cheatsheets/Book_LinuxNotesForProfessionals.sh
sudo chmod a+rx ~/localbashops/cheatsheets/GitCheatSheet.sh
sudo chmod a+rx ~/localbashops/cheatsheets/NanoCheatSheet.sh
sudo chmod a+rx ~/localbashops/cheatsheets/TestVM01Setup.sh
sudo chmod a+rx ~/localbashops/cheatsheets/tmuxCheatSheet.sh
sudo chmod a+rx ~/localbashops/cheatsheets/UbuntuCheatSheet.sh
sudo chmod a+rx ~/localbashops/cheatsheets/vimCheatSheet.sh
```

Cheat Sheets Symbolic Links

Setup Symbolic links for execution of Cheatsheet scripts and fast display of Cheatsheets :

```
sudo ln -s ~/localbashops/cheatsheets/Book_BashNotesForProfessionals.sh
/usr/bin/Book_BashNotesForProfessionals

sudo ln -s ~/localbashops/cheatsheets/Book_LinuxNotesForProfessionals.sh
/usr/bin/Book_LinuxNotesForProfessionals

sudo ln -s ~/localbashops/cheatsheets/GitCheatSheet.sh /usr/bin/GitCheatSheet

sudo ln -s ~/localbashops/cheatsheets/NanoCheatSheet.sh /usr/bin/NanoCheatSheet

sudo ln -s ~/localbashops/cheatsheets/TestVM01Setup.sh /usr/bin/TestVM01Setup

sudo ln -s ~/localbashops/cheatsheets/tmuxCheatSheet.sh /usr/bin/tmuxCheatSheet

sudo ln -s ~/localbashops/cheatsheets/UbuntuCheatSheet.sh
/usr/bin/UbuntuCheatSheet

sudo ln -s ~/localbashops/cheatsheets/vimCheatSheet.sh /usr/bin/vimCheatSheet
```

htop

htop is used to monitor system resources (i.e. memory usage, processes, disk storage).

htop has a config file at:

```
$HOME/.config/htop/htoprc
OR
/home/mike/snap/htop/3335/.config/htop/htoprc
```

Note: Find the location with:

```
locate htoprc
```

Make a backup of the file before making modifications. Set the color scheme to zero or five (whatever works best).

```
color_scheme=5
```

This will promote compatibility with *alacritty*.

ssh

Resources

[How to install / enable SSH server on Ubuntu - Ubuntu 21.04 & old](#)

[How To Enable SSH On Ubuntu 21.04 Server](#)

Check OS Version

```
cat /etc/os-release
```

systemctl

```
sudo systemctl start ssh
```

ssh.service not found

The ssh service is NOT installed on this OS.

Install ssh service

Update System

```
sudo apt update && sudo apt full-upgrade -y
```

Install ssh

```
sudo apt install openssh-server
```

Verify Install and Start ssh

```
sudo systemctl start ssh
```

Get IP Address for this machine (Target Ubuntu)

```
ip a
```

Get User Name for this machine (Target Ubuntu)

```
whoami
```

Do ssh Login from Remote Machine

Log On to Remote Machine (Windows MikeMark10). Try to ssh into target Ubuntu machine

```
ssh shusain@192.168.1.2
```

'fingerprint' will ask if you want to continue connecting. Respond: **yes**

Configure Static IP Address For VM

- Access your network router.
- Assign a static IP address to this VM.

SSH Key-Based Authentication

SSH Key-Base Authentication allows one to log onto a remote machine without using a password.

Resources

[SSH Key Authentication](#)

[Linux 101 Hacks](#)

[ssh-keygen - Unix, Linux Command](#)

[How to use the ssh-keygen Command in Linux](#)

[scp command in Linux](#)

Background: Private and Public Keys

The Private Key will live on our host machine.

The Public Key is pushed to, and will live on, the remote machine; that is, the machine we want to log onto.

Creating Keys With ssh-keygen

ssh-keygen generates, manages and converts authentication keys for **ssh**. **ssh-keygen** can create RSA keys for use by SSH protocol version 1 and RSA or DSA keys for use by SSH protocol version 2. The type of key to be generated is specified with the **-t** option. If invoked without any arguments, **ssh-keygen** will generate an RSA key for use in SSH protocol 2 connections.

ssh-keygen Options

Tag	Description
-a <i>trials</i>	
	Specifies the number of primality tests to perform when screening DH-GEX candidates using the -T command.
-B	Show the bubblebabble digest of specified private or public key file.
-b <i>bits</i>	
	Specifies the number of bits in the key to create. For RSA keys, the minimum size is 768 bits and the default is 2048 bits. Generally, 2048 bits is considered sufficient. DSA keys must be exactly 1024 bits as specified by FIPS 186-2.
-C <i>comment</i>	
	Provides a new comment.
-c	Requests changing the comment in the private and public key files. This operation is only supported for RSA1 keys. The program will prompt for the file containing the private keys, for the passphrase if the key has one, and for the new comment.
-D <i>reader</i>	
	Download the RSA public key stored in the smartcard in <i>reader</i> .
-e	This option will read a private or public OpenSSH key file and print the key in a 'SECSH Public Key File Format' to stdout. This option allows exporting keys for use by several commercial SSH implementations.
-F <i>hostname</i>	
	Search for the specified <i>hostname</i> in a <code>known_hosts</code> file, listing any occurrences found. This option is useful to find hashed host names or addresses and may also be used in conjunction with the -H option to print found keys in a hashed format.
-f <i>filename</i>	
	Specifies the filename of the key file.
-G <i>output_file</i>	
	Generate candidate primes for DH-GEX. These primes must be screened for safety (using the -T option) before use.
-g	Use generic DNS format when printing fingerprint resource records using the -r command.

Tag	Description
-H	Hash a <code>known_hosts</code> file. This replaces all hostnames and addresses with hashed representations within the specified file; the original content is moved to a file with a <code>.old</code> suffix. These hashes may be used normally by ssh and sshd , but they do not reveal identifying information should the file's contents be disclosed. This option will not modify existing hashed hostnames and is therefore safe to use on files that mix hashed and non-hashed names.
-i	This option will read an unencrypted private (or public) key file in SSH2-compatible format and print an OpenSSH compatible private (or public) key to stdout. ssh-keygen also reads the 'SECSH Public Key File Format'. This option allows importing keys from several commercial SSH implementations.
-l	Show fingerprint of specified public key file. Private RSA1 keys are also supported. For RSA and DSA keys ssh-keygen tries to find the matching public key file and prints its fingerprint.
-M <i>memory</i>	
	Specify the amount of memory to use (in megabytes) when generating candidate moduli for DH-GEX.
-N <i>new_passphrase</i>	
	Provides the new passphrase.
-P <i>passphrase</i>	
	Provides the (old) passphrase.
-p	Requests changing the passphrase of a private key file instead of creating a new private key. The program will prompt for the file containing the private key, for the old passphrase, and twice for the new passphrase.
-q	Silence ssh-keygen . Used by <code>/etc/rc</code> when creating a new key.
-R <i>hostname</i>	
	Removes all keys belonging to <i>hostname</i> from a <code>known_hosts</code> file. This option is useful to delete hashed hosts (see the -H option above).
-r <i>hostname</i>	
	Print the SSHFP fingerprint resource record named <i>hostname</i> for the specified public key file.
-S <i>start</i>	
	Specify start point (in hex) when generating candidate moduli for DH-GEX.
-T <i>output_file</i>	

Tag	Description
	Test DH group exchange candidate primes (generated using the -G option) for safety.
-t <i>type</i>	
	Specifies the type of key to create. The possible values are "rsa1" for protocol version 1 and "rsa" or "dsa" for protocol version 2.
-U <i>reader</i>	
	Upload an existing RSA private key into the smartcard in <i>reader</i> .
-v	Verbose mode. Causes ssh-keygen to print debugging messages about its progress. This is helpful for debugging moduli generation. Multiple -v options increase the verbosity. The maximum is 3.
-W <i>generator</i>	
	Specify desired generator when testing candidate moduli for DH-GEX.
-y	This option will read a private OpenSSH format file and print an OpenSSH public key to stdout.

Generating the Public and Private Keys

```
ssh-keygen -t rsa -b 4096
```

The system will offer the opportunity to store the generated keys in a directory other than the default directory. The default directory is **/.ssh/id_rsa**.

The system will now ask you for a passphrase. You have the option to leave the passphrase blank. If you enter a passphrase, you will be required to enter it twice.

SCP Command - A Means Of Copying Files Over SSH

scp (secure copy) command in Linux system is used to copy file(s) between servers in a secure way. The SCP command or secure copy allows secure transferring of files in between the local host and the remote host or between two remote hosts. It uses the same authentication and security as it is used in the Secure Shell (SSH) protocol. SCP is known for its simplicity, security and pre-installed availability.

SCP Syntax

```
[scp [-346BCpqrTV] [-c cipher] [-F ssh_config] [-i identity_file] [-l limit] [-o ssh_option] [-P port] [-S program] [[user@]host1:]file1 ... [[user@]host2:]file2]
(https://www.thegeekdiary.com/using-the-ssh-keygen-command-in-linux/)
```

SCP Example

Step-1 - Copy Local Host File to Target Remote Machine

Copy Local Host File to Target Remote Machine and Target Directory.

```
scp (File To Copy on Local Host) (SPACE) (Target Machine):(Target Directory File Name Location)
```

```
scp ~/.ssh/id_rsa.pub coreyms@192.168.56.100:/home/coreyms/.ssh/uploaded_key.pub
```

This procedure will require you to log in with a password since the authorized keys are not yet configured.

Step-2 - Add New Copied Key to authorized_keys

On the Target Machine, add the new copied file containing the key (`uploaded_key.pub`) to the `authorized_keys` file.

```
cat ~/.ssh/uploaded_key.pub >> ~/.ssh/authorized_keys
```

This will append the key contained in the `uploaded_key.pub` file to the content of the `authorized_keys` file.

Step-3 Verify that the key was in fact added to the *authorized_keys* File

```
cat ~/.ssh/authorized_keys
```

Set Permissions On .ssh Folder and Files

Set Permissions on the .ssh Folder

```
chmod 700 ~/.ssh/
```

Set Permissions on All Files Within the .ssh Folder

```
chmod 600 ~/.ssh/*
```

SSH to Remote Machine with SSH Key Authentication

A password is no longer required in order to access the remote machine. The keys have been successfully configured and access from the Local Host to the Target Machine will be granted immediately.

Syntax

```
ssh (TargetMachine)
```

Example

```
ssh coreyms@192.168.56.100
```

Disable Password Authentication

This is a security measure designed to prevent log ins to the Target Remote Machine using password login procedures. After implementing this security measure, all log ins must be performed with SSH Key Authentication. External Password SSH Logins will be no longer allowed.

Step-1 Make a copy of the SSH Configuration File

Save the current SSH Configuration File as a backup.

```
sudo cp /etc/ssh/sshd_config /etc/ssh/sshd_config.bak
```

Step-2 Edit and Re-Configure SSH Configuration File

These changes will turn off password configuration.

```
sudo nano /etc/ssh/sshd_config
```

Step-3 Locate and Change *Password Authentication* Parameter

Scroll down the sshd_config file and locate the *Password Authentication* parameter. It will be styled as follows:

```
#PasswordAuthentication yes
```

Change the *Password Authentication* parameter to:

```
PasswordAuthentication no
```

Save the file. On **nano** this done with **Ctrl+X -y**.

SSH Password Authentication is now disabled. However, the ssh service must first be restarted to implement the changes.

Step-4 Restart The SSH Service

```
sudo service ssh restart
```

Summary: Disable Password Authentication

Disabling Password Authentication in favor of SSH Key Authentication will make the system more secure and prevent unwanted logins in case the password is compromised.

